Active multiple matrix completion with adaptive confidence sets

Andrea Locatelli
OvGU Magdeburg

Alexandra Carpentier
OvGU Magdeburg

Michal Valko
SequeL team, Inria Lille

Abstract

In this work, we formulate a new multi-task active learning setting in which the learner's goal is to solve multiple matrix completion problems simultaneously. At each round, the learner can choose from which matrix it receives a sample from an entry drawn uniformly at random. Our main practical motivation is *market segmentation*, where the matrices represent different regions with different preferences of the customers. The challenge in this setting is that each of the matrices can be of a different size and also of a different rank which is unknown. We provide and analyze a new algorithm, MALocate that is able to adapt to the unknown ranks of the different matrices. We then give a lower-bound showing that our strategy is minimax-optimal and demonstrate its performance with synthetic experiments.

1 Introduction

In this work, we consider the setting of completing multiple matrices in a sequential and active way, under a budget constraint on the number of observations the learner may request. The learner's objective is to estimate each of these matrices well (in some precise sense that we define later) and is akin to the *pure exploration* problems considered in the multi-armed bandits (Bubeck et al., 2011; Gabillon et al., 2011). As the learner is trying to solve multiple learning problems simultaneously, a decent strategy should naturally allocate a larger portion of the observational budget to harder problems. Such challenge is for example considered in a very different model by Riquelme et al. (2017). Of course, since knowing the hardness or *complexity* of each instance is typically out of reach in practice, a good strategy should be *adaptive* to the different complexity scenarios, without requiring any tuning. This is in contrast with previous results for regret minimization with a low-rank structure (Katariya et al., 2017b,a), where the learner explicitly takes advantage of the rank-1 structure of the setting.

We consider matrix completion in the *trace-regression model* (Klopp, 2014; Rohde and Tsybakov, 2011; Koltchinskii et al., 2011; Negahban and Wainwright, 2012). There are important reasons regarding this choice as opposed to the *Bernoulli model* (Candès and Recht, 2009; Chatterjee, 2015), another common model for the matrix completion. In particular, in the trace-regression model it is possible that some of the matrix entries are sampled multiple times. In the Bernoulli model, this cannot happen, as each entry is observed either never or once with probability p in the simplest model. The implication of this *multi-sampling* is fundamental as it allows, in the trace-regression model, to construct honest confidence sets that *adapt to the rank* of the matrix, even if the level of noise is unknown. On the other hand, it has been shown that in the Bernoulli model such confidence sets provably do not exist (Carpentier et al., 2017). This is very important, as we will see that our adaptive strategy crucially depends on the existence of these adaptive confidence sets: Consider for example the problem of minimizing the maximum of the losses across multiple matrix completion problems. A good strategy should roughly equalize the diameter of the confidence sets across instances when the budget expires, as it pays the price for the largest diameter by definition of the maximum loss. In order to do that, it is important to leverage adaptive confidence sets.

The main application domain we target is market segmentation (Wedel and Kamakura, 2000) and polling. However, being able to multi-sample decides the situations where exactly this model applies. For example, for music recommendations in music streaming services, it is possible that the users listen to the same song twice or more and we can get multiple samples of their appreciations, either by rating or by not-skipping. For movie or product ratings, multi-sampling is much less

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

applicable. Yet it possible to ask the customer for a second opinion later in time. In other situations, the multi-sampling happens by design. For example, in tasting experiments, the human subjects are sometimes given same two samples, that they have to taste and evaluate with a week-long break in between. Our algorithm and results apply to these situations, whether the multiples-sample for the same entry are possible because of the nature of the setting or by design.

In this work, we introduce the active multiple matrix completion problem and propose an anytime algorithm (MALocate) that solves this problem *adaptively* to the unknown ranks of each sub-problem. For the max loss, which corresponds to the case where the learner pays the price of the largest loss on the set of matrix completion problems it has to solve, we show that our strategy is optimal by deriving a matching lower bound. Finally, we show that MALocate indeed performs well with a synthetic experiment.

2 Multiple matrix completion setting

We start by defining the single matrix completion problem and state the known results that we build on. Then, we introduce our active setting, which can be thought of as solving K matrix completion problems simultaneously (as the objective is to optimize the loss when the budget n expires) and sequentially as we may decide where to allocate our budget at round $t \leq n$.

2.1 Single matrix completion setting

We first introduce the matrix completion setting and a matrix lasso estimator. Let $\mathbf{M}_0 \in \mathbb{R}^{d_1 \times d_2}$ be an unknown matrix. The task of matrix completion is that of estimating $\mathbf{M}_0$ accurately in some precise sense, that we define later, by an estimator $\widehat{\mathbf{M}}$ given n independent random pairs $(\mathbf{X}_i, Y_i)_{i \leq n}$ such that

$$Y_i \triangleq \operatorname{Tr}(\mathbf{X}_i^\intercal \mathbf{M}_0) + \sigma \varepsilon_i,$$

where the ε_i are centered independent random variables with unit variance.[1] We consider the matrix completion setting where $\mathbf{X}_i^\intercal$ are i.i.d. uniformly distributed on the set

$$\mathcal{X} \triangleq \left\{ e_i(d_1) e_j^\intercal(d_2), i \in [d_1], j \in [d_2] \right\},$$

where $e_i(d)$ are the canonical basis vectors in $\mathbb{R}^d$. Typically, in this setting, we do not observe the entire matrix of size $d_1 \times d_2$ as we have $n \ll d_1 d_2$, and we consider matrices of low rank r, with respect to $\min(d_1, d_2)$, for which completion is still possible despite the low number of observations. Let $d \triangleq \max(d_1, d_2)$ and $\|\mathbf{M}\|_F$ is the Frobenius norm of a matrix $\mathbf{M} = (\mathbf{M}_{ij}) \in \mathbb{R}^{d_1 \times d_2}$ defined as

$$\|\mathbf{M}\|_F^2 \triangleq \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \mathbf{M}_{ij}^2 = \operatorname{Tr}(\mathbf{M}^\intercal \mathbf{M}).$$

For this problem, it is possible to construct good estimators $\widehat{\mathbf{M}}_n$ such that

$$\frac{\|\widehat{\mathbf{M}}_n - \mathbf{M}_0\|_F^2}{d_1 d_2} \leq \rho(r, n, d),$$

where $\rho(r, n, d) \ll \|\mathbf{M}_0\|_\infty$ for $r \ll \min(d_1, d_2)$ and $n \geq rd$. Intuitively, the higher the rank r of $\mathbf{M}$, the harder the problem should be, as there are more parameters to estimate. A good estimator should be *adaptive* to the rank of the matrix without requiring it as an input to allow the tuning of hyperparameters.

