Matrix Completion under Gaussian Models using MAP and EM Algorithms

Gang Wu[1], Viswanathan Swaminathan[2], Ratnesh Kumar[1]
[1] Dept. of Elec. & Comp. Eng., Iowa State University, Ames, IA 50010, United States
[2] Adobe Research, Adobe Systems Inc., San Jose, CA 95110, United States
Email: wugang@iastate.edu; vishy@adobe.com; rkumar@iastate.edu

Abstract —Completing a partially-known matrix (matrix completion) is an important problem in the field of data mining and signal processing, and has been successfully applied to sensor localization and recommendation system. Low-rank and factorization models are the two most popular and successful classes of models used for matrix completion. In this paper, we investigate another approach based on statistical estimation which has previously been used for matrix completion. In an initial work involving Gaussian models (GM), the formulation was inaccurate necessitating an ad-hoc empirical diagonal loading to a covariance matrix, requiring additional tuning, and making the final estimate of model parameters difficult to interpret. An accurate formulation using a correct objective function based on likelihood estimation already exists in statistical literature, which we utilize here to learn the model parameters using an Expectation Maximization (EM) algorithm. This approach no longer needs tuning and performs better in the numerical experiments. Owing to the difference that stems from the difference in choice of objective function, we note that the original method leads to an underestimated covariance matrix necessitating an artificial diagonal loading, while the method we use provides a Maximum Likelihood (ML) estimate of the model parameters. We also validate our approach using real-world data from MovieLens, EachMovie and Netflix.

Index Terms— Matrix completion, sensor localization, recommendation system, Expectation Maximization algorithm, Gaussian model, maximum likelihood estimate

I. Introduction

The matrix completion problem has received significant attention in the signal processing community, and been successfully applied to image recovery, sensor localization, etc. [16, 17]. Also, in the field of data mining, matrix completion is used as a core technique in many recommender and prediction systems [2, 7, 12]. Matrix completion refers to recovering a matrix X from a given subset of its entries contaminated with noise. In this context, the row, column, and matrix entry can be assigned to different attributes, which leads to a wide range of applications of matrix completion. For example, in its application to the sensor localization problem, the row i and column j corresponds to sensor i and sensor j respectively, and entry $x_{i,j}$ refers to Euclidean distance between sensor i and sensor j. Such a matrix X is called a Euclidean distance matrix. In the sensor localization problem, only some of the distances $x_{i,j}$ are known, and the goal is to determine the missing distances and thus the sensor positions. In this problem, the problem size, determined by the number of sensors, is usually small, and the matrix sparsity, i.e., the ratio of given distances over all possible distances, is usually in a reasonable range. As a result, the problem can usually be solved efficiently using some traditional optimization techniques [18].

This work was supported in part by a Collaborative Research Award from Adobe and by the National Science Foundation under the Grants NSF-CCF-1331390 and NSFECCS-1509420. Authors would also like to thank Prof. Alekssandar Dognadzic of Iowa State University for providing literature information related to the completion problems.

However, in the applications of matrix completion to recommendation systems, the problem is much more ill-posed. In such applications, the matrices can be viewed as user-item preference matrices, which is usually large in size and very tall, i.e., "rows" >> "columns", and the fraction of observed entries is typically very low (1%~10%). For example in the well known Netflix Challenge, the dataset is represented with a $480{,}189 \times 17{,}770$ user by movie rating matrix, but of the possible $8{,}533$ million entries, only 100 million or 1.2%, were available for training. See Table 1 for a comparison of the matrices in the problems of sensor localization and recommendation system.

Table 1: Comparison of the matrices in the problems of sensor localization and recommendation system

	sensor localization	recommendation system
# rows	$<1{,}000$	$>100{,}000$
# columns	$<1{,}000$	$>10{,}000$
sparsity	$10\% \sim 90\%$	$1\% \sim 10\%$

To make matrix completion meaningful in such an ill-posed problem, certain assumption on X is always necessary. And the most popular one for user-item preference matrices is the approximate low-rank assumption, i.e., there exists a low-rank matrix that can approximate X with low fitting error. One interpretation of such an assumption is that, there are only a few implicit factors or features (such as representative users and movies) that determine the values in X [4]. Based on such an assumption, a number of methods using

Singular Value Decomposition (SVD) or other matrix factorization techniques have achieved good results [10, 3, 6, 11], particularly on simulated data with known rank and low level noise. However, this class of methods usually requires knowledge of the rank and the noise level, which are unknown for most real world user-item preference matrices. Therefore, extra effort for tuning is needed to find a choice of rank and regularization parameter, which is inherently determined by the noise level of the given data. This type of tuning typically uses computationally intensive grid search and/or cross-validation techniques. Wu et al. [14] showed that in real world applications, different choices of the rank and regularization parameter significantly impact the prediction accuracy. Another popular class of matrix completion methods uses matrix/tensor factorization models like PARAFAC [5]. Reference [13] proposed a method using Factorization Machines (FM) that can act as a general predictor working with any real valued feature vector and reported good results. Another class of matrix completion methods is based on certain probabilistic models of X, which is more robust against different types and levels of noise [21, 22]. Also such methods can be computationally efficient when the model parameters can be learned linearly.

