# AN EFFICIENT ONLINE ADAPTIVE SAMPLING STRATEGY FOR MATRIX COMPLETION

*Lucas Claude, Symeon Chouvardas and Moez Draief*

Mathematical and Algorithmic Sciences Lab, Huawei Technologies France

## ABSTRACT

Matrix Completion (MC) i.e., estimating the missing values of an unknown matrix based on limited information, has been a prominent topic of study in the last decades. In this paper, we focus more precisely on a little-known aspect of MC, namely how the recommendation of new entries can help improve the accuracy of the reconstruction. We present an efficient online algorithm to solve the MC task, and propose an Adaptive Sampling under Smoothness Assumption (AdSSA) strategy, which is suitable for operation on smooth low-rank matrices. This technique is able to predict iteratively which entries are the most informative. Our numerical examples illustrate that AdSSA performs significantly better than the Uniform Random Sampling (URS) and the Query by Committee (QbC). In addition, AdSSA algorithm can be straightforwardly implemented in an online and efficient manner, which constitutes another advantage.

***Index Terms*—** matrix completion, adaptive sampling, online algorithms.

## 1. INTRODUCTION AND RELATED WORK

Matrix Completion has attracted considerable interest in the last decades. This problem occurs in many scenarios, such as : recommender systems, image inpainting, network monitoring, video denoising. The exact MC problem has been treated successfully in the seminal work of [1]. This work proposes a very interesting relaxation of the classical (non-convex, NP-hard) rank minimization problem into a (convex) nuclear norm minimization problem, which has been widely used since. Following this breakthrough, many methods have been proposed to complete matrices when all the training data become available simultaneously. Among them we find mainly: spectral methods relying on SVD [2, 3], and matrix factorization methods [4]. Moreover, models are now able to incorporate more and more complex constraints: affine constraints [5], graph constraints [6], interval uncertainty [7], to name a few.

With the recent and spectacular increase in data volumes, there has been an important focus on tackling the MC problem in an online manner, i.e., updating the estimate when some matrix coefficients, that were previously unknown, are sequentially revealed, so as to avoid doing all the computations from scratch. The most common approach considers that the columns of the matrix to be completed arrive one by one, from left to right, with missing entries. Then, the task mainly consists in online subspace estimation. Numerous papers took this approach to solving this problem. These were initiated by different communities : online dictionary learning [8], online PCA [9], online non-negative matrix factorization [10, 11], just to name a few. However, when the new uncovered entries are randomly located, the online problem becomes more difficult and can't be treated with the previous column-by-column approach. [12] develops an algorithm that addresses this problem with a systematic update of the rank $k$ decomposition.

Another practical topic of interest in the study of MC concerns the sampling strategy [13]. In most real life applications, we have the possibility to recommend entries that will be observed next, e.g., in collaborative filtering [12] or wireless network coverage maps [14]. So far it remains rather unclear how the choice of those entries can influence the accuracy and help getting a better reconstruction of the ground truth matrix. Previous work proposed a technique named Query by Commitee (QbC), which utilizes different models to find the predicted entries that present the highest degree of disagreement and recommend them next [15]. A drawback of this method is that it is computationally demanding.

The main contribution of this work, the focus of which is adaptive sampling for online matrix completion, is twofold. First, we propose a novel adaptive sampling rule. With the same number of new entries, this method outperforms Uniform Random Sampling (URS) and QbC in terms of accuracy of the reconstruction. Unlike previous methods [14] it does not require the simultaneous training of several models, and thus allows for real-time recommendations. We show experimentally the benefit of this approach for the *coverage map reconstruction problem*. The second contribution is that we properly reformulate the algorithm presented in [16] so that to be able to operate in an online fashion.

The remainder of this paper is organized as follows. In Section 2 we introduce the notations and the online matrix completion algorithm. Section 3 is devoted to the presentation and justification of the adaptive sampling strategy. Section 4 presents the numerical experiments: we describe the proto-

**Table 1**. Alternating Least Squares for Matrix Completion

**Initialize:**

$\Omega$ observed entries

$\mathbf{X}_\Omega$ values of observed entries

$\mathbf{U} \in \mathcal{M}_{m,k}(\mathbb{R})$ randomly initialized

**Repeat:**

**1:** fix $\mathbf{U}$ and update

$\mathbf{V} = \arg\min_{\mathbf{V}} \sum_{(i,j)\in\Omega} \left(\mathrm{X}_{ij} - \mathbf{u}_i\mathbf{v}_j^T\right)^2 + \gamma \sum_j \|\mathbf{v}_j\|_2^2$

**2:** fix $\mathbf{V}$ and update

$\mathbf{U} = \arg\min_{\mathbf{U}} \sum_{(i,j)\in\Omega} \left(\mathrm{X}_{ij} - \mathbf{u}_i\mathbf{v}_j^T\right)^2 + \gamma \sum_i \|\mathbf{u}_i\|_2^2$

col used to conduct our simulations, and the results. Finally, Section 5 concludes the paper and discusses future work.