2.2 Square-root lasso estimator

In this work, we consider the matrix square-root lasso estimator, which has been shown to have favorable properties (Candès and Tao, 2006; Klopp, 2014; Gaïffas and Lecué, 2011; Koltchinskii et al., 2011). We define the nuclear norm of a matrix

$$\|\mathbf{M}\|_\star \triangleq \operatorname{Tr}\left(\sqrt{\mathbf{M}^\intercal \mathbf{M}}\right) = \sum_{i=1}^{r} \sigma_i,$$

where σ_i are the singular values of $\mathbf{M}$. The matrix square-root lasso estimator is defined as

$$\widehat{\mathbf{M}}_n(\lambda) \in \operatorname*{arg\,min}_{\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}} \left\{ \sqrt{\sum_{i=1}^{n} \frac{(Y_i - \langle \mathbf{X}_i, \mathbf{M} \rangle)^2}{n}} + \lambda \|\mathbf{M}\|_\star \right\}. \quad (1)$$

Importantly, for this estimator Klopp (2014) showed that

$$\rho(r, n, d) = \mathcal{O}\left(\frac{rd \log d}{n}\right)$$

for λ defined in the following proposition, that *does not depend* on r, the unknown rank of matrix $\mathbf{M}$. It also does not require the variance σ^2 of the noise as an input to tune λ, only an upper bound such that $A \geq \sigma$.

Proposition 1 (upper bound, Klopp, 2014)**.** *There exist numerical constants c and C such that with probability at least $1 - 3/d - 2\exp(-cn)$, the matrix square-root lasso estimator $\widehat{\mathbf{M}}_n$ satisfies*

$$\frac{\|\widehat{\mathbf{M}}_n - \mathbf{M}\|_F^2}{d^2} \leq \frac{CA^2 \cdot rd \log d}{n},$$

where $\widehat{\mathbf{M}}_n$ is defined as the solution to the minimization problem in Equation 1 with $\lambda \triangleq C' A \sqrt{(\log d)/(nd)}$ where C' is a numerical constant.

[1] In this paper, we will restrict ourselves to the case of bounded noise, but our results can be extended to sub-exponential noise as in the work of Klopp (2014).

We also restate a lower bound for the single matrix completion problem shown by Koltchinskii et al. (2011, Theorem 5), which shows that the previous procedure is minimax optimal up to an extra $\log d_k$ factor.

Proposition 2 (lower bound, Koltchinskii et al., 2011)**.** *For any estimation procedure that outputs $\widehat{\mathbf{M}}_n$ from n noisy observations corrupted with independent noise $\varepsilon_t \sim \mathcal{N}(0, A^2)$, there exists a matrix $\mathbf{M}$ of size $(d \times d)$ and rank at most r such that*

$$\mathbb{E}\left[\frac{\|\widehat{\mathbf{M}}_n - \mathbf{M}\|_F^2}{d_1 d_2}\right] \geq \frac{cA^2 rd}{n},$$

where c is a small numerical constant and the expectation is taken with respect to both the distribution of the samples and the possible internal randomization of the estimation procedure.

This result easily extends to the bounded noise case.

2.3 Adaptive confidence sets

An important theoretical result in the trace-regression model with uniform sampling of the entries is the existence of *adaptive and honest* confidence bands on the error $||\widehat{\mathbf{M}} - \mathbf{M}||_F^2$. Importantly, the knowledge of σ is again not necessary for this estimator. This procedure, `EstimateError`, is described in Section 3, and makes use of the entries X_i that have been observed twice to compute an unbiased estimator of the error. This procedure comes with the following guarantee.

Proposition 3 (concentration bound for $\widehat{R}_N$ estimator, Carpentier et al., 2017)**.** *Let N be the number of entries that have been observed twice in the second half of the sample and $\widehat{R}_N$ be the (unbiased) estimation procedure (sub-procedure* `EstimateError`*)* of $\|\widehat{\mathbf{M}} - \mathbf{M}\|_F^2$, for some $\widehat{\mathbf{M}}$. Then with probability at least $1 - \frac{2}{d}$, we have

$$\left|\widehat{R_N} - \frac{\left\|\widehat{\mathbf{M}} - \mathbf{M}\right\|_F^2}{d^2}\right| \leq 8A^2\sqrt{\frac{\log d}{N}}.$$

For minimax-optimal estimation procedures, such as the square-root lasso, we can show (by bounding both the estimation error as above and $N \geq Cn^2/d^2$ for some numerical constant C, on a favorable event) that with high probability,

$$\widehat{R_N} + 8A^2\sqrt{\frac{\log d}{N}} \leq \mathcal{O}\left(\frac{rd\log d}{n}\right),$$

which shows that this quantity is an *adaptive* (as it does not require the rank as an input) and *honest* (as it upper bounds the true error with high probability) confidence band on $\|\widehat{\mathbf{M}} - \mathbf{M}\|_F^2$.