In image processing, by assuming that local image patches follow Gaussian mixture models (GMM), [15] reported excellent results in a number of inverse problems, i.e., problems that involve estimating parameters or data from indirect observations. In their work, the missing data and model parameters are estimated via "coupled MAP and ML", which we refer to as C-MAP+ML (see details in Section III.B). Following the research of [15] and applying to matrix completion, [8][1] introduced a model in which every row has the same Gaussian prior distribution, and where the missing entries and model parameters are estimated together in a C-MAP+ML technique employed in [15]. However, in a subsequent work, [19] discovered that the formulation of C-MAP-ML in [15] has drawbacks and it doesn't necessarily provide a ML estimation of the model parameters, and proposed the corrected algorithm, which provides a ML estimation of the parameters. In fact a correct formulation already exists in statistical literature for a long time; see [20, ch. 11, pp. 223-225].

In this paper, we investigate the same Gaussian models for matrix completion as in [8]. But unlike their objective function, we employ the correct objective function to learn the model parameters, obtaining an exact ML estimation. MAP estimation is then used to learn the missing data, making the overall scheme MAP+ML$_{\text{EM}}$. In contrast we refer to approach of [8] as C-MAP+ML since as shown in Section III.B, [8] actually maximized the data and parameter likelihood in a coupled manner. We also make a thorough comparison between our method, MAP+ML$_{\text{EM}}$ with C-MAP+ML of [8]. As noted above, MAP+ML$_{\text{EM}}$ first appeared in [20, ch. 11, pp. 223-225], where the authors use the EM algorithm for incomplete multivariate normal samples, while to the best of our knowledge our work is a first application of this approach to the matrix completion problem. Another contribution is the successful demonstration through the application to real-world data from MovieLens, EachMovie and Netflix.

The rest of this paper is organized as follows. Section II describes the matrix completion problem and the Gaussian model for it. Section III presents MAP+ML$_{\text{EM}}$ for matrix completion, and compares it with C-MAP+ML. In Section IV, we analyze the experimental results on public movie rating datasets and also on simulated datasets. Section V concludes with some discussions and suggests a direction for potential future improvement through imposition of a practical structure on the covariance matrix of the underlying GM.

[1] Even though [15] was published in 2012 and [8] was archived in 2010, [15] was initially submitted earlier than [8], and in fact was cited in [8].

II. Matrix completion problem and the Gaussian model

A. *Matrix Completion Problem*

Matrix completion seeks to recover a same matrix $X \in \mathcal{R}^{M\times N}$ from its partial observation contaminated with noise,

$$Y = P_O(X+E), \quad (1)$$

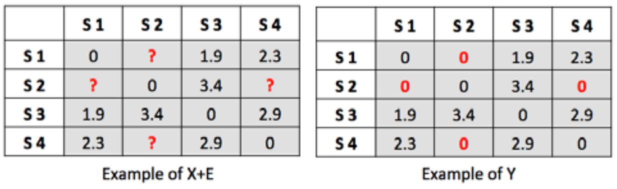
where $E \in \mathcal{R}^{M\times N}$ is a noise matrix, $O \subseteq R^{M\times N}$ is the set of indices of observed entries in X, and $P_O(\bullet)$ is a projection operator defined by,

$$P_O(Z)(i,j) = \begin{cases} Z_{i,j} & \text{if } (i,j) \in O \\ 0 & \text{if } (i,j) \notin O. \end{cases} \quad (2)$$

See Fig. 1 for an example of observed matrix $X+E$ and projected matrix Y in the sensor localization problem.

B. *The Gaussian Model*

Let X_i be the ith row of $X+E$. The Gaussian Model assumes that each X_i^T is drawn from a multivariate Gaussian distribution $\mathcal{N}\{\mu,\Sigma\}$, where μ and Σ are unknown but are the same for $i \in \{1,2,...,M\}$.



	S1	S2	S3	S4
S1	0	?	1.9	2.3
S2	?	0	3.4	?
S3	1.9	3.4	0	2.9
S4	2.3	?	2.9	0

Example of X+E

	S1	S2	S3	S4
S1	0	0	1.9	2.3
S2	0	0	3.4	0
S3	1.9	3.4	0	2.9
S4	2.3	0	2.9	0

Example of Y

Fig. 1: An example of $X+E$ and Y in the sensor localization problem

Let $X_{O,i}$ and $X_{U,i}$ denote the observed sub-vector and the missing sub-vector of X_i respectively, given by,

$$\begin{aligned} X_{O,i} &= X_i\Omega_i^T \\ X_{U,i} &= X_i\Phi_i^T \end{aligned} \qquad (3)$$

where $\Omega_i \in \mathcal{R}^{|X_{O,i}|\times |X_i|}$ is the extraction matrix that extracts the observed entries in X_i, and $\Phi_i \in \mathcal{R}^{|X_{U,i}|\times|X_i|}$ is the extraction matrix that extracts the missing entries in X_i. Note that $\Omega_i\Omega_i^T$, $\Phi_i\Phi_i^T$, and $\Omega_i^T\Omega_i + \Phi_i^T\Phi_i$ are all identity matrices, but of different dimensions. Then the following relation always holds:

$$X_i = X_{O,i}\Omega_i + X_{U,i}\Phi_i. \qquad (4)$$

Note in this model, the parameters to be estimated are $\Theta = \{\mu, \Sigma\}$.