## 2. ONLINE MATRIX FACTORIZATION

### 2.1. Problem statement

As stated in the introduction, matrix completion is a well studied problem and many formulations have been proposed in the literature to address it. First let's introduce some notations. Throughout the paper, $\mathcal{M}_{m,n}(\mathbb{R})$ is the set of matrices of size $m \times n$ with real coefficients. We consider a matrix $\mathbf{X} \in \mathcal{M}_{m,n}(\mathbb{R})$ and denote by $X_{ij}$ the coefficient in the $i^{\text{th}}$ row and $j^{\text{th}}$ column. The set of observed entries (the mask) is denoted by $\Omega$. We adopt the matrix factorization framework: our aim is to find a good low rank decomposition $\mathbf{U}\mathbf{V}^T$ that approximates $\mathbf{X}$, where $\mathbf{U} \in \mathcal{M}_{m,k}(\mathbb{R})$ and $\mathbf{V} \in \mathcal{M}_{n,k}(\mathbb{R})$. In addition, let $\mathbf{u}_i$ (resp. $\mathbf{v}_j$) be the $i^{\text{th}}$ row of $\mathbf{U}$ (resp. $j^{\text{th}}$ row of $\mathbf{V}$). More precisely we consider the following objective:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} \left(\mathrm{X}_{ij} - \mathbf{u}_i\mathbf{v}_j^T\right)^2 + \gamma \left(\sum_i \|\mathbf{u}_i\|_2^2 + \sum_j \|\mathbf{v}_j\|_2^2\right). \quad (1)$$

with $\gamma$ being a regularization parameter that controls the tradeoff between the closeness to data and the nuclear norm of the reconstructed matrix [4]. Note that $k$ is not known in advance and has to be fixed arbitrarily. In practice a grid search is used to find a value that meets simultaneously good accuracy ($k$ rather large) and computational efficiency ($k$ rather small).

A common approach to solve (1) is Stochastic Gradient Descent (SGD), popularized by [17] in the context of the Netflix prize. It is well known to be a fast algorithm in large scale settings. However, as pointed out in [16], it requires a tedious tuning of the learning rate. To avoid this, we resort to another method called Alternating Least Squares (ALS) [18].

### 2.2. Alternating Least Squares

Alternating Least Squares is a two-step iterative method, described in Table 1. If we study more carefully the first step (the second step follows the same logic), we have to solve a quadratic program :

$$\min_{\mathbf{V}} \sum_j \left( \sum_{i\ st.(i,j)\in\Omega} \left(\mathrm{X}_{ij} - \mathbf{u}_i\mathbf{v}_j^T\right)^2 + \gamma\|\mathbf{v}_j\|_2^2 \right). \quad (2)$$

This amounts to solving $n$ least-squares problems to find the optimum solution for each row of $\mathbf{V}$:

$$\mathbf{v}_p = \left( \sum_{i\ st.(i,p)\in\Omega} \left(\mathbf{u}_i^T\mathbf{u}_i + \gamma\mathbf{I}_k\right) \right)^{-1} \sum_{i\ st.(i,p)\in\Omega} \mathrm{X}_{ip}\mathbf{u}_i \quad (3)$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix. The computation of each $\mathbf{v}_p$ is dominated by the matrix inversion, hence a global cost of $O(nk^3)$ operations. This may become prohibitive if the parameter $k$ is large. However [16] proposed an approximate accelerated method : instead of computing the optimum vector $\mathbf{v}_p$ directly, we update its coefficients one by one. A single update only consists in putting the derivative of (2) with respect to $\mathrm{V}_{pl}$ to zero and is given by :

$$\mathrm{V}_{pl} = \frac{\sum_{i\ st.(i,p)\in\Omega} \left(\sum_{m\neq l} \mathrm{U}_{im}\mathrm{V}_{pm} - \mathrm{X}_{ip}\right) \mathrm{U}_{il}}{\sum_{i\ st.(i,p)\in\Omega} \mathrm{U}_{ip}^2 + \gamma}. \quad (4)$$

The expression for $\mathrm{U}_{p'l'}$ is similar. Although it is an approximation of the optimal solution given by (3), it reduces the overall computation time of one main loop to $O(|\Omega|\bar{\mu}k)$ where $\bar{\mu}$ is the average number of samples per row/column. For sparse datasets the improvement in terms of speedup is significant, while preserving good estimation of the optimum according to [16]. Most of the time, ALS is terminated when a given number of iterations of the main loop is reached (e.g., 1000).