2.4 Active multiple matrix completion

In the active multiple matrix completion, the learner's goal is to complete multiple matrices $\{\mathbf{M}^k\}_k$ simultaneously, by *actively choosing* from which matrix it should ask for a new observation in a sequential and adaptive manner. For ease of notation, we restrict this setting to square matrices of dimension d_k, but our techniques directly extend to non-square matrices. At each round the active learner has to choose an action $k_t \in [K]$ and receives a pair $(\mathbf{X}_t^{k_t}, Y_t^{k_t})$ such that $\mathbf{X}_t^{k_t}$ corresponds to the location of the *entry* $(i_{k_t,t}, j_{k_t,t})$ of the k_t-th data matrix $\mathbf{M}^{k_t} = (\mathbf{M}_{ij}^{k_t}) \in \mathbb{R}^{d_{k_t} \times d_{k_t}}$ chosen uniformly at random such that $i_{k_t,t} \in [d_{k_t}]$ and $j_{k_t,t} \in [d_{k_t}]$, and

$$\begin{aligned} Y_t^{k_t} &\triangleq \mathrm{Tr}(e_{i_{k_t,t}}(d_{k_t}) e_{j_{k_t,t}}^\intercal(d_{k_t}) \mathbf{M}^{k_t}) + \varepsilon_{k_t,t} \\ &= \mathbf{M}_{i_{k_t,t} j_{k_t,t}} + \varepsilon_{k_t,t}, \end{aligned}$$

where the $e_i(d)$ are the canonical basis vectors of $\mathbb{R}^d$. Here, $\mathbf{X}_t^{k_t} = e_{i_{k_t,t}}(d_{k_t}) e_{j_{k_t,t}}^\intercal(d_{k_t})$. Informally, the learner chooses to observe one of the K matrices, and receives a noisy observation of one of the entries (corrupted by $\varepsilon_{k_t,t}$) chosen uniformly at random from that matrix. The goal of the learner is to adaptively choose which matrix $\mathbf{M}^{k_t}$ to sample based on the observations collected so far up to round $t-1$,

$$\left\{(\mathbf{X}_1^{k_1}, Y_1^{k_1}), \ldots, (\mathbf{X}_{t-1}^{k_{t-1}}, Y_{t-1}^{k_{t-1}})\right\}.$$

At the end of the game, once it has collected at most n pairs $(\mathbf{X}_t^{k_t}, Y_t^{k_t})$, the learner has to output estimates $\widehat{\mathbf{M}}_n^k$ of each matrix $\mathbf{M}^k$ to suffer the following loss,

$$\mathcal{L}_n^p \triangleq \left(\sum_{k \in [K]} \|\widehat{\mathbf{M}}_n^k - \mathbf{M}^k\|_F^{2p}\right)^{1/p},$$

where p characterizes the objective and is decided as part by the learner at the start of the game. As special and interesting cases, for $p = 1$, we recover the unnormalized squared Frobenius norm if the sub-problems were the blocks of a block-diagonal matrix, and for $p = \infty$ the max loss $\max_{k \in [K]} \|\widehat{\mathbf{M}}_n^k - \mathbf{M}^k\|_F^2$.

Remark 1. *As an extension, we can consider the reweighted loss, characterized by a given weight vector $\mathbf{w} = (w_1, ..., w_K)$, where $w_i \in \mathbb{R}^+$ for $i \in [K]$ is a parameter given to the learner along with p,*

$$\mathcal{L}_n^p(\mathbf{w}) = \left(\sum_{k=1}^K w_k \|\widehat{\mathbf{M}}_n^k - \mathbf{M}^k\|_F^{2p}\right)^{1/p}.$$

Taking $w_k = d_k^{-2}$ allows to consider the normalized Frobenius norm for each matrix, which is particularly interesting in combination with $p = \infty$ as it is simply the maximum average loss per entry within each sub-problem, regardless of the dimension.

For each matrix $\mathbf{M}_k$, $k \in [K]$, we denote by r_k, the rank of $\mathbf{M}^k$. We further assume that all the observations $Y_t^{k_t}$ and the entries of $\mathbf{M}^k$ are bounded by some known constant A. The first condition is $|Y_t^k| \leq A$ for any k, t and the second condition is simply $\|\mathbf{M}^k\|_\infty \leq A$. This is a mild assumption in applications such as recommendation systems, where ratings are bounded.

3 MALocate algorithm

We now describe our active strategy MALocate for the active multiple matrix completion given as Algorithm 1. The input for MALocate is the maximum budget input n and the loss parameter p. This parameter defines which loss $\mathcal{L}_n^p$ the strategy is should optimize for. We shall see that p governs the exploration. During the initialization, while $B_k(t) = \infty$, the strategy requests for each $\mathbf{M}^k$ a dataset $\mathcal{D}_t^k$ of size $\mathcal{O}(d_k \log d_k)$. MALocate uses the requested samples for two goals: computing the estimators *and* adaptively estimating their error. In particular, the first half of the requested sample is used to compute an estimator $\widehat{\mathbf{M}}_t^k$ of $\mathbf{M}^k$ using the square-root lasso estimator. The second half of the sample is used by the EstimateError $(\widehat{\mathbf{M}}_t^k, \mathcal{D}_t^k)$ sub-procedure to construct an estimator of the error $\widehat{R}_{N_k^t}$ and an *upper-bound* on this error $B_k(t)$, using the *double-sampled* entries. After the initialization, at round t, the strategy allocates the next samples to the matrix

$$m \triangleq \arg\max_k d_k^2 B_k(t) T_k(t)^{-1/p},$$

where $T_k(t)$ is the number of samples allocated to matrix k up to round t. The previous estimator $\widehat{\mathbf{M}}^m$ for matrix m is then replaced by $\widehat{\mathbf{M}}_t^m$ *only if* the upper bound on the error has decreased. The strategy operates on a *doubling schedule*: Each round an index m is chosen, a new dataset $\mathcal{D}_t^m$ of size $T_m(t)$ (and thus, a total budget of $2T_m(t)$ is spent on m) is used to construct a new estimator $\widehat{\mathbf{M}}_t^m$, and estimate its error.