III. Learning the Gaussian Model with EM Algorithm

A. *Proposed Method*

Based on the Gaussian Model given in Section II.A, the matrix completion problem turns to recovering $\{X_i\}$ from $\{X_{O,i}\}$. Once $\{\mu,\Sigma\}$ are estimated (we will use $\hat{\mu}$ and $\hat{\Sigma}$ denote their estimates), X_i can be estimated with its MAP estimator, which is linear and can be obtained in closed-form,

$$\begin{aligned} \{\widehat{X_i^T}\}_{MAP} &= \arg\max \log P(X_i \mid X_{O,i}, \hat{\mu}, \hat{\Sigma}) \\ &= \hat{\Sigma}\Omega_i^T(\Omega_i\hat{\Sigma}\Omega_i^T)^{-1}X_{O,i}^T + (I - \hat{\Sigma}\Omega_i^T(\Omega_i\hat{\Sigma}\Omega_i^T)^{-1}\Omega_i)\hat{\mu}. \end{aligned} \qquad (5)$$

Therefore, for matrix completion it remains to estimate $\{\mu,\Sigma\}$.

The proposed method MAP+ML$_{EM}$ for matrix completion seeks for estimates of $\{\mu,\Sigma\}$ that maximizes the likelihood $\Pi_{i=1}^{M} P(X_{O,i} \mid \mu, \Sigma)$,

$$\hat{\Theta}_{ML} = \arg\max_{\Theta} \sum_{i=1}^{M}\left(\log P(X_{O,i} \mid \Theta)\right). \qquad (6)$$

Considering the effect of missing data, EM (expectation maximization) algorithm is used here to determine ML estimate of the Gaussian parameters, by iteratively applying the following two expectation and maximization steps,

Expectation step (E-step): Calculate the expected value of the log likelihood function $\sum_{i=1}^{M}\log P(X_i \mid \Theta)$, with respect to the conditional distribution $X_{U,i} \mid X_{O,i}$ under the current parameter estimate $\hat{\Theta}^k$, i.e.,

$$Q(\Theta \mid \hat{\Theta}^k) = \sum_{i=1}^{M} E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k} \log P(X_i \mid \Theta). \qquad (7)$$

Maximization step (M-step): find the new estimates of $\{\Theta\}$ that maximizes the Q function from E-step,

$$\hat{\Theta}^{k+1} = \arg\max Q(\Theta \mid \hat{\Theta}^k). \qquad (8)$$

The value of Q function in (7), i.e., current expected value of $\sum_{i=1}^{M}\log P(X_i \mid \Theta)$, increases as the iterations of E-step and M-step continue. As the most important property of EM algorithm, increasing of the value of Q function guarantees non-decreasing of the observed data likelihood, i.e., $\Pi_{i=1}^{M} P(X_{O,i} \mid \mu, \Sigma)$ in our case. See Reference [20, ch. 8, pp. 172-173] for the proof. Therefore, when the sequence of iterations converges, $\Pi_{i=1}^{M} P(X_{O,i} \mid \mu, \Sigma)$ also reaches its maximum. The details of E-step and M-step are given below in Subsection III.A.a and III.A.b respectively.

a) *Expectation step*

The log likelihood for X_i is given by,

$$\log P(X_i \mid \Theta) = C - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(X_i - \mu^T)\Sigma^{-1}(X_i^T - \mu), \qquad (9)$$

where $C = -\frac{N}{2}\log 2\pi$ is a constant. Plugging (9) into (7), we have the Q function,

$$\begin{aligned} Q(\Theta \mid \hat{\Theta}^k) &= \sum_{i=1}^{M} E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k} \log P(X_i \mid \hat{\Theta}) \\ &= \sum_{i=1}^{M} C - \frac{\ln|\Sigma|}{2} - \frac{1}{2}E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k}[(X_i - \mu^T)\Sigma^{-1}(X_i^T - \mu)] \\ &= \sum_{i=1}^{M} C - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}tr[\Sigma^{-1}(E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k}(X_i^T - \mu) \\ &\qquad E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k}(X_i - \mu^T) + cov_{X_i\mid X_{O,i},\hat{\Theta}^k})], \end{aligned} \qquad (10)$$

where $tr[\bullet]$ is the matrix trace operator, and $cov_{X_i\mid X_{O,i},\hat{\Theta}^k}$ denotes the the covariance matrix of X_i with respect to the the conditional distribution $X_i \mid X_{O,i}, \hat{\Theta}^k$.