### 2.3. Online version

Suppose that we have already found the optimum solution to equation (1) : $\mathbf{U}^*, \mathbf{V}^*$. If we observe a new entry $(i_0, j_0)$, recomputing the MC from scratch is by no means efficient. Instead, it is quite intuitive that the new solution should be close to $\mathbf{U}^*, \mathbf{V}^*$. Moreover, equation (4) shows that, with our model, in first approximation, i.e., a single pass update over the coefficients of $\mathbf{U}$ and $\mathbf{V}$, only vectors $\mathbf{u}_{i_0}$ and $\mathbf{v}_{j_0}$ should be impacted by the observation of the new entry $\mathrm{X}_{i_0 j_0}$. This idea is very popular among the recommender systems community when it comes to efficient online updates [12]. Thus, in our experiments to come, each time a new entry is observed, we perform the online update as presented in Table 2.

**Table 2**. Online Update

**Initialize:**

$\Omega$ observed entries

$(i_0, j_0)$ new entry

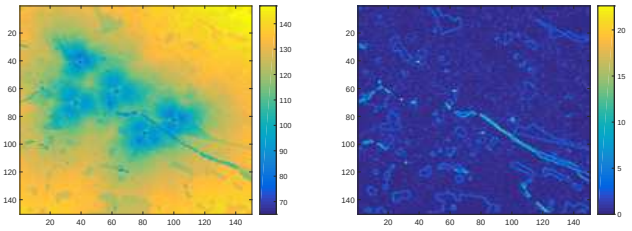
$\mathbf{X}_\Omega, \mathrm{X}_{i_0,j_0}$ values of observed entries + new entry

$\mathbf{U}, \mathbf{V}$ decomposition obtained at previous step

**Do:**

$\Omega \leftarrow \Omega \cup (i_0, j_0)$

update $\mathbf{v}_{j_0}$ according to (4) and similarly for $\mathbf{u}_{i_0}$.



(a) Original Coverage Map. (b) Filtered by $f$.

**Fig. 1**. Berlin Coverage Map, from the Momentum Project.

## 3. ADAPTIVE SAMPLING STRATEGY

In the sequel we assume that the matrix we want to complete is smooth. Namely, if two entries $(i_1, j_1)$, $(i_2, j_2)$ are such that $|i_1 - i_2| \leq 1$ and $|j_1 - j_2| \leq 1$, then $|\mathrm{X}_{i_1 j_1} - \mathrm{X}_{i_2 j_2}| \leq \Delta$ for some small positive number $\Delta$. From an image processing perspective, it actually constrains the variation of intensity from a pixel to any of its neighbors. It could also be seen as a condition on the Lipschitz constant of the gradient. In that sense, it is rather restricted to matrices that represent images. In Figure 1(a) we give a concrete example to illustrate our point. It consists of a coverage map i.e. a set of radio measurements over a discretized geographical area. The smoothness assumption is especially relevant in this context, as the intensity of radio signals slowly decreases with the distance to the base stations.

Now consider the 2-dimensional filtering $\tilde{\mathbf{X}}$ of the original image $\mathbf{X}$ by the following kernel:

$$f = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

In image processing, this filtering is known as an edge detection operator. The data smoothness implies that the coefficients of $\tilde{\mathbf{X}}$ should be relatively small, more precisely, for each set of indices $(i, j)$ : $|\tilde{\mathrm{X}}_{ij}| \leq \frac{8}{9}\Delta$.

Figure 1(b) shows the image corresponding to $|\tilde{\mathbf{X}}|$ for the coverage map. We see that apart from very few points of interest in the relief, the values of the pixels are close to zero. In this example, less than $0.5\%$ of the entries have absolute values greater than $10$, so the smoothness assumption approximately hold for $\Delta = 10$.

**Table 3**. Adaptive sampling under smoothness assumption

**Initialize:**

$\mathbf{Y}$ Current Solution of the MC,

$\Omega$ observed entries,

$N$ size of the mini-batch

**1:** Compute $\tilde{\mathbf{Y}} = \text{2d-filter}(\mathbf{Y}, f)$

**2:** Sort the entries of $\tilde{\mathbf{Y}}$ in descending order and return the indices of the $N$ largest entries (in absolute value) that are not in $\Omega$.