In this case, $B_m(t)$ is also updated to the new (smaller) upper bound on the error. This ensures that the estimation error is *non-increasing* with t for every matrix. This is a crucial ingredient for the proof of Theorem 1, which characterizes the performance of MALocate. The loop is repeated until the budget has been used, at which point the algorithm stops and outputs estimator $\widehat{\mathbf{M}}^k$ for each matrix k.

Algorithm 1 MALocate algorithm

```
Input: $n$, $\{d_k\}_{k\in[K]}$, $p$ {loss parameter}
$\mathcal{D}_t^k \leftarrow \emptyset \quad \forall k \in [K]$
Initialization:
  $t \leftarrow 0$
  for $k \in [K]$ do
    $T_k(t) \leftarrow 0$
    $B_k(t) \leftarrow \infty$
  end for
while $t \leq n$ do
  $m \leftarrow \arg\max_{k\in[K]} d_k^2 B_k(t) T_k(t)^{-1/p}$
  $T_m \leftarrow \max(T_m(t), 4\lceil(d_k \log(d_k) + 1)/2\rceil)$
  $t \leftarrow t + T_m$
  $T_m(t) \leftarrow T_m(t) + T_m$
  $\mathcal{D}_t^m \leftarrow$ NewSamples$(m, T_m)$
  $\widehat{\mathbf{M}}_t^m \leftarrow$ GetEstimator$(m, \mathcal{D}_t^m)$
  $N_t^m, \widehat{R}_{N_t^m} \leftarrow$ EstimateError$(\widehat{\mathbf{M}}_t^m, \mathcal{D}_t^m)$
  $B_k(t) \leftarrow B_k(t - T_m) \quad \forall k \in [K]$
  if $\widehat{R}_{N_m^t} + 8A^2\sqrt{\log(d_m)/N_t^m} \leq B_m(t)$ then
    $\widehat{\mathbf{M}}^m \leftarrow \widehat{\mathbf{M}}_t^m$
    $B_m(t) \leftarrow \widehat{R}_{N_t^m} + 8A^2\sqrt{\log(d_m)/N_t^m}$
  end if
  $T_k(t) \leftarrow T_k(t - T_m) \quad \forall k \neq m$
end while
Output: $\{\widehat{\mathbf{M}}^k\}_{k\in[K]}$
```

Algorithm 2 NewSamples (k, T)

```
Input: $k, T$
  Sample uniformly at random
    $T$ new observations $\{(X_i, Y_i)\}_{i\leq T}$ from $\mathbf{M}^k$
Output: New dataset $\{(X_i, Y_i)\}_{i\leq T}$
```

Computing the estimator As explained previously, we use the square-root lasso estimator. Notice that we perform a splitting of the sample $\mathcal{D}_t^k$, where the first half is used to compute the estimator, and the second half is used to estimate its error. In practice, we propose instead to split the sample between entries that have been sampled only once to compute the estimator, and the other entries to estimate the error. While this introduces a small dependence (as we may only estimate the error for entries on which the estimator was not trained) which is difficult to analyze, in practice, this greatly improves the power of the estimator.

Estimating the error The sub-procedure EstimateError uses the second half of a dataset $\mathcal{D}_t^k$ to build an estimator of the error for some estimator $\widehat{\mathbf{M}}^k$ of the matrix $\mathbf{M}^k$. It proceeds as the estimator of Carpentier et al. (2017) by finding entries (X_i, Y_i) and (X_j, Y_j) such that $X_i = X_j$ to form the triplet (X_i, Y_i, Y_j), and the dataset $\mathcal{D}'$ of double-sampled entries with $N_t^k \triangleq |\mathcal{D}'|$. $\mathcal{D}'$ is then used to compute the unbiased estimator of the error,

$$\widehat{R_N} \triangleq \frac{1}{N}\sum_{i=1}^{N}\Big(Y_i - \langle X_i, \widehat{\mathbf{M}}\rangle\Big)\Big(Y_i' - \langle X_i, \widehat{\mathbf{M}}\rangle\Big),$$

which *does not require the variance* of the noise as an input to the estimation procedure. We can then deduce an upper bound on $\widehat{R_N}$ that holds with high probability $B_k(t) \triangleq \widehat{R_{N_t^k}} + 8A^2\sqrt{\log(d_k)/N_t^k}$. Importantly, this upper bound on the error is *honest* and *adaptive* to the unknown rank r_k as proved by Carpentier et al. (2017) and is upper bounded as $\mathcal{O}\big(r_k d_k^3 \log(d_k)/T_k(t)\big)$, as $\widehat{R_{N_t^k}}$ dominates the stochastic error term.

Algorithm 3 GetEstimator $(k, \mathcal{D})$

```
Input: $k, \mathcal{D}$
  $T \leftarrow |\mathcal{D}|/2, \lambda \leftarrow C\sqrt{\log(d_k)/d_k T}$
  $\widehat{\mathbf{M}} \leftarrow \underset{\|\mathbf{M}\|_\infty \le A}{\arg\min} \sqrt{\frac{1}{T}\sum_{i=1}^T (Y - \langle X_i, \mathbf{M}\rangle)^2} + \lambda\|\mathbf{M}\|_\star$
Output: Estimator $\widehat{\mathbf{M}}$
```

The sampling criterion The exploration crucially depends on the interplay between the loss parameter p, $T_k(t)$, and the upper bound on the error $B_k(t)$ rescaled by d_k^2. For $p = 1$ (sum loss), the chosen index is

$$\arg\max_k d_k^2 B_k(t) T_k(t)^{-1},$$

and can be interpreted as the index that maximizes the error per sample, which is a rough approximation of $\partial B_k(t)/\partial T_k(t)$. The idea behind this heuristic is that since we expect the sum loss to decrease the most for this matrix, the next sample is allocated to this index. On the other hand, for $p = \infty$, the index chosen is simply the one that currently suffers the largest upper bound on the rescaled error.