b) *Maximization step*

The Q function in (10) is maximized by,

$$\begin{aligned} \hat{\mu}^{k+1} &= \frac{1}{M}\sum E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k}(X_i^T) \\ \hat{\Sigma}^{k+1} &= \frac{1}{M}\sum E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k}(X_i^T - \hat{\mu}^{k+1}) \\ &\qquad E_{X_{U,i}\mid X_{O,i},\hat{\Theta}^k}(X_i^T - \hat{\mu}^{k+1})^T + cov_{X_i\mid X_{O,i},\hat{\Theta}^k}. \end{aligned} \qquad (11)$$

Now the problem is reduced to determine $\boldsymbol{P(X_{U,i} \mid X_{O,i}, \Theta)}$ and $\boldsymbol{P(X_i \mid X_{O,i}, \Theta)}$. Arranging $\boldsymbol{X_i}$ such that $\boldsymbol{X_{O,i}}$ and $\boldsymbol{X_{U,i}}$ are separated, we have,

$$\begin{bmatrix} X_{O,i}^T \\ X_{U,i}^T \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \Omega_i \mu \\ \Phi_i \mu \end{bmatrix}, \begin{bmatrix} \Omega_i \Sigma \Omega_i^T & \Omega_i \Sigma \Phi_i^T \\ \Phi_i \Sigma \Omega_i^T & \Phi_i \Sigma \Phi_i^T \end{bmatrix} \right). \quad (12)$$

From the property of multivariate Gaussian we know that $\boldsymbol{X_{U,i} \mid X_{O,i}, \Theta}$ is still a Gaussian distribution, given by:

$$\begin{aligned} X_{U,i} \mid X_{O,i}, \Theta \quad &\sim \mathcal{N}(\Phi_i \Sigma \Omega_i^T (\Omega_i \Sigma \Omega_i^T)^{-1} (X_{O,i}^T - \Omega_i \mu) + \Phi_i \mu, \\ &\quad \Phi_i \Sigma \Phi_i^T - \Phi_i \Sigma \Omega_i^T (\Omega_i \Sigma \Omega_i^T)^{-1} \Omega_i \Sigma \Phi_i^T) \\ &:= \mathcal{N}(\mu_{X_{U,i}|X_{O,i},\Theta}, cov_{X_{U,i}|X_{O,i},\mu,\Sigma}). \end{aligned} \quad (13)$$

Note that $\boldsymbol{X_i = X_{O,i}\Omega_i + X_{U,i}\Phi_i}$, and so $\boldsymbol{X_i \mid X_{O,i}, \Theta}$ is also a Gaussian distribution, given by,

$$\begin{aligned} X_i \mid X_{O,i}, \Theta \sim \mathcal{N}(\quad & \Omega_i X_{O,i}^T + \Phi_i \mu_{X_{U,i} X_{O,i},\Theta}, \\ & \Phi_i^T cov_{X_{U,i}|X_{O,i},\Theta} \Phi), \end{aligned} \quad (14)$$

where $\boldsymbol{\mu_{X_{U,i}|X_{O,i},\Theta}}$ and $\boldsymbol{cov_{X_{U,i}|X_{O,i},\Theta}}$ are given in (13).

Plugging (13) and (14) into (11), we finally get the closed-form maximizer of the $\boldsymbol{Q}$ function in (10).

B. *A Comparison with C-MAP-ML of [8]*

a) *C-MAP+ML*

C-MAP+ML seeks for the estimate that maximizes the coupled data and missing entries likelihood,

$$\begin{aligned} (\tilde{X}_i, \tilde{\Theta}) \quad &= \underset{(X_i,\Theta)}{\arg\max} \sum_{i=1}^{M} \left(\log P(X_{O,i} \mid X_i, \Theta) \right) \\ &= \underset{(X_i,\Theta)}{\arg\max} \sum_{i=1}^{M} \left(\log P(X_{O,i} \mid X_i) + \log P(X_i \mid \Theta) \right). \end{aligned} \quad (15)$$

The above optimization is computed using a pair of coupled recursions [8]:

Data Estimation step: Given previous estimate of parameters $\tilde{\Theta}^k$, each $\boldsymbol{X_i}$ is estimated with MAP estimator,

$$\begin{aligned} \tilde{X}_i^{k+1} \quad &= \underset{X_i}{\operatorname{argmax}} \sum_{i=1}^{M} \left(\log P(X_{O,i} \mid X_i) + \log P(X_i \mid \tilde{\Theta}^k) \right) \\ &= \tilde{\mu}^k + \tilde{\Sigma}^k \Omega_i^T \left(\Omega_i \tilde{\Sigma}^k \Omega_i^T \right)^{-1} (X_{O,i}^T - \Omega_i \tilde{\mu}^k). \end{aligned} \quad (16)$$

