Self-Paced Learning for Matrix Factorization

Qian Zhao[1], Deyu Meng[1,*], Lu Jiang[2], Qi Xie[1], Zongben Xu[1], Alexander G. Hauptmann[2]
[1]School of Mathematics and Statistics, Xi'an Jiaotong University
[2]School of Computer Science, Carnegie Mellon University
timmy.zhaoqian@gmail.com, dymeng@mail.xjtu.edu.cn, lujiang@cs.cmu.edu
xq.liwu@stu.xjtu.edu.cn, zbxu@mail.xjtu.edu.cn, alex@cs.cmu.edu
[*]Corresponding author

Abstract

Matrix factorization (MF) has been attracting much attention due to its wide applications. However, since MF models are generally non-convex, most of the existing methods are easily stuck into bad local minima, especially in the presence of outliers and missing data. To alleviate this deficiency, in this study we present a new MF learning methodology by gradually including matrix elements into MF training from easy to complex. This corresponds to a recently proposed learning fashion called self-paced learning (SPL), which has been demonstrated to be beneficial in avoiding bad local minima. We also generalize the conventional binary (hard) weighting scheme for SPL to a more effective real-valued (soft) weighting manner. The effectiveness of the proposed self-paced MF method is substantiated by a series of experiments on synthetic, structure from motion and background subtraction data.

Introduction

Matrix factorization (MF) is one of the fundamental problems in machine learning and computer vision, and has wide applications such as collaborative filtering (Mnih and Salakhutdinov 2007), structure from motion (Tomasi and Kanade 1992) and photometric stereo (Hayakawa 1994). Basically, MF aims to factorize an $m \times n$ data matrix $\mathbf{Y}$, whose entries are denoted as y_{ij}s, into two smaller factors $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$, where $r \ll \min(m,n)$, such that $\mathbf{U}\mathbf{V}^T$ is possibly close to $\mathbf{Y}$. This aim can be achieved by solving the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} \ell(y_{ij}, [\mathbf{U}\mathbf{V}^T]_{ij}) + \lambda R(\mathbf{U},\mathbf{V}), \qquad (1)$$

where $\ell(\cdot,\cdot)$ denotes a certain loss function, Ω is the index set indicating the observed data, and $R(\mathbf{U},\mathbf{V})$ is the regularization term to guarantee generalization ability and numerical stability.

Under the Gaussian noise assumption, it is natural to utilize the least square (LS) loss in (1), leading to an L_2-norm MF problem. This problem has been extensively studied (Srebro and Jaakkola 2003; Buchanan and Fitzgibbon 2005; Okatani and Deguchi 2007; Mitra, Sheorey, and Chellappa 2010; Okatani, Yoshida, and Deguchi 2011). Another commonly utilized loss function is the least absolute deviation (LAD), which results in an L_1-norm MF problem (Ke and Kanade 2005; Eriksson and van den Hengel 2010; Zheng et al. 2012; Wang et al. 2012; Meng et al. 2013). Solving this optimization problem has also been attracting much attention since it performs more robust in the presence of heavy noises and outliers. Other loss functions beyond L_2- or L_1-norm have also been considered for specific applications (Srebro, Rennie, and Jaakkola 2005; Weimer et al. 2007; Meng and De la Torre 2013).

The main limitation of existing MF methods lies on the non-convexity of the objective functions they aim to solve. This deficiency often makes the current MF approaches getting stuck into bad local minima, especially in the presence of heavy noises and outliers. A heuristic approach for alleviating this problem is to run the algorithm multiple times with different initializations and pick the best solution among them. However, this strategy is ad hoc and generally inconvenient to implement in unsupervised setting, since there is no straightforward criterion for choosing a proper solution.

Recent advances in *self-paced learning* (SPL) (Kumar, Packer, and Koller 2010) provide a possible solution to this local minimum problem. The core idea of SPL is to train a model on "easy" samples first, and then gradually add "complex" samples into consideration, which well simulates the process of human learning. This methodology has been empirically demonstrated to be beneficial in avoiding bad local minima and achieving a better generalization result (Kumar, Packer, and Koller 2010; Tang et al. 2012; Kumar et al. 2011). Therefore, incorporating it into MF is expected to alleviate the local minimum issue.

In this paper, we present a novel approach, called *self-paced matrix factorization* (SPMF), for the MF task. Specifically, we construct a concise SPMF formulation which can be easily employed to embed the SPL strategy into general MF objectives, including the L_2- and L_1-norm MF. We also design a simple yet effective algorithm for solving the proposed SPMF problem. Experimental results substantiate that our method improves the performance of the state-of-the-art L_2- and L_1-norm MF methods. Besides, we theoretically explain the insight for its effectiveness in the L_2-norm case by deducing an error bound for the weighted MF problem.

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related Work

Matrix Factorization

Matrix factorization in the presence of missing data has attracted much attention in machine learning and computer vision for decades. The L_2-norm based MF problem has been mostly investigated along this line. In machine learning community, Srebro and Jaakkola (2003) proposed an EM based method and applied it to collaborative filtering. Mnih and Salakhutdinov (2007) prompted the L_2-norm MF model under probabilistic framework, and further investigated it using Bayesian approach (Salakhutdinov and Mnih 2008). In computer vision circle, Buchanan and Fitzgibbon (2005) presented a damped Newton algorithm, using the information of second derivatives with a damping factor. To enhance the efficiency in large-scale computation, Mitra et al. (2010) converted the original problem into a low-rank semidefinite programming. Okatani and Deguchi (2007) extended the Wiberg algorithm to this problem, which has been further improved by Okatani et al. (2011) via incorporating a damping factor.