Algorithm 4 EstimateError $(\widehat{\mathbf{M}}, \mathcal{D})$

```
Input: $\widehat{\mathbf{M}}, \mathcal{D}$
  $T = |\mathcal{D}|/2$
  Find double-sampled entries
    $\mathcal{D}' \leftarrow \{(X_i, Y_i, Y_i')\}_{i=1,\dots,N}$ in $\mathcal{D}_{T+1,\dots,2T}$
  $\widehat{R_N} \leftarrow \frac{1}{N}\sum_{i=1}^N \Big(Y_i - \langle X_i, \widehat{\mathbf{M}}\rangle\Big)\Big(Y_i' - \langle X_i, \widehat{\mathbf{M}}\rangle\Big)$
Output: Number of double-sampled entries $N$ and
        error estimate $\widehat{R_N}$
```

More generally, by plugging the upper bound given by Proposition 1 into the loss $\mathcal{L}_n^p$, we see that a good allocation is one that minimizes

$$\sum_k \left(\frac{r_k d_k^3 \log d_k}{T_k(n)}\right)^p$$

under the constraint $\sum_k T_k(n) = n$. By solving the corresponding optimization problem, we see that this good allocation should be such that

$$T_k(n)^{1+1/p} = (r_k d_k^3 \log d_k) C(n),$$

where $C(n)$ is constant for all k. Note however, that this good allocation is de facto out of reach for the learner, which does not have access to the underlying ranks $\{r_k\}_{k\in[K]}$ of the matrices. Now, as $d_k^2 B_k(t)$ can be upper bounded as $\mathcal{O}\big(r_k d_k^3 \log d_k / T_k(t)\big)$, it is clear that our strategy, which picks the index that maximizes $d_k^2 B_k(t) T_k(t)^{-1/p}$ mimics the good allocation that keeps the quantity

$$r_k d_k^3 \log(d_k) T_k(n)^{-(1+1/p)}$$

constant across the arms.

Remark 2. *An important algorithmic particularity of our strategy is that it operates on a doubling schedule. Namely, when index k is picked, the number of observations for $\mathbf{M}^k$ is doubled from $T_k(t)$ to $2T_k(t)$, as a new dataset of size $T_k(t)$ is generated. This allows us to analyze* MALocate *without considering correlations between the different estimators, as each estimator is trained on a fresh sample $\mathcal{D}_t^k$. This also has the benefit of greatly reducing the computational complexity, as we only need to train a logarithmic number of estimators, while recomputing estimators at each round t would be too costly. However, if there is an empirical need to recalculate the estimator every round we received a new observation, the proofs for the guarantee that we provide in the next section can be modified to reflect it.*

4 Analysis

In this section, we give guarantees on the performance of MALocate for general p, and prove a lower bound in the case $p = \infty$, showing that our strategy is optimal for the max loss, up to logarithmic factors.

4.1 Upper bound on the loss of MALocate

We start with upper bounding the loss of MALocate that holds with high probability.

Theorem 1. *After n sample requests,* MALocate *started with loss parameter p outputs K estimators, such that with probability at least $1 - \sum_k 16\log(d_k)/d_k$,*

$$\mathcal{L}_n^p \triangleq \left(\sum_{k\in[K]} \left\|\widehat{\mathbf{M}}_n^k - \mathbf{M}^k\right\|_F^{2p}\right)^{1/p}$$
$$\le \mathcal{O}\left(\frac{\left(\sum_{k=1}^K (r_k d_k^3 \log d_k)^{\frac{p}{p+1}}\right)^{\frac{p+1}{p}}}{n}\right).$$

We prove this result in Appendix A. It relies on a careful bounding of the estimation error of $\widehat{\mathbf{M}}_n$ directly, as it is *not possible*[2] to prove bounds on $T_k(n)$, the number

[2] For example, if one of the estimators $\widehat{\mathbf{M}}^k$ is *by chance* very good despite having been given few samples, then it is possible that it will not be given more samples.

of times that each arm has been sampled at the end of the horizon, as opposed to many regret analyses used for bandit settings. In particular, the proof proceeds by showing that the following bounds on the error hold with high probability. First, using the sampling criterion we prove that for all k a bound of the form

$$\begin{aligned}&\left\|\widehat{\mathbf{M}}^k - \mathbf{M}^k\right\|_F^2\\ &\quad\leq \mathcal{O}\left(T_k(n)^{\frac{1}{p}}\left(\sum_k (r_k d_k^3 \log d_k)^{\frac{p}{p+1)}}\right)^{\frac{p+1}{p}} n^{-\frac{p+1}{p}}\right).\end{aligned}$$

Importantly, this *grows* with $T_k(n)$. On other hand, Proposition 1 yields that

$$\left\|\widehat{\mathbf{M}}^k - \mathbf{M}^k\right\|_F^2 \leq \mathcal{O}\left(\frac{r_k d_k^3 \log d_k}{T_k(n)}\right),$$

which *decreases* with $T_k(n)$. By balancing both bounds with respect to $T_k(n)$, we get an upper bound on the estimation error that does not depend on $T_k(n)$.

This result shows that the complexity of the problem crucially depends on the interaction between both the intrinsic difficulty of each sub-problem associated with $\mathbf{M}_k$, characterized by r_k and d_k, and the loss parameter p. Namely, if we set

$$c_k \triangleq r_k d_k^3 \log d_k$$

for the *complexity* of problem k, and $\mathbf{c} = (c_1, \ldots, c_K)$, then the complexity of the active problem is $\|\mathbf{c}\|_{\frac{p}{p+1}}$ i.e., the loss is upper bounded as

$$\mathcal{O}\left(\|\mathbf{c}\|_{\frac{p}{p+1}} n^{-1}\right).$$

On the other hand, it is easy to see that the uniform strategy suffers a loss of order $\frac{K}{n}\|\mathbf{c}\|_p$, which is always larger[3] than $\frac{1}{n}\|\mathbf{c}\|_{\frac{p}{p+1}}$. This shows that our active strategy, `MALocate`, adapts on-the-fly to the difficulty of the problem at hand, without requiring any input parameter that depends on this complexity.