Model Parameters Estimation step: Given the updated estimate of all $\boldsymbol{X_i}$, $\boldsymbol{\Theta}$ is estimated with ML estimator,

$$\begin{aligned} \tilde{\Theta}^{k+1} &= \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^{M} \log P\left(\tilde{X}_i^{k+1} \mid \Theta \right) \\ &= \frac{1}{M} \left(\sum_{i=1}^{M} \{\tilde{X}_i^{k+1}\}^T, \sum_{i=1}^{M} (\{\tilde{X}_i^{k+1}\}^T - \tilde{\mu}^k)(\tilde{X}_i^{k+1} - \{\tilde{\mu}^k\}^T) \right). \end{aligned} \quad (17)$$

Owing to invertibility issue of the above computation that can result from $\boldsymbol{\hat{\Sigma}^k}$ being non-positive definite in (16), the authors suggest the following ad-hoc regularization of $\tilde{\Sigma}$ after each iteration:

$$\tilde{\Sigma}^{k+1} \leftarrow \tilde{\Sigma}^{k+1} + \varepsilon I, \quad (18)$$

where $\boldsymbol{\varepsilon}$ is an ad-hoc tuning parameter. In Section III.B.b, we shed light on why this method suffers from invertability issues.

b) *MAP+ML$_{EM}$ vs. C-MAP+ML*

The fundamental differences between MAP+ML$_{EM}$ and C-MAP+ML are their different choices of objective function to maximize, and the ways they handle missing data. MAP+ML$_{EM}$ seeks for estimate of $\boldsymbol{\{\mu, \Sigma\}}$ that maximizes the observed data likelihood. To eliminate the effects of the missing data, in each iteration of MAP+ ML$_{EM}$, expectation with respect to missing-data, conditioned under observed data and current estimate of model parameters, is performed. In contrast, C-MAP+ML seeks for estimates of $\boldsymbol{\{X_i, \Theta\}}$ that maximizes the likelihood over both Gaussian model parameters $\boldsymbol{\Theta}$ and the missing data, i.e.,

$$\prod_{i=1}^{M} P(X_{o,i} \mid X_{u,i}, \Theta). \quad (19)$$

The iterations in C-MAP+ML can be viewed as a coordinate descent optimization of the likelihood in (19). However, this approach of treating both model parameters and missing data as parameters is flawed (see for example [9]).

To illustrate the deficiency of C-MAP+ML, we give here an example of a special case of the matrix completion problem. Consider first the simpler case where the matrix of interests $\boldsymbol{X}$ has only one column, so that $\boldsymbol{\Theta}$ is a partially observed univariate Gaussian. Without loss of generality, we assume that the first $\boldsymbol{m}$ entries are observed, and the remaining $\boldsymbol{M - m}$ are missing. MAP+ML$_{EM}$ gives the ML estimation, $\boldsymbol{\hat{\mu}_{ML} = \sum_{i=1}^{m} \frac{X_i}{m}}$, and $\boldsymbol{\hat{\Sigma}_{ML} = \sum_{i=1}^{m} \frac{(X_i - \hat{\mu}_{ML})^2}{m}}$. In contrast, C-MAP+ML gives the estimates $\tilde{\mu} = \hat{\mu}_{ML}$, and $\tilde{\Sigma} = \frac{m}{M} \hat{\Sigma}_{ML}$, in which while ML estimation of $\boldsymbol{\mu}$ is obtained, but $\boldsymbol{\Sigma}$ is proportionally underestimated.

In the general case of multiple columns in $\boldsymbol{X}$, we cannot analytically compare the two types of estimates of $\boldsymbol{\Theta}$. Yet we have similar observations as in the univariate case on the estimate of $\boldsymbol{\Theta}$ in each iteration. We rewrite the iterative estimation of Θ in C-MAP+ML in the following equivalent form:

$$\begin{aligned}
\overset{\sim}{\mu}^{k+1} &= \frac{1}{M}\sum_{i-1}^{M} E_{X_{U,i}|X_{O,i},\tilde{\Theta}^{k}}(X_i^T) \\
\overset{\sim}{\Sigma}^{k+1} &= \frac{1}{M}\sum_{i-1}^{M} E_{X_{U,i}|X_{O,i},\tilde{\Theta}^{k}}(X_i^T - \overset{\sim}{\mu}^{k+1}) \qquad (20) \\
& E_{X_{U,i}X_{O,i},\tilde{\Theta}^{k}}(X_i^T - \overset{\sim}{\mu}^{k-1})^T .
\end{aligned}$$

Then comparing (20) with (11) we can see that, in each iteration of both MAP+ML$_{\text{EM}}$ and C-MAP+ML, $\boldsymbol{\mu}$ is estimated as the average of expectation of X_i under the conditional distribution $X_{U,i}\,|\,X_{O,i},\Theta^{k}$. But the estimation of Σ is different: In C-MAP+ML it is estimated as the corresponding sample covariance matrix, whereas Σ in MAP+ML$_{\text{EM}}$ contains an additional term $\frac{1}{M}\sum_{i=1}^{M} cov_{X_i|X_{O,i},\hat{\Theta}^{k}}$. The absence of this additional term indicates underestimation of Σ in C-MAP+ML, similar to the univariate case. This also explains the inevitability issue witnessed in C-MAP+ML and the necessarity for the proposed ad-hoc diagonal loading in (18). Also note that while the expression for $\boldsymbol{\mu}$is same in (20) vs. (11), its dependence on Σ renders the computed $\boldsymbol{\mu}$values to differ.