In order to introduce robustness to outliers, the L_1-norm MF problem has also been paid much attention in recent years. Ke and Kanade (2005) solved this problem via alternative linear/quadratic programming (ALP/AQP). To enhance the efficacy, Eriksson and van den Hengel (2010) designed an L_1-Wiberg approach by extending the classical Wiberg method to L_1 minimization. Through adding convex trace-norm regularization, Zheng et al. (2012) proposed a RegL1ALM method to improve convergence. Wang et al. (2012) considered the problem in a probabilistic framework, and Meng et al. (2013) proposed a cyclic weighted median approach to further improve the efficiency.

Beyond L_2- or L_1-norm, other loss functions have also been attempted. Srebro et al. (2005) and Weimer et al. (2007) utilized Hinge loss and NDCG loss, respectively, for collaborative filtering application. Besides, Lakshminarayanan et al. (2011) and Meng and De la Torre (2013) adopted some robust likelihood functions to make MF less sensitive to outliers.

Most of the current MF methods are developed on nonconvex optimization problems, and thus always encounter the local minimum problem, especially when missing data and outliers exist. We thus aim to alleviate this issue by advancing it into the SPL framework.

Self-Paced Learning

Inspired by the intrinsic learning principle of humans/animals, Bengio et al. (2009) proposed a new machine learning framework, called *curriculum learning*. The core idea is to incrementally involve sequence of samples into learning, where easy samples are introduced first and more complex ones are gradually included when the learner is ready for them. These gradually included sample sequences from easy to complex correspond to the curriculums learned in different grown-up stages of humans/animals. This strategy, as supported by empirical evaluation, is helpful in alleviating the bad local optimum problem in non-convex optimization (Ni and Ling 2010; Basu and Christensen 2013).

Instead of using the aforementioned heuristic strategies, Kumar et al. (2010) formulated the key principle of curriculum learning as a concise optimization model, called *self-paced learning* (SPL), and applied it to latent variable models. The SPL model includes a weighted loss term on all samples and a general SPL regularizer imposed on sample weights. By sequentially optimizing the model with gradually increasing penalty parameter on the SPL regularizer, more samples can be automatically included into training from easy to complex in a pure self-paced way. Multiple applications of this SPL framework have also been attempted, such as object detector adaptation (Tang et al. 2012), specific-class segmentation learning (Kumar et al. 2011), visual category discovery (Lee and Grauman 2011), and long-term tracking (Supančič III and Ramanan 2013).

The current SPL framework can only select samples into training in a "hard" way (binary weight). This means that the selected/unselected samples are treated equally easy/complex. However, this assumption tends to lose flexibility since any two samples are less likely to be strictly equally learnable. We thus expect to abstract an insightful definition for the SPL principle, and then extend it to a "soft" version (real-valued weights).

Self-Paced Matrix Factorization

Model Formulation

The core idea of the proposed SPMF framework is to sequentially include elements of $\mathbf{Y}$ into MF training from easy to complex. This aim can be realized by solving the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{w}} \sum_{(i,j)\in\Omega} w_{ij}\ell(y_{ij},[\mathbf{U}\mathbf{V}^T]_{ij}) + \lambda R(\mathbf{U},\mathbf{V}) + \sum_{(i,j)\in\Omega} f(w_{ij},k),$$
$$s.t. \quad \mathbf{w}\in[0,1]^{|\Omega|}, \tag{2}$$

where $\mathbf{w} = \{w_{ij}|(i,j)\in\Omega\}$ denotes the weights imposed on the observed elements of $\mathbf{Y}$, and $\sum_{(i,j)\in\Omega} f(w_{ij},k)$ is the self-paced regularizer determining the samples to be selected in training.

The previously adopted $f(w,k)$ was simply $f(w,k) = -\frac{1}{k}w$ (Kumar, Packer, and Koller 2010). Under this regularizer, when $\mathbf{U},\mathbf{V}$ are fixed, the optimal w_{ij} is calculated as:

$$w_{ij}^*(k,\ell_{ij}) = \begin{cases} 1 & \text{if } \ell_{ij} \le 1/k \\ 0 & \text{if } \ell_{ij} > 1/k, \end{cases} \tag{3}$$

where the abbreviation ℓ_{ij} represents $\ell(y_{ij},[\mathbf{U}\mathbf{V}^T]_{ij})$ for the convenience of notation. It is easy to see that if a sample's loss is less than the threshold $1/k$, it will be selected as an easy sample ($w_{ij}^* = 1$), or otherwise unselected ($w_{ij}^* = 0$). The parameter k controls the pace at which the model learns new samples. Physically, $1/k$ corresponds to the "age" of the model: when $1/k$ is small, only the easiest samples with the smallest losses will be considered; as $1/k$ increases, more samples with larger losses will be gradually included to train a "mature" model.

From this intuition, we present a general definition for the self-paced regularizer by mathematically abstracting its insightful properties as follows:

Definition 1 (Self-paced regularizer) *Suppose that w is a weight variable, ℓ is the loss, and k is the learning pace parameter. $f(w,k)$ is called self-paced regularizer, if*

1. *$f(w,k)$ is convex with respect to $w \in [0,1]$;*
2. *$w^*(k,\ell)$ is monotonically decreasing with respect to ℓ, and it holds that $\lim_{\ell \to 0} w^*(k,\ell) = 1$, $\lim_{\ell \to \infty} w^*(k,\ell) = 0$;*
3. *$w^*(k,\ell)$ is monotonically increasing with respect to $\frac{1}{k}$, and it holds that $\lim_{k \to 0} w^*(k,\ell) \leq 1$, $\lim_{k \to \infty} w^*(k,\ell) = 0$;*

where $w^*(k,\ell) = \arg\min_{w \in [0,1]} w\ell + f(w,k)$.