Let us define as $\mathbf{Y}$ our current estimate of the incompletely observed matrix $\mathbf{X}$. Our claim is that, if we want to improve the reconstruction with new samples, we have to pay more attention to the entries $(i, j)$ for which $|\tilde{\mathrm{Y}}_{ij}|$ is large. To understand why, recall that the filtering presented here is a bounded linear operator: there exists a constant $M$ such that $\|\tilde{\mathbf{Z}}\| \leq M\|\mathbf{Z}\|$ for any matrix $\mathbf{Z} \in \mathcal{M}_{m,n}(\mathbb{R})$. Hence, if the reconstruction is close enough to the original matrix, we should have : $\|\tilde{\mathbf{Y}} - \tilde{\mathbf{X}}\| \leq M\|\mathbf{Y} - \mathbf{X}\| \leq \epsilon$ for some small $\epsilon$. We are in finite dimension so we can choose $\|\cdot\|$ to be the infinity-norm. Thus, by triangular inequality, we have the following constraint on every coefficient $\tilde{\mathrm{Y}}_{ij}$ :

$$|\tilde{\mathrm{Y}}_{ij}| \leq |\tilde{\mathrm{Y}}_{ij} - \tilde{\mathrm{X}}_{ij}| + |\tilde{\mathrm{X}}_{ij}| \leq \epsilon + \frac{8}{9}\Delta.$$

As announced, the coefficients of $\mathbf{Y}$ should be small. If, on the contrary, $|\tilde{\mathrm{Y}}_{ij}|$ is large (typically larger than $\Delta$), it indicates that $|\tilde{\mathrm{Y}}_{ij} - \tilde{\mathrm{X}}_{ij}|$ is also large i.e. our MC algorithm fails at predicting the correct values in the vicinity of point $(i, j)$. So it is relevant to ask for further information at this location. Following this principle, we propose the rule in Table 3 to select a mini-batch of $N$ unknown entries that should enhance the prediction.

## 4. SIMULATIONS

In this section we perform experiments to show the benefit of our AdSSA strategy compared to URS and QbC.

The global idea of QbC is to quantify the uncertainty of prediction using several different models for solving the same task [15]. For each coefficient of the matrix, we define a global level of disagreement between the models. Then, the entries with the highest disagreement are recommended. We refer the reader to [14] for more details on this rationale in the case of the MC task. In this paper, we have used our algorithm and two more schemes:

- **SVT**: the well-known Singular Value Thresholding algorithm [5]

- **KNN Regression**: for each missing entry $(i, j)$, the predicted value is a weighted average of the $i^{\text{th}}$ coefficients of the $K$ columns that are the closest to the $j^{\text{th}}$ column. We took $K = 5$ in the sequel.

In all experiments we consider the pathloss map of Berlin, originating from the data of the Momentum project (Figure 1(a)): it is a $150 \times 150$ gray-scale image, with values ranging from 70 to 150. To assess the quality of reconstruction, we use the Normalized Mean Squared Error (NMSE), defined as follows :

$$\text{NMSE} = \frac{\|\mathbf{X} - \mathbf{Y}\|_F^2}{\|\mathbf{X}\|_F^2} \tag{5}$$

where $\mathbf{X}$ is the ground truth matrix with all coefficients and $\mathbf{Y}$ the prediction we want to test.

In the first experiment, 2000 entries (around $8\%$ of the data) are measured uniformly at random and a first estimate is computed with our MC algorithm. Then we let the size of the mask grow progressively: at each time step, we select $N = 20$ entries that will be incorporated to the model, according to the different strategies. AdSSA is confronted with URS and QbC. We found parameters $k = 6$ and $\gamma = 2$ to yield good results for the current dataset. Figure 2 shows the evolution of the NMSE with those values. It is clear that AdSSA permits a much faster reduction of the error than URS. For instance 100 well chosen entries give the same accuracy as 1000 randomly chosen entries. It also performs slightly better than QbC. The areas recommended by AdSSA allows for improved accuracy of the reconstruction.

Two important remarks about the online and computational aspects of this simulation: first, since we recommend entries at each time step, this is likely to be much more efficient than using a single (large) batch of equivalent size. As a matter of comparison, with 3000 entries, the NMSE reaches $1.18 \times 10^{-4}$ at the end, whereas a single batch gives $1.86 \times 10^{-4}$. Secondly, this method requires constant updates of the MC solution. These are accomplished efficiently thanks to our online algorithm. At the same time, the cost of the edge detection filtering is about $O(nm)$ operations and thus reasonable (same order of magnitude as the MC update). However, because kNN and SVT are offline methods that must be retrained at each iteration, with a very high computational cost, it is impractical to use QbC in this context. For the purpose of comparison we have "emulated" an online process but in fact QbC had to be recomputed from scratch at each time step (hence the dotted line on Figure 2). Even on this small dataset, this "online QbC" took one day of computation on a standard desktop computer with CPU 2.60 GHz and 8.00 GB RAM (less than a minute for AdSSA with the same configuration). This highlights another advantage of AdSSA over QbC : scalability.