To summarize the differences between C-MAP+ML and MAP+ML$_{\text{EM}}$, the diagonal loading in (18) is introduced to circumvent the inevitability issue arising due to underestimation in (20), and also makes the estimate of the Gaussian model parameters ad-hoc and different from the ML estimate. Also the tuning effort in adjusting the loading parameter εmakes the computation more time-consuming. In contrast, MAP+ML$_{\text{EM}}$ provides ML estimation of parameters, and is computationally more tractable as it doesn't require any tuning.

IV. Numerical Experiments

A. *Experiments with movie rating datasets*

In this experiment, we test C-MAP+ML and MAP+ML$_{\text{EM}}$ on three public movie rating datasets: MovieLens, EachMovie, and Netflix. The size of data sets is as follows: MovieLens: 6,040×3,592, with 95.42% sparsity; EachMovie: 18,328×811, with 91.85% sparsity; Netflix: 46,905×600, with 74.5% sparsity. To show the applicability of Gaussian model on the datasets, we also include results from two other popular and successful methods for matrix completion, viz., SoftImpute [10], as a representative of low-rank models based methods, and FM [13], as a representative of factorization models based methods. The experiments are setup in the following way. For the SoftImpute method, the regularization parameter λ(see [10] for details) is tuned with a 10-candidate grid search. For FM, we use the configuration recommended by the author [23]. For C-MAP+ML, the regularization parameter ε is tuned with K-5 cross-validation and a 10-candidate grid search. No tuning is needed for MAP+ML$_{\text{EM}}$. Following the standard in Netflix Challenge [1], the performance of the methods is measured by the root mean square error (RMSE).

The results of the four different methods on the public movie rating datasets are given in Table 2. We see that MAP+ML$_{\text{EM}}$ performs slightly better than C-MAP+ML, but more significantly compared to the other two methods of SoftImpute and FM. See Table 3 for the detailed improvements. This also validates the applicability of Gaussian models to the public movie rating datasets.

Results using C-MAP+ML requires tuning of parameter ε by grid search and cross-validation. Such tuning efforts are expensive but necessary for C-MAP+ML owing to the sensitivity to the tuning parameter, as can be seen in Fig. 2 for case of EachMovie dataset, and Fig. 3 for case of Netflix dataset.

Time performance of the four different methods on the public movie rating datasets is provided in Table 4. FM method is the fastest, while SoftImpute is the slowest (except for the case of Netflix). As expected, owing to the exemption from tuning, MAP+ML$_{\text{EM}}$ consumes less time than C-MAP-ML on all three datasets: 61.16% faster on MovieLens, 96.28% faster on EachMovie, and 239.0% faster on Netflix.

Table 2: Comparison based on RMSE of SoftImpute, FM, C-MAP+ML and MAP+MLEM on public movie rating datasets.

	MovieLens	EachMovie	Netflix
SoftImpute	0.9180	1.0853	0.8480
FM	0.8648	1.0916	0.8459
C-MAP+ML	0.9151	1.0783	0.8441
MAP+ML$_{\text{EM}}$	0.9129	1.0704	0.8419

Table 3: MAP+MLEM’s improvements over SoftImpute, FM, and C-MAP+ML on public movie rating datasets, using MAP+MLEM’s RMSE as base.

	MovieLens	EachMovie	Netflix
over SoftImpute	0.5587%	1.3920%	0.7246%
over FM	-5.2689%	1.9806%	0.4751%
over C-MAP+ML	0.2410%	0.7380%	0.2613%

Table 4: Time performance (tuning process included) of SoftImpute, FM, C-MAP+ML and MAP+MLEM on public movie rating datasets, measured in minutes.

	MovieLens	EachMovie	Netflix
SoftImpute	263.1	133.2	113.7
FM	2.2	1.3	6.5
C-MAP+ML	127.8	47.5	197.3
MAP+ML$_{\text{EM}}$	79.3	24.2	58.2

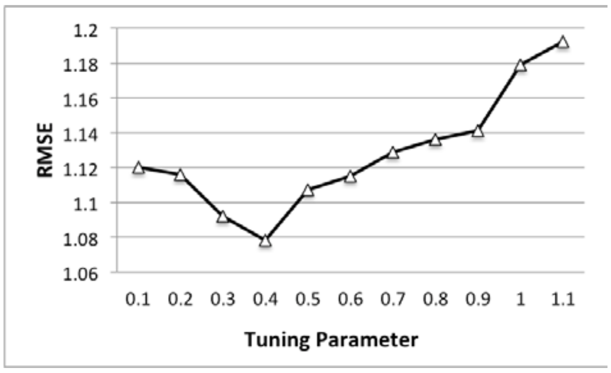


Fig. 2: C-MAP+ML's RMSE against tuning parameter on EachMovie data.