The three conditions in Definition 1 provide an axiomatic understanding for the SPL. Condition 2 indicates that the model inclines to select easy samples (with smaller losses) in favor of complex samples (with larger losses). Condition 3 states that when the model "age" (controlled by the pace parameter k) gets larger, it tends to incorporate more, probably complex, samples to train a "mature" model. The limits in these two conditions impose the upper and lower bounds for w. The convexity in Condition 1 further ensures the soundness of this regularizer for optimization.

It is easy to verify that the function $f(w,k) = -\frac{1}{k}w$ satisfies all of the three conditions in Definition 1, and implements "hard" selection of samples by assigning binary weights to them, as shown in Figure 1. It has been demonstrated that in many real applications, e.g., Bag-of-Words quantization, however, "soft" weighting is more effective than the "hard" way (Jiang, Ngo, and Yang 2007). Besides, in practice, the noises embedded in data are generally nonhomogeneous across samples. Soft weighting, which assigns real-valued weights, inclines to more faithfully reflect the latent importance of samples in training.

Instead of hard weighting, the proposed definition facilitates us to construct the following soft regularizer:

$$f(w,k) = \frac{\gamma^2}{w + \gamma k}, \tag{4}$$

where parameter $\gamma > 0$ is introduced to control the strength of the weights assigned to the selected samples. It is easy to derive the optimal solution to $\min_{w \in [0,1]} w\ell + f(w,k)$ as:

$$w^*(k,\ell) = \begin{cases} 1 & \text{if } \ell \leq \frac{1}{\sqrt{k+1/\gamma}}, \\ 0 & \text{if } \ell \geq \frac{1}{\sqrt{k}}, \\ \gamma\left(\frac{1}{\sqrt{\ell}} - k\right) & \text{otherwise.} \end{cases} \tag{5}$$

The $w^*(k,\ell)$ tendency curve with respect to ℓ is shown in Figure 1. It can be seen that, when the loss is less than a threshold $1/\sqrt{k}$, the corresponding sample is treated as an easy sample and assigned to a non-zero weight; if the loss is further smaller than $1/\sqrt{k+1/\gamma}$, the sample is treated as a faithfully easy sample weighted by 1. This on one hand inherits the easy-sample-first property of the original self-paced regularizer (Kumar, Packer, and Koller 2010), and on the other hand incorporates the soft weighting strategy into training. The effectiveness of the proposed self-paced regularizer will be evaluated in the experiment section.

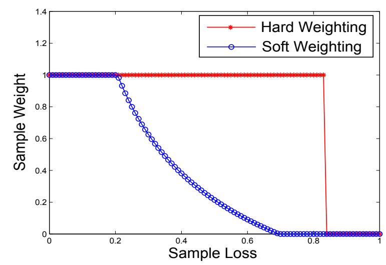

Figure 1: Comparison of hard weighting scheme and soft weighting ($\gamma = 1$) scheme, with $k = 1.2$.

Combining (2) and (4), the proposed SPMF model can then be formulated as follows:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{w}} \sum_{(i,j)\in\Omega} w_{ij}\ell(y_{ij}, [\mathbf{U}\mathbf{V}^T]_{ij}) + \lambda R(\mathbf{U},\mathbf{V}) + \sum_{(i,j)\in\Omega} \frac{\gamma^2}{w_{ij} + \gamma k},$$
$$s.t. \quad \mathbf{w} \in [0,1]^{|\Omega|}. \tag{6}$$

It should be noted that, based on Definition 1, it is easy to derive other types of self-paced regularizers (Jiang et al. 2014). However, in our practice, we found that the proposed regularizer generally performs better for MF problem.

Self-Paced Learning Process

Similar as the method utilized in (Kumar, Packer, and Koller 2010), we use alternative search strategy (ASS) to solve SPMF. ASS is an iterative method for solving optimization by dividing variables into two disjoint blocks and alternatively optimizing each of them with the other fixed. In SPMF, under fixed $\mathbf{U}$ and $\mathbf{V}$, $\mathbf{w}$ can be optimized by

$$\min_{\mathbf{w}\in[0,1]^{|\Omega|}} \sum_{(i,j)\in\Omega} \left\{ w_{ij}\ell_{ij} + \frac{\gamma^2}{w_{ij} + \gamma k} \right\}, \tag{7}$$

where ℓ_{ij} is calculated under the current $\mathbf{U}$ and $\mathbf{V}$. This optimization is separable with respect to each w_{ij}, and thus can be easily solved by (5). When $\mathbf{w}$ is fixed, the problem corresponds to the weighted MF:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} w_{ij}\ell(y_{ij}, [\mathbf{U}\mathbf{V}^T]_{ij}) + \lambda R(\mathbf{U},\mathbf{V}), \tag{8}$$

and off-the-shelf algorithms can be employed for solving it. The whole process is summarized in Algorithm 1.