We now rewrite the previous theorem for the important case $p = \infty$.

Corollary 1. *(upper bound for max loss) After n sample requests, `MALocate` started with loss parameter $p = \infty$ outputs K estimators, such that with probability at least $1 - \sum_k 16 \log(d_k)/d_k$,*

$$\max_{k\in[K]} \left\|\widehat{\mathbf{M}}_n^k - \mathbf{M}^k\right\|_F^2 \leq \mathcal{O}\left(\frac{\sum_{k=1}^K r_k d_k^3 \log d_k}{n}\right).$$

This result is a direct corollary of our main upper bound. It shows that interestingly, even in the case $p = \infty$, the complexity of each *individual* problem comes into play. Namely, in this setting, the total complexity is simply the sum of the complexities for each sub-problem.

Remark 3. *While our results are stated in the fixed-budget setting, our strategy can easily be adapted to the (ε, δ)-correct setting, by slightly modifying the estimators, in particular by replacing $\log d_k$ terms by $\log(1/\delta)$ and re-deriving the bounds on their performance. The sample complexity would be of order $\widetilde{\mathcal{O}}(\|c\|_{\frac{p}{p+1}} \varepsilon^{-1})$. Interestingly, in this setting, it is also possible to design a stopping rule, as we have adaptive confidence bands on the $\widehat{\varepsilon}_t$, the error at round t.*

4.2 Lower bound

We now show a lower bound for the active multiple matrix completion problem in the case $p = \infty$. The offline part of our lower bound proof is inspired by Koltchinskii et al. (2011). The challenge of our proof is the active setting as we have to consider strategies that may actively spread their observations over the different matrices.

Theorem 2. *For any active strategy $\mathcal{S}$, there exists a problem $P = (\mathbf{M}^1, \ldots, \mathbf{M}^K)$, where $\mathbf{M}^k$ is of rank at most r_k and dimension $(d_k \times d_k)$, such that after $\mathcal{S}$ (actively) collects at most n observations corrupted with $\mathcal{N}(0, A^2)$ noise and outputs K estimators $(\widehat{\mathbf{M}}^1, \ldots, \widehat{\mathbf{M}}^K)$, we have*

$$\mathbb{E}_{P,\mathcal{S}}\left[\max_{k\in[K]}\left(\left\|\widehat{\mathbf{M}}^k - \mathbf{M}^k\right\|_F^2\right)\right] \geq \frac{A^2}{2048}\frac{\sum_{k=1}^K r_k d_k^3}{n}.$$

We prove this theorem in Appendix B. The main argument is that for any active strategy $\mathcal{S}$, for any fixed problem P, there exists one index $m \in [K]$ such that

$$\mathbb{E}_{P,\mathcal{S}}[T_k(n)] \leq \frac{r_m d_m^3}{\sum_k r_k d_k^3} n.$$

Then, we carefully adapt the arguments of the lower bound for $K = 1$ to our active setting.

This shows that our active strategy is minimax-optimal (up to logarithmic factors) over the class of problems with dimension $\{d_k\}_{k\in[K]}$ and ranks at most $\{r_k\}_{k\in[K]}$, fully adaptive to the unknown ranks of the sub-problems. Importantly, the lower bound also holds for strategies that have apriori knowledge of $\{r_k\}_{k\in[K]}$.

Remark 4. *Notice, that while Algorithm 3 uses a particular square-root lasso estimator with associated guarantees, our approach straightforwardly extends to other estimators. For example, Klopp (2015) provides sharp bounds in the Bernoulli model, i.e., without the extra $\log d_k$ factor. Therefore, this or any other result,*

[3] as we have $\|\mathbf{x}\|_{q_1} \leq K^{1/q_1 - 1/q_2}\|\mathbf{x}\|_{q_2}$ for $0 < q_1 < q_2$

that provides a sharper estimator could be used instead in Algorithm 3. This would improve the overall complexity of our active strategy by removing the extraneous $\log d_k$ *factors in the complexity, matching exactly the lower bound for* $p = \infty$.

5 Synthetic experiments

We now support our analysis `MALocate` with synthetic experiments. To create a square matrix of rank r and dimension d, we generate two matrices $\mathbf{U} \in \mathbb{R}^{d\times r}$ and $\mathbf{V} \in \mathbb{R}^{r\times d}$ with entries distributed as $\mathcal{N}(0, \sigma_r^2 \triangleq r^{-1/2})$. The standard deviation σ_r is chosen such that the entries of $\mathbf{M} = \mathbf{UV}$ have the same scaling, regardless of the rank of the matrix. Observations are corrupted with Gaussian white noise $\mathcal{N}(0, \sigma \triangleq 0.1)$. We consider both objectives $\mathcal{L}_p$ for $p = 1$ and $p = \infty$, on which we run `MALocate` also with both parameters $p = 1$ and $p = \infty$. We also compare `MALocate` to the naïve uniform strategy, and for the max loss also with the oracle strategy that has access to the true Frobenius error of the estimators and allocates the next samples to the index $\arg\max_k \|\widehat{\mathbf{M}}_t^k - \mathbf{M}\|_F^2$. Note that this strategy (for a fixed estimation procedure) is optimal for $p = \infty$, as the max loss may only decrease if the worst estimator is improved.