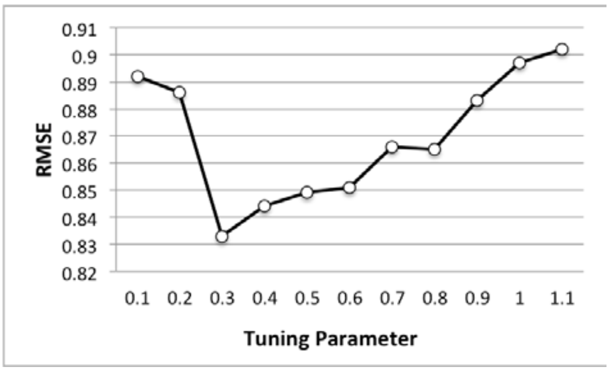


Fig. 3: C-MAP+ML's RMSE against tuning parameter on Netflix data.

B. Experiments with Simulated Data

We simulated a dataset of size $10{,}000 \times 20$, with covariance matrix $\Sigma = W_{20\times3}W_{20\times3}^T + 0.1^2 I_{20\times20}$, and mean vector μ randomly drawn from a uniform distribution in the range 1 to 5. $38{,}000$ entries were randomly picked to make up the training set, and $2{,}000$ entries were used as the test set. With such a simulated dataset generated from a known underlying model, we are able to provide a more comprehensive comparison between the MAP+ML$_{EM}$ and C-MAP+ML: First, we can compare the two methods' RMSEs in estimation of $\{\mu, \Sigma\}$, which measures their capabilities of recovering the underlying Gaussian model; secondly, we can construct an artificial estimator, which uses the exact $\{\mu, \Sigma\}$ for estimation. In the experiment, we also use an estimator 'col' as a baseline, which uses column-wise sample mean and variance as the estimates.

Results of the experiment with the simulated data are provided in Table 5. Regarding prediction on the test data, similar to the results on the public datasets, MAP+ML$_{EM}$ performs slightly better than C-MAP+ML and slightly worse than the 'exact' estimate. Regarding recovering the underlying model parameters, MAP+ML$_{EM}$ and C-MAP+ML report similar RMSE on estimation of μ. But MAP+ML$_{EM}$ significantly outperforms C-MAP+ML on estimation of Σ, reporting a 78.18% improvement (using the former as base). This implies that MAP+ML$_{EM}$ provides a better estimation of the parameters of the underlying Gaussian model.

Table 5: Comparison based on RMSE on simulated dataset, where estimator 'exact' uses true μ and Σ for estimation, and 'col' uses column-wise sample mean and variance for estimation. Running of C-MAP+ML takes 113 seconds, while MAP+MLEM takes 87 seconds.

	C-MAP+ML	MAP+ML$_{EM}$	exact	col
RMSE on test	0.8754	0.8663	0.8618	1.767
RMSE on μ	0.0504	0.0225	0	*NA*
RMSE on Σ	0.3877	0.0846	0	*NA*

V. Conclusions and discussion

In this work, we investigated matrix completion under Gaussian models as source for their values, and learned the underlying Gaussian models using the ML$_{EM}$ (maximum likelihood based on expectation maximization) algorithm. This is then followed by a MAP computation to estimate the missing data. Compared to the existing method of (C-MAP+ML), our method (MAP+ML$_{EM}$) requires no tuning. This is a significant improvement as tuning is ad-hoc and computationally expensive, affecting the time performance. We also provide a thorough comparison between MAP+ML$_{EM}$ and C-MAP+ML and formally point out the underlying fundamental flaw with C-MAP+ML in form of a missing additive term in the estimation of the co-variance matrix, resulting in its underestimation and hence the associated inevitability problem.

In the experiments on public movie ratings datasets, MAP+ML$_{EM}$ achieves better results than C-MAP+ML when compared using RSME as well as the computational time. The results also show improvements compared to results from low-rank model based methods as well as factorization methods. This also validates the use of Gaussian model for the public movie rating datasets.

For the estimates to be useful, the number of observations must significantly outnumber the number of parameters $\frac{N(N+1)}{2} + N$. When this is violated, there is a risk of over-fitting. Future work can be done in exploring more structured Gaussian models requiring less parameters which can still serve as a good model for the data.

References

[1] R. M. Bell and Y. Koren. "Lessons from the netflix prize challenge." *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[2] P. Chen, C. Tsai, Y. Chen, K. Chou, C. Li, C. Tsai, K. Wu, Y. Chou, C. Li, W. Lin, et al. "A linear ensemble of

individual and blended models for music rating prediction." *KDDCup*, 2011.

[3] M. Fazel. "Matrix rank minimization with applications." PhD diss., PhD thesis, Stanford University, 2002.

[4] A. Feuerverger, Y. He, S. Khatri, et al. "Statistical significance of the netflix challenge." *Statistical Science*, 27(2):202–231, 2012.