This is a general SPMF framework and can be incorporated with any MF task by specifying the loss functions and regularization terms. In this paper we focus on two commonly used loss functions: the LS loss $\ell(y_{ij}, [\mathbf{U}\mathbf{V}^T]_{ij}) = \left(y_{ij} - [\mathbf{U}\mathbf{V}^T]_{ij}\right)^2$ and the LAD loss $\ell(y_{ij}, [\mathbf{U}\mathbf{V}^T]_{ij}) = \left|y_{ij} - [\mathbf{U}\mathbf{V}^T]_{ij}\right|$, which lead to L_2- and L_1-norm MF problems, respectively. We considered two types of regularization terms: one is the widely used L_2-norm regularization $R(\mathbf{U},\mathbf{V}) = \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right)$ (Buchanan and Fitzgibbon 2005; Mnih and Salakhutdinov 2007); the other is the trace-norm regularization (Zheng et al. 2012) $R(\mathbf{U},\mathbf{V}) = \|\mathbf{V}\|_* + I_{\{\mathbf{U}:\mathbf{U}^T\mathbf{U}=\mathbf{I}\}}(\mathbf{U})$, where $I_A(x)$ is the indicator function, which equals 1 if $x \in A$ and $+\infty$ otherwise. The latter

Algorithm 1 Self-paced matrix factorization algorithm

Input: Incomplete data matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$ with observation indexed by Ω, k_0, k_{end}, $\mu > 1$

1: Initialization: solve the MF problem with all the observation equally weighted to obtain $\mathbf{U}_0, \mathbf{V}_0$; calculate $\{\ell_{ij}\}_{(i,j)\in\Omega}$, $t \leftarrow 0$, $k \leftarrow k_0$

2: **while** $k > k_{\text{end}}$ **do**

3: $\quad \mathbf{w}_{t+1} = \underset{\mathbf{w}\in[0,1]^{|\Omega|}}{\arg\min} \sum_{(i,j)\in\Omega} \left\{ w_{ij}\ell_{ij} + \frac{\gamma^2}{w_{ij}+\gamma k} \right\}$.

4: $\quad \{\mathbf{U}_{t+1}, \mathbf{V}_{t+1}\} = \underset{\mathbf{U},\mathbf{V}}{\arg\min} \sum_{(i,j)\in\Omega} w_{ij}^{t+1} \ell(y_{ij}, [\mathbf{U}\mathbf{V}^T]_{ij}) + \lambda R(\mathbf{U},\mathbf{V})$.

5: $\quad$ Compute current $\{\ell_{ij}\}_{(i,j)\in\Omega}$.

6: $\quad t \leftarrow t+1$, $k \leftarrow k/\mu$.

7: **end while**

Output: $\mathbf{U} = \mathbf{U}_t$, $\mathbf{V} = \mathbf{V}_t$.

has been shown to be effective for rigid structure from motion problem (Zheng et al. 2012). For Step 4 of our algorithm, we modified the solvers proposed by Cabral et al. (2013) and Wang et al. (2012) to solve the L_2-norm regularized MF with the LS and LAD loss, respectively; and modified the solver proposed by Zheng et al. (2012) to solve the trace-norm regularized MF with both the LS and LAD loss.

Theoretical Explanation

In this section, we give a preliminary explanation for the effectiveness of SPMF under the LS loss. Let $\sqrt{\mathbf{W}}$ denote the element-wise square root of $\mathbf{W}$, and $\odot$ the Hadamard product (element-wise product) of matrices. We consider the following weighted L_2-norm MF problem:

$$\min_{\mathbf{U},\mathbf{V}} \left\| \sqrt{\mathbf{W}} \odot (\hat{\mathbf{Y}} - \mathbf{U}\mathbf{V}^T) \right\|_F^2 \quad s.t. \quad \left|[\mathbf{U}\mathbf{V}^T]_{ij}\right| \le b, \tag{9}$$

where $\hat{\mathbf{Y}} = \mathbf{Y} + \mathbf{E}$ is the corrupted matrix with ground truth $\mathbf{Y}$ and noise $\mathbf{E}$, and $\mathbf{W}$ is the weight matrix which satisfies $w_{ij} > 0$ if $(i,j) \in \Omega$ and $w_{ij} = 0$ otherwise, and $\sum_{(i,j)\in\Omega} w_{ij} = |\Omega|$. Note that the L_2-norm regularization term $R(\mathbf{U},\mathbf{V}) = \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right)$ imposed on the matrices $\mathbf{U}$ and $\mathbf{V}$ can naturally induce the magnitude constraint on each element of their product. We utilize this simpler boundness constraint for the convenience of proof. As a comparison, we also consider the following un-weighted L_2-norm MF problem:

$$\min_{\mathbf{U},\mathbf{V}} \left\| P_\Omega(\hat{\mathbf{Y}} - \mathbf{U}\mathbf{V}^T) \right\|_F^2 \quad s.t. \quad \left|[\mathbf{U}\mathbf{V}^T]_{ij}\right| \le b, \tag{10}$$

where P_Ω is the sampling operator defined as $[P_\Omega(\mathbf{Y})]_{ij} = y_{ij}$ if $(i,j) \in \Omega$ and 0 otherwise.

Denoting the optimal solution of (9) as $\mathbf{Y}^* = \mathbf{U}^*\mathbf{V}^{*T}$, the following theorem presents an upper bound for the closeness between $\mathbf{Y}^*$ and the gound truth matrix $\mathbf{Y}$ by *root mean square error* (RMSE): $\frac{1}{\sqrt{mn}} \|\mathbf{Y}^* - \mathbf{Y}\|_F$.