As our goal is to study the active advantage of `MALocate`, all the strategies have access to the same estimator `SoftImpute`, tuned with the same parameters. Moreover, we discretize time in a similar fashion for all the strategies: The initialization phase of each estimator is done with $8d_k$ samples and after that, the budget is divided evenly in approximately 100 sub-samples. This allows to bypass the negative effects associated with a doubling schedule. As our strategy is naturally anytime, we plot the results as the time horizon grows from the initialization up to $n = Kd^2/2$. At each round t where a new estimator has been trained, we use the knowledge of $\mathbf{M}^k$ to compute $\mathcal{L}_t^p$ for $p \in \{1, \infty\}$. For both experiments, we draw and fix the problem, and average the results over 15 runs.

5.1 First experiment

We fix $d_k \triangleq d \triangleq 200$, $K \triangleq 10$, and the ranks are such that $r_k \triangleq 10$ for all k besides $r_1 = 40$. We choose this instance as it forces the strategy into a tradeoff with respect to the loss parameter p. Heuristically, to optimize the sum loss ($p = 1$), reaching a good error on each of the easy problems is very important. On the other hand, to optimize the max loss, it is necessary to spend a large portion of the budget on the hardest instance. In Figure 1, we see that our strategies perform favorably in the setting they are designed for. We also see that the uniform strategy only catches up when the number of samples is high enough such that the careful sample allocation has little effect on the performance.

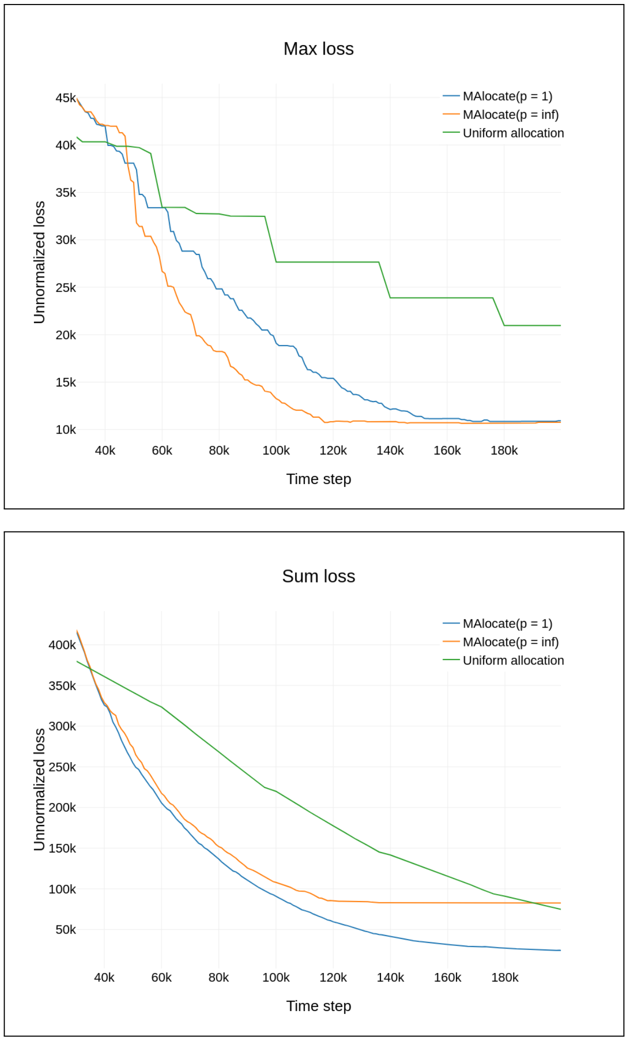

Figure 1: Results for the first experiment

5.2 Second experiment

We fix $d_k \triangleq d \triangleq 200$ and $K \triangleq 15$. The ranks r_k are given by $r_k \triangleq 18 + 0.0015k^4$. Note that the hardest instance is such that $r_{15} = 76$ and half of the subproblems have rank at most 22. This set of problems is more varied than the previous one and shows the adaptivity of our strategy (Figure 2).

5.3 Implementation of `MALocate`

As we discuss in Remark 4, our generic strategy can be used for *any estimator*, which may be chosen appropriately with respect to the exact noise setting. For performance reasons, we used the `SoftImpute` estimator (Mazumder et al., 2010) from the python package **`fancyimpute`**, which we tweak to have a warm-start