[5] R. A. Harshman. "Foundations of the parafac procedure: Model and conditions for an"explanatory"multi-mode factor analysis." In UCLA Working Papers in, 1969.

[6] R. H. Keshavan, A. Montanari, and S. Oh. "Matrix completion from a few entries." *Information Theory*, IEEE Transactions on, 56(6):2980–2998, 2010.

[7] Y. Koren, R. Bell, and C. Volinsky. "Matrix factorization techniques for recommender systems." *Computer*, (8):30–37, 2009.

[8] F. Léger, G. Yu, and G. Sapiro. "Efficient matrix completion with gaussian models." *arXiv preprint arXiv*:1010.4050, 2010.

[9] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.

[10] R. Mazumder, T. Hastie, and R. Tibshirani. "Spectral regularization algorithms for learning large incomplete matrices." *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[11] B. Recht, M. Fazel, and P. A. Parrilo. "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization." *SIAM review*, 52(3):471–501, 2010.

[12] B. Recht and C. Ré. "Parallel stochastic gradient algorithms for large-scale matrix completion." *Mathematical Programming Computation*, 5(2):201–226, 2013.

[13] S. Rendle. "Factorization machines with libfm." *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[14] G. Wu, V. Swaminathan, S. Mitra, and R. Kumar. "Online video session progress prediction using low-rank matrix completion." In *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference* on, pages 1–6. IEEE, 2014.

[15] G. Yu, G. Sapiro, and S. Mallat. "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. Image Processing." *IEEE Transactions* on, 21(5):2481–2499, 2012.

[16] Zhou, Xiaowei, C., Zhao, H., & Yu, W. "Low-rank modeling and its applications in image analysis." *ACM Computing Surveys (CSUR)* 47.2 (2015): 36.

[17] Biswas, Pratik, et al. "Semidefinite programming based algorithms for sensor network localization." *ACM Transactions on Sensor Networks (TOSN)* 2.2 (2006): 188-220.

[18] Fang, Haw-ren, and Dianne P. O'Leary. "Euclidean distance matrix completion problems." *Optimization Methods and Software* 27.4-5 (2012): 695-717.

[19] Yang, J., Liao, X., Yuan, X., Llull, P., Brady, D. J., Sapiro, G., & Carin, L. (2015). Compressive Sensing by Learning a Gaussian Mixture Model From Measurements. IEEE Transactions on Image Processing, 24(1), 106-119.

[20] Roderick J. A. Little and Donald B. Rubin, *Statistical Analysis with Missing Data*, 2nd ed., Wiley, 2002.

[21] Lawrence, Neil D., and Raquel Urtasun. "Non-linear matrix factorization with Gaussian processes." In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 601-608. ACM, 2009.

[22] Zhou, Mingyuan, Chunping Wang, Minhua Chen, John Paisley, David Dunson, and Lawrence Carin. "Nonparametric Bayesian matrix completion." *Proc. IEEE SAM 2* (2010): 12.

[23] Steffen Rendle. "libFM 1.4.2-Manual." [Online]. Spetember 14, 2014. Available: http://www.libfm.org/libfm-1.42.manual.pdf

Gang Wu was born in China, in 1989. He received the B.S. degree from Dalian University of Technology (DLUT), China, in 2011, majored in electrical engineering. He is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, Iowa State University, in USA. Since September of 2012, he has also been working with Adobe Research on the research of online video recommendation engine. His research interests include matrix completion, recommendation system, statistical signal processing, and machine learning.

Viswanathan (Vishy) Swaminathan is a principal scientist in Adobe Research working on next generation video technologies. His areas of research include video streaming, analytics, processing, coding, and protection. Vishy received his MS and Ph.D. in electrical engineering from Utah State University. He received his B.E (ECE) degree from College of Engineering, Guindy, Anna University, Chennai. Vishy has authored several papers, articles, RFCs, and book chapters, has several issued patents, and has been invited to talk at multiple conferences on video systems related topics. Vishy received 3 certificates from ISO for his contribution to MPEG Standards.

Ratnesh Kumar is a Professor with the Iowa State University since 2002, and previously with the University of Kentucky from 1991 to 2002, in electrical and computer engineering at both places. Prof. Kumar has held visiting positions with the University of Maryland, the Applied Research Laboratory (at Penn State University), NASA Ames, the Idaho National Laboratory, the United Technologies Research Center, and Air Force Research Laboratory. He received the B. Tech. degree in electrical engineering from IIT Kanpur, India, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Texas, Austin, TX, USA, in 1987, 1989, and 1991, respectively. He received the Gold Medals for the

Best EE Undergrad and the Best All Rounder from IIT Kanpur, and the Best Dissertation Award from UT Austin. He is a Fellow of the IEEE, a Distinguished Lecturer for the IEEE Control Systems Society, Best Paper Award recipient from IEEE Transactions on Automation Science and Engineering, and Best Paper Finalist at the IEEE ICNSC (International Conf. on Networking, Sensors and Controls) as well as IEEE Sensors.