Theorem 1 *For a given matrix* $\mathbf{W}$ *which satisfies* $w_{ij} \begin{cases} > 0, & (i,j) \in \Omega \\ = 0, & \text{otherwise} \end{cases}$*, with* $\sum_{(i,j)\in\Omega} w_{ij} = |\Omega|$ *and* $\sum_{(i,j)\in\Omega} w_{ij}^2 \le 2|\Omega|$*, there exists a constant* C*, such that with probability at least* $1 - 2\exp(-n)$*,*

$$\text{RMSE} \le \frac{1}{\sqrt{|\Omega|}} \left\| \sqrt{\mathbf{W}} \odot \mathbf{E} \right\|_F + \frac{1}{\sqrt{mn}} \|\mathbf{E}\|_F + Cb\left(\frac{nr\log(n)}{|\Omega|}\right)^{\frac{1}{4}} \tag{11}$$

Here, we assume $m \le n$ *without loss of generality.*

The proof is listed in supplementary material due to page limitation. When $|\Omega| \gg nr\log(n)$, i.e., sufficiently many entries of $\hat{\mathbf{Y}}$ are sampled, the last term of the above bound diminishes, and the RMSE is thus essentially bounded by the first two terms. Also note that the second term is a constant irrelevant to sampling and weighting mechanism, and the RMSE is thus mainly affected by the first term.

Based on this result, we give an explanation for the effectiveness of the proposed framework for L_2-norm MF. Given observed entries from $\hat{\mathbf{Y}}$ whose indices are denoted by Ω_0, assuming $\Omega_0 \gg nr\log(r)$, we can solve (10) with Ω_0. Then the RMSE can be bounded using (11) with $w_{ij} = 1$ if $(i,j) \in \Omega_0$ and $w_{ij} = 0$ otherwise, and thus mainly determined by $(\sqrt{|\Omega_0|})^{-1} \|P_{\Omega_0}(\mathbf{E})\|_F$. This is also the case studied by Wang and Xu (2012). We choose $\Omega_1 \subset \Omega_0$, which indexes the first $|\Omega_1|$ smallest elements of $\{e_{ij}^2\}_{(i,j)\in\Omega_0}$, where e_{ij}s are entries of $\mathbf{E}$, also assuming $\Omega_1 \gg nr\log(r)$, and solve (10) with Ω_1. Then the corresponding RMSE will be mainly affected by $(\sqrt{|\Omega_1|})^{-1} \|P_{\Omega_1}(\mathbf{E})\|_F$, which is smaller than $(\sqrt{|\Omega_0|})^{-1} \|P_{\Omega_0}(\mathbf{E})\|_F$. Now we can further assign weights $\{w_{ij}\}$ according to Ω_1 such that the assumptions of Theorem 1 are satisfied, and solve the corresponding problem (9). The obtained RMSE can be bounded using (11), which is mainly affected by $(\sqrt{|\Omega_1|})^{-1}\|\sqrt{\mathbf{W}} \odot \mathbf{E}\|_F$. If w_{ij}s are specified from small to large in accordance with the descending order of $\{e_{ij}^2\}_{(i,j)\in\Omega_1}$, $(\sqrt{|\Omega_1|})^{-1}\|\sqrt{\mathbf{W}} \odot \mathbf{E}\|_F$ is then further smaller than $(\sqrt{|\Omega_1|})^{-1} \|P_{\Omega_1}(\mathbf{E})\|_F$.

From the above analysis, we can conclude that by properly selecting samples and assigning weights, better approximation to the ground truth matrix $\mathbf{Y}$ can be attained by weighted L_2-norm MF, compared with the un-weighted version. Since the underlying $\{e_{ij}^2\}$ is unknown in practice, we cannot guarantee to select samples and assign weights in an exactly correct way. However, we can still estimate $\{e_{ij}^2\}$ with losses evaluated by current approximation, which is exactly what SPMF does. Besides, by iteratively selecting and re-weighting samples, this estimation is expected to be gradually more accurate. This thus provides a rational explanation for the effectiveness of SPMF.

Experiments

We evaluate the performance of the proposed SPMF approach, denoted as SPMF-L_2 (L_2-reg), SPMF-L_2 (trace-reg), SPMF-L_1 (L_2-reg) and SPMF-L_1 (trace-reg) for L_2- and L_1-norm MF with L_2- and trace-norm regularization, respectively, on synthetic, structure from motion and background subtraction data. The competing methods include representative MF methods designed for handling missing data: DWiberg (Okatani, Yoshida, and Deguchi 2011), RegL1ALM (Zheng et al. 2012), PRMF (Wang et al. 2012), CWM (Meng et al. 2013), and a recently proposed MoG

Table 1: Performance comparison of 11 competing MF methods in terms of RMSE and MAE on synthetic data. The results are averaged over 50 runs, and the best and the second best results are highlighted in bold with and without underline, respectively.