To compare AdSSA and QbC on an equal footing, we consider the possibility of recommending one single batch of data. As previously we start with 2000 observed entries

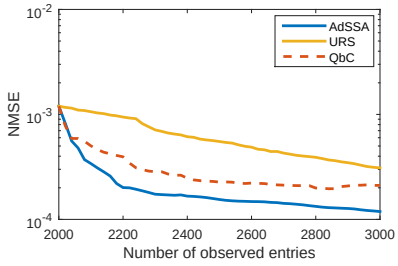



**Fig. 2**. Comparison of AdSSA, URS and QbC in an online manner. The evolution of the NMSE with respect to the number of observed entries.

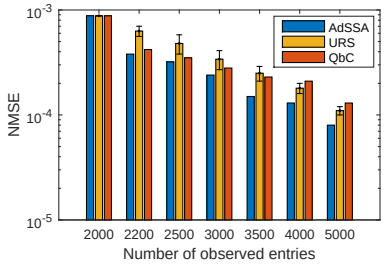



**Fig. 3**. Comparison of AdSSA, URS and QbC in an offline manner. The NMSE for different sizes of the batch of recommended entries.

and recompute the MC solution with a certain amount of recommended entries. Figure 3 presents the NMSE for different sizes of the batch. Even in this case, AdSSA does significantly better than QbC, especially when we recommend many entries : when the batch size is larger than 2000, QbC is beaten by URS whereas AdSSA still outperforms URS.

To sum up, AdSSA yields very good recommendations to help improve accuracy of the MC task. It outperforms URS and QbC in terms of accuracy and, unlike QbC, it is an online method, which gives all the more freedom and flexibility in the sampling. In particular we can easily recommend small batches one at a time.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have presented an efficient online MC algorithm, together with a new adaptive sampling strategy to improve accuracy of the reconstruction. It has been successfully applied to the coverage map completion problem, outperforming both naive URS and QbC in terms of accuracy. To the best of our knowledge it is also the first attempt to provide online recommendations with practical run time. The only assumption necessary for AdSSA is the smoothness of the data matrix. Future directions of work involve generalization of this result to other types of matrices (with adapted kernels).

## 6. REFERENCES


[1] Emmanuel J Candès and Benjamin Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[2] Raghunandan H Keshavan, Sewoong Oh, and Andrea Montanari, "Matrix completion from a few entries," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE, 2009, pp. 324–328.

[3] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 9, pp. 2117–2130, 2013.

[4] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, , no. 8, pp. 30–37, 2009.

[5] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[6] Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst, "Matrix completion on graphs," *arXiv preprint arXiv:1408.1717*, 2014.

[7] Jakub Marecek, Peter Richtárik, and Martin Takác, "Matrix completion under interval uncertainty," *arXiv preprint arXiv:1408.2467*, 2014.

[8] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[9] Jiashi Feng, Huan Xu, and Shuicheng Yan, "Online robust pca via stochastic optimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 404–412.

[10] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1087–1099, 2012.

[11] Symeon Chouvardas, Yannis Kopsinis, and Sergios Theodoridis, "Robust subspace tracking with missing entries: the set-theoretic approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5060–5070, 2015.

[12] Steffen Rendle and Lars Schmidt-Thieme, "Online-updating regularized kernel matrix factorization models for large-scale recommender systems," in *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 2008, pp. 251–258.

[13] Akshay Krishnamurthy and Aarti Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Advances in Neural Information Processing Systems*, 2013, pp. 836–844.

[14] Symeon Chouvardas, Stefan Valentin, Moez Draief, and Mathieu Leconte, "A method to reconstruct coverage loss maps based on matrix completion and adaptive sampling," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6390–6394.

[15] Shiladri Chakraborty, Jiayu Zhou, Vineeth Balasubramanian, Sethuraman Panchanathan, Ian Davidson, and Jieping Ye, "Active matrix completion," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013, pp. 81–90.

[16] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 635–644.

[17] Simon Funk, "Netflix update: Try it at home," 2006.

[18] Marcus Hardt, "Understanding alternating minimization for matrix completion," in *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE, 2014, pp. 651–660.