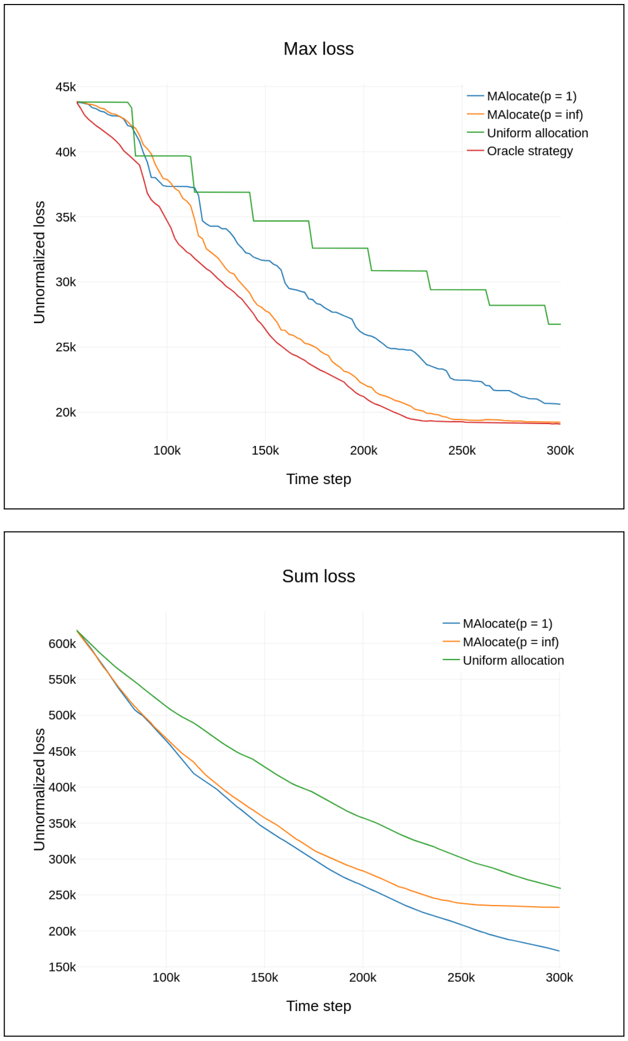

Figure 2: Results for the second experiment

heuristic that fills missing entries with the previous estimator $\widehat{\mathbf{M}}^k$. This allows us to speed-up the running time. More generally, online matrix completion results such as the ones by Dhanjal et al. (2014); Lois and Vaswani (2015); Jin et al. (2016) fit in our active and sequential framework. We tune the confidence intervals on the error in a conservative way. As we use a time discretization instead of a geometric grid, we also re-use samples throughout the run. Finally, as explained in Section 3, instead of splitting the entire sample, we use entries that have been observed once to train the estimator, and the other entries (sampled at least twice) to estimate the error.

Across the experiments, we see that MALocate run with the proper loss parameter p indeed performs better on the associated loss $\mathcal{L}^p$. For the max loss, we also see that MALocate with $p = \infty$ performs only slightly worse than the optimal oracle strategy in this setting. On the other hand, the uniform strategy performs poorly across the problems. We see that for the max loss, the loss peters out when the hardest matrix to estimate has been sampled d_k^2 times, as we cap the number of observations for each matrix to d_k^2. We remark however that we are interested in settings with smaller $n \ll K d_k^2$, where we see that MALocate with $p = \infty$ performs very favorably.

6 Conclusion and discussion

We presented a new active matrix completion setting and provided MALocate, an active strategy that is able to adapt to the different complexities of the problems and proved that up to log factors, it achieves minimax-optimal guarantees. We also showed that empirically, it performs in accordance with its theoretical guarantees for two loss settings. We our work as the first step towards a more systematic understanding of the links between adaptive confidence sets (in any statistical setup) and the corresponding active learning setting.

We considered the *high-dimensional* regime where the number of samples n satisfies $d \leq n \ll d^2$. The number of doubly-sampled entries scales (w.h.p.) by Proposition 6 as n^2/d^2 for any n in this interval. This remains true for $n \gg d^2$ and generally our results would also hold in this regime. However, we do not address this case here at all, as from an algorithmic point of view, much simpler estimation strategies solve this problem, for example, least squares with a projection on the set of rank r matrices coupled with Lepski's method to adapt to the rank) .

Finally it is, unfortunately, not possible to extend our approach to datasets where entries are not observed twice, because it is *provably impossible* to obtain a good estimator of the error.

Acknowledgements The research presented was supported by European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, Inria and Otto-von-Guericke-Universität Magdeburg associated-team north-European project Allocate, and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01) and BoB (n.ANR-16-CE23-0003). The work of A. Carpentier is also partially supported by the Deutsche Forschungsgemeinschaft (DFG) Emmy Noether grant MuSyAD (CA 1488/1-1), by the DFG - 314838170, GRK 2297 MathCoRe, by the DFG GRK 2433 DAEDALUS, by the DFG CRC 1294 Data Assimilation, Project A03, and by the UFA-DFH through the French-German Doktorandenkolleg CDFA 01-18. This research has also benefited from the support of the FMJH Program PGMO and from the support to this program from Criteo.

References

Bubeck, S., Munos, R., and Stoltz, G. (2011). Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772.

Candès, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425.

Carpentier, A., Klopp, O., Löffler, M., and Nickl, R. (2017). Adaptive confidence sets for matrix completion. *Bernoulli*.

Castro, R. M. and Nowak, R. D. (2008). Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *Annals of Statistics*, 43(1):177–214.

Dhanjal, C., Gaudel, R., and Clémençon, S. (2014). Online matrix completion through nuclear norm regularisation. In *International Conference on Data Mining*.

Gabillon, V., Ghavamzadeh, M., Lazaric, A., and Bubeck, S. (2011). Multi-bandit best arm identification. In *Neural Information Processing Systems*.

Gaïffas, S. and Lecué, G. (2011). Sharp oracle inequalities for high-dimensional matrix prediction. *IEEE Transactions on Information Theory*, 57(10):6942–6957.

Gilbert, E. N. (1952). A comparison of signalling alphabets. *Bell System Technical Journal*, 31(3):504–522.

Jin, C., Kakade, S. M., and Netrapalli, P. (2016). Provable efficient online matrix completion via nonconvex stochastic gradient descent. In *Neural Information Processing Systems*.

Katariya, S., Kveton, B., Szepesvári, C., Vernade, C., and Wen, Z. (2017a). Bernoulli rank-1 bandits for click feedback. In *International Joint Conference on Artificial Intelligence*.

Katariya, S., Kveton, B., Szepesvári, C., Vernade, C., and Wen, Z. (2017b). Stochastic rank-1 bandits. In *International Conference on Artificial Intelligence and Statistics*.

Klopp, O. (2014). Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*.

Klopp, O. (2015). Matrix completion by singular value thresholding: Sharp bounds. *Electronic journal of statistics*, 9(2):2348–2369.

Koltchinskii, V., Lounici, K., and Tsybakov, A. B. (2011). Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329.

Lois, B. and Vaswani, N. (2015). Online Matrix Completion and Online Robust PCA. In *IEEE International Symposium on Information Theory*.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322.

Negahban, S. and Wainwright, M. J. (2012). Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697.

Riquelme, C., Ghavamzadeh, M., and Lazaric, A. (2017). Active learning for accurate estimation of linear models. In *International Conference on Machine Learning*.

Rohde, A. and Tsybakov, A. B. (2011). Estimation of high-dimensional low-rank matrices. *Annals of Statistics*, 39(2):887–930.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, New York, NY.

Varshamov, R. R. (1957). Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741.

Wedel, M. and Kamakura, W. A. (2000). *Market segmentation : Conceptual and methodological foundations*. Springer US.