Method	L2-ALM (L_2-reg)	L2-ALM (trace-reg)	DWiberg	RegL1ALM	PRMF	CWM	MoG	SPMF-L_2 (L_2-reg)	SPMF-L_2 (trace-reg)	SPMF-L_1 (L_2-reg)	SPMF-L_1 (trace-reg)
RMSE	3.7520	3.8460	4.3658	0.1412	0.2688	0.1359	0.1622	0.1152	0.1119	**0.0632**	**0.0636**
MAE	2.7147	2.7522	2.8224	0.0761	0.1768	0.0890	0.0634	0.0714	0.0688	**0.0481**	**0.0487**

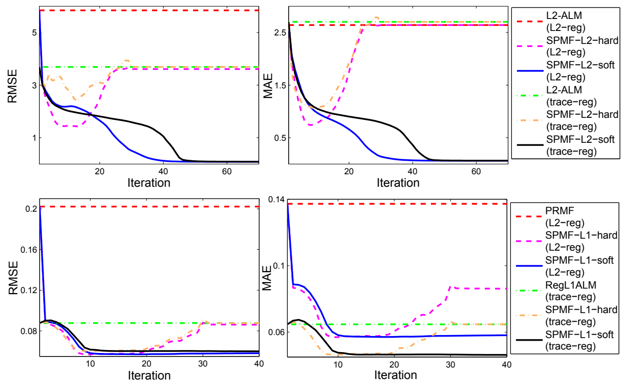

Figure 2: Tendency curves of RMSE and MAE with respect to iterations for SPMF-L_2 (top) and SPMF-L_1 (bottom).

method (Meng and De la Torre 2013). We used the publicly available codes from the authors' websites except MoG provided by the authors.

Synthetic Data

The data were generated as follows: two matrices $\mathbf{U}$ and $\mathbf{V}$, both of which are of size 100×4, were first randomly generated with each entry drawn from the Gaussian distribution $\mathcal{N}(0,1)$, leading to a ground truth rank-4 matrix $\mathbf{Y}_0 = \mathbf{U}\mathbf{V}^T$. Then 40% of the entries were designed as missing data, 20% of the entries were added to uniform noise on $[-20, 20]$, and the rest entries were added to Gaussian noise drawn from $\mathcal{N}(0, 0.1^2)$.

The experiments were implemented with 50 realizations. For each realization, we ran each method, except SPMF methods, 80 times with randomly initializations and pick the best output in terms of the objective function. This is aimed to heuristically alleviate the bad local minimum issue of the conventionally MF methods with similar computational cost as SPMF (80 is larger than the number of subproblems solved in SPMF). Two criteria were adopted for performance assessment. (1) RMSE: $\frac{1}{\sqrt{mn}}\|\mathbf{Y}_0 - \hat{\mathbf{U}}\hat{\mathbf{V}}^T\|_F$, and (2) *mean absolute error* (MAE): $\frac{1}{mn}\|\mathbf{Y}_0 - \hat{\mathbf{U}}\hat{\mathbf{V}}^T\|_1$, where $\hat{\mathbf{U}}, \hat{\mathbf{V}}$ denote the outputs from a utilized MF method. The performance of each competing method was evaluated in terms of these two criteria, as the average over the 50 realizations, and reported in Table 1.

As can be seen from Table 1, for both regularization terms, SPMF-L_1 achieves the best performance among all the competing methods. It can also be observed that, although based on the LS loss, SPMF-L_2 outperforms the utilized robust MF methods. This shows that, by the proposed strategy, L_2-norm MF can be more robust against outliers.

To better understand the behavior of the proposed self-paced regularizer, we plot in Figure 2 the curves of RMSE and MAE with respect to SPL iterations using both the hard and soft self-paced regularizers. We also show the performance of the baseline methods, i.e., L2-ALM (L_2-reg), L2-ALM (trace-reg), PRMF and RegL1ALM, for easy comparison. The figure shows that, by iteratively selecting samples and assigning weights, both of the two regularizers can improve the baseline in the first several iterations. When the iteration continues, the performance of the hard regularizer gradually degenerates, while the estimation by the soft regularizer consistently becomes more accurate. This shows that the utilized soft regularizer is more stable than the hard regularizer. Similar behavior was also observed in the experiments on real data, and thus we only report the results of the proposed soft regularizer in what follows.

Structure From Motion

Structure from motion (SFM) aims to estimate 3-D structure from a sequence of 2-D images which are coupled with local motion information. There are two types of SFM problems, namely rigid and nonrigid SFM, both of which can be formulated as MF problems. For rigid SFM, we employ the *Dinosaur* sequence[1] which contains 319 feature points tracked over 36 views, corresponding to a matrix $\mathbf{Y}_0$ of size 72×319 with 76.92% missing entries. We added uniform noise on $[-50, 50]$ to 10% randomly chosen observed entries to simulate outliers. For nonrigid SFM, we use the *Giraffe* sequence[2], which includes 166 feature points tracked over 120 frames. The data matrix $\mathbf{Y}_0$ is of size 240×166 with 30.24% missing entries. 10% of the elements were randomly chosen and added to outliers, generated from uniform distribution on $[-30, 30]$. Following the papers by Ke and Kanade (2005) and Buchanan and Fitzgibbon (2005), the rank was set to 4 and 6 for rigid and nonrigid SFM, respectively.

The performance in terms of RMSE and MAE[3] averaged over 20 runs are reported in Table 2. Similar as before, the output of each method, except SPMF methods, was chosen from 80 runs with random initializations by evaluating the objective function. It can be seen that, most of the competing methods are negatively affected by the outliers embedded in data, while our methods can still achieve reasonable approximations. Specifically, the performance of an MF method can be significantly improved using the SPL strategy. For example, the averaged RMSE of the Dinosaur sequence by PRMF, using LAD loss and L_2 regularization, is decreased from 13.205 to 3.0757 by SPMF-L_1 (L_2-reg). Besides, the

[1] `http://www.robots.ox.ac.uk/~abm/`.

[2] `http://www.robots.ox.ac.uk/~abm/`.

[3] Since the full ground truth matrix is unavailable, the RMSE and MAE were evaluated on the observed data.

Table 2: Performance comparison of 11 competing MF methods in terms of RMSE and MAE on SFM data. The results are averaged over 20 runs, and the best and the second best results are highlighted in bold with and without underline, respectively.

Method	Dinosaur		Giraffe	
	RMSE	MAE	RMSE	MAE
L2-ALM (L_2-reg)	5.4324	3.6165	1.7450	1.2768
L2-ALM (trace-reg)	5.3229	3.5916	0.7115	0.2931
DWiberg	5.4532	3.5962	2.0679	1.3566
RegL1ALM	3.8744	1.4706	0.7278	0.2929
PRMF	13.205	6.2341	0.7293	0.3749
CWM	11.114	5.1563	0.7738	0.3888
MoG	5.8979	3.6975	1.6845	1.2041
SPMF-L_2 (L_2-reg)	**1.9817**	**0.5310**	1.4511	0.6650
SPMF-L_2 (trace-reg)	2.8630	1.0125	**0.5547**	0.3576
SPMF-L_1 (L_2-reg)	3.0757	0.9810	**0.4872**	**0.2514**
SPMF-L_1 (trace-reg)	**2.2275**	**0.4714**	0.6379	**0.2748**

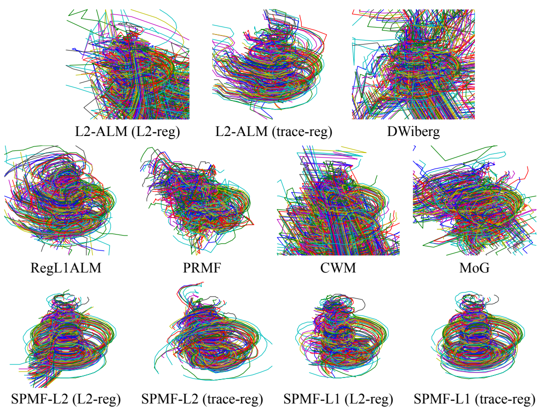

Figure 3: Recovered tracks from the Dinosaur sequence of 11 competing methods.

best performance, in terms of either RMSE or MAE, can always be achieved by the proposed methods.

We also depict typical recovered tracks of the Dinosaur sequence in Figure 3 to visualize the results. It can be observed that our methods can recover the tracks with high quality, while other methods produced comparatively more disordered results. This further substantiates the effectiveness of the proposed methods.

Background Subtraction

The background subtraction from a video sequence captured by a static camera can be modeled as a low-rank matrix analysis problem (Wright et al. 2009). Four video sequences provided by Li et al. (2004)[4] were adopted in our evaluation, including two indoor scenes (*Curtain* and *Escalator*) and two outdoor scenes (*Fountain* and *WaterSurface*). Ground truth foreground regions of 20 frames were provided for each sequence. Thus we can quantitatively compare the subtraction results using the S-measure[5] (Li et al. 2004)

[4]`http://perception.i2r.a-star.edu.sg/bk_model/bk_index`

[5]Defined as $S(A,B) = \frac{A \cap B}{A \cup B}$, where A denotes the detected region and B is the corresponding ground truth region.

Table 3: Quantitative comparison of the background results by 9 competing MF methods in terms of the S-measure. The best results are highlighted in bold.

Method	Curtain	Escalator	Fountain	WaterSurface
SVD	0.4774	0.2823	0.5170	0.2561
RegL1ALM	0.5187	0.3803	0.6296	0.2104
PRMF	0.5179	0.5581	0.7562	0.3080
CWM	0.5039	0.3877	0.7491	0.2581
MoG	0.4983	0.0531	0.5245	0.2498
SPMF-L_2 (L_2-reg)	0.4811	0.2908	0.5368	0.2611
SPMF-L_2 (trace-reg)	0.7694	0.4006	0.6480	0.5314
SPMF-L_1 (L_2-reg)	**0.8176**	**0.6049**	**0.7659**	**0.7950**
SPMF-L_1 (trace-reg)	0.6370	0.4084	0.6681	0.2694

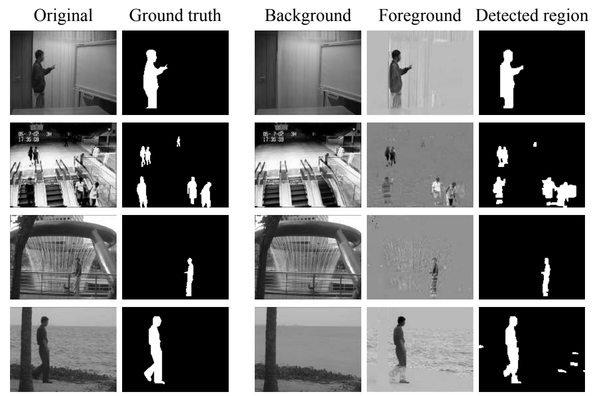

Figure 4: Background subtraction results of SPMF-L_1 (L_2-reg) on sample frames.

on these frames. To do this, we first ran an MF method on the sequence with rank-6 factorization to estimate the background. Then we applied the Markov random filed (MRF) model (Li and Singh 2009) to the absolute values of the difference between the original frame and the estimated background. This procedure can label each pixel as either foreground or background. The related optimization was solved using the well known *Graph Cut* method (Boykov, Veksler, and Zabih 2001; Kolmogorov and Zabin 2004; Boykov and Kolmogorov 2004).

We compared our methods with singular value decomposition (SVD), RegL1ALM, PRMF, CWM and MoG. We employed SVD as the representative of the L_2-norm MF methods, since it is theoretically optimal for the matrix without missing entries under the LS loss. The results are summarized in Table 3. It can be seen that, the proposed SPMF-L_1 (L_2-reg) achieves the best performance for all the four sequences, especially the WaterSurface sequence.

We also show in Figure 4 the visual results of SPMF-L_1 (L_2-reg) on some sample frames. It can be observed that, our method can reasonably separate the background and foreground, and faithfully detect the foreground region.

Conclusion

We proposed a new MF framework by incorporating the SPL methodology with traditional MF methods. This SPL manner evidently alleviates the bad local minimum issue of MF methods, especially in the presence of outliers and missing data. The effectiveness of our method for L_2- and L_1-norm MF was demonstrated by experiments on synthetic, structure from motion and background subtraction data. The proposed method shows its advantage over current MF methods on more accurately approximating the ground truth matrix from corrupted data.

Acknowledgments

This work was partially supported by the National Basic Research Program of China (973 Program) with No. 2013CB329404, the NSFC projects with No. 61373114, 11131006, 91330204, J1210059, and China Scholarship Council. It was also partially supported by the US Department of Defense, U. S. Army Research Office (W911NF-13-1-0277) and the National Science Foundation under Grant No. IIS-1251187. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARO, the National Science Foundation or the U.S. Government.

References

Basu, S., and Christensen, J. 2013. Teaching classification boundaries to humans. In *AAAI*.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *ICML*.

Boykov, Y., and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(9):1124–1137.

Boykov, Y.; Veksler, O.; and Zabih, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(11):1222–1239.

Buchanan, A., and Fitzgibbon, A. 2005. Damped Newton algorithms for matrix factorization with missing data. In *CVPR*.

Cabral, R.; De la Torre, F.; Costeira, J. P.; and Bernardino, A. 2013. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In *CVPR*.

Eriksson, A., and van den Hengel, A. 2010. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L_1 norm. In *CVPR*.

Hayakawa, H. 1994. Photometric stereo under a light source with arbitrary motion. *Journal of the Optical Society of America A* 11(11):3079–3089.

Jiang, L.; Meng, D.; Mitamura, T.; and Hauptmann, A. G. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *ACM MM*.

Jiang, Y.; Ngo, C.; and Yang, J. 2007. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR*.

Ke, Q., and Kanade, T. 2005. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*.

Kolmogorov, V., and Zabin, R. 2004. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(2):147–159.

Kumar, M.; Turki, H.; Preston, D.; and Koller, D. 2011. Learning specific-class segmentation from diverse data. In *ICCV*.

Kumar, M.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *NIPS*.

Lakshminarayanan, B.; Bouchard, G.; and Archambeau, C. 2011. Robust Bayesian matrix factorisation. In *AISTATS*.

Lee, Y., and Grauman, K. 2011. Learning the easy things first: Self-paced visual category discovery. In *CVPR*.

Li, S. Z., and Singh, S. 2009. *Markov random field modeling in image analysis*. Springer.

Li, L.; Huang, W.; Gu, I.; and Tian, Q. 2004. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing* 13(11):1459–1472.

Meng, D., and De la Torre, F. 2013. Robust matrix factorization with unknown noise. In *ICCV*.

Meng, D.; Xu, Z.; Zhang, L.; and Zhao, J. 2013. A cyclic weighted median method for L_1 low-rank matrix factorization with missing entries. In *AAAI*.

Mitra, K.; Sheorey, S.; and Chellappa, R. 2010. Large-scale matrix factorization with missing data under additional constraints. In *NIPS*.

Mnih, A., and Salakhutdinov, R. 2007. Probabilistic matrix factorization. In *NIPS*.

Ni, E., and Ling, C. 2010. Supervised learning with minimal effort. In *KDD*.

Okatani, T., and Deguchi, K. 2007. On the Wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision* 72(3):329–337.

Okatani, T.; Yoshida, T.; and Deguchi, K. 2011. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *ICCV*.

Salakhutdinov, R., and Mnih, A. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *ICML*.

Srebro, N., and Jaakkola, T. 2003. Weighted low-rank approximations. In *ICML*.

Srebro, N.; Rennie, J.; and Jaakkola, T. 2005. Maximum-margin matrix factorization. In *NIPS*.

Supančič III, J., and Ramanan, D. 2013. Self-paced learning for long-term tracking. In *CVPR*.

Tang, K.; Ramanathan, V.; Li, F.; and Koller, D. 2012. Shifting weights: Adapting object detectors from image to video. In *NIPS*.

Tomasi, C., and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision* 9(2):137–154.

Wang, Y., and Xu, H. 2012. Stability of matrix factorization for collaborative filtering. *ICML*.

Wang, N.; Yao, T.; Wang, J.; and Yeung, D. 2012. A probabilistic approach to robust matrix factorization. In *ECCV*.

Weimer, M.; Karatzoglou, A.; Le, Q.; and Smola, A. 2007. Maximum margin matrix factorization for collaborative ranking. In *NIPS*.

Wright, J.; Peng, Y.; Ma, Y.; Ganesh, A.; and Rao, S. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. In *NIPS*.

Zheng, Y.; Liu, G.; Sugimoto, S.; Yan, S.; and Okutomi, M. 2012. Practical low-rank matrix approximation under robust L_1-norm. In *CVPR*.