# Algebraic-Combinatorial Methods for Low-Rank Matrix Completion with Application to Athletic Performance Prediction

Duncan A.J. Blythe*, Louis Theran, Franz Kiraly

June 12, 2014

**Abstract**

This paper presents novel algorithms which exploit the intrinsic algebraic and combinatorial structure of the matrix completion task for estimating missing entries in the general low rank setting. For positive data, we achieve results outperforming the state of the art nuclear norm, both in accuracy and computational efficiency, in simulations and in the task of predicting athletic performance from partially observed data.

## 1 Introduction

Matrix completion (MC) is the task of filling in the missing entries of a matrix. The most popular statistical model to this end is the low-rank matrix model. Recently, an algebraic-combinatorial approach to the low rank-matrix completion problem was developed, where the authors derived the first:

1. algorithms which determine which entries of a matrix may be estimated exactly in the limit of low-noise, in arbitrary rank and with *any* method whatsoever [7].
2. algorithms for the rank 1 case which utilize *all* information available with regard to missing entries and are thus optimal for this case [6], outperforming, for example, the nuclear norm approach [1]. These algorithms work *locally* on each missing entry, leading to a fraction of the computational cost of completing all missing entries.
3. error guarantees and computable error bounds for the rank 1 case on the single entries [6].

In this paper, we extend these benefits to the general low-rank setting, when the true matrix takes positive values, as in, e.g. the *Netflix* challenge. We propose algebraic-combinatorial algorithms which are

*D.A.J.B and F.K. contributed equally to this work

1. more accurate in simulations in certain regimes (high noise and low-noise, low-medium observation probability) and on real-world data.

2. considerably faster than the nuclear norm approach in completing the entire matrix.

3. *even* faster when only certain entries of a matrix are required to be completed, since the algebraic-combinatorial algorithms operate locally on the observed matrix.

### Related work

Low-rank matrix completion has received a great deal of attention from the machine learning community. Three main strands of research have developed: (1) convex relaxations of the rank constraints (e.g., [1, 10, 12, 4, 13]) (2) spectral methods (e.g., [5, 9, 2]), and (3) the novel algebraic approach discussed above [7, 6].

The approaches (1) and (2) focus on (i) estimating every missing entry; (ii) denoising every observed entry; and (iii) minimizing the MSE over the whole matrix. The algebraic approach (3) allows for the construction of single-entry estimators which minimize the error of the entry under consideration.

## 2 Theory

### 2.1 Positive Low Rank Model

We assume that we observe some incomplete set of entries $E \subseteq [m] \times [n]$ of an unknown matrix $A \in \mathbb{R}^{m \times n}$ with all-positive entries. We also assume that $A$ is a low-rank perturbation (with potentially multiplicative or additive noise) of a low rank "true"matrix.

Positive data of this form are common in applications, e.g. the *NetFlix* challenge dataset (and in general in machine learning e.g. motivating non-negative matrix factorization [8, 3]). This paper presents methods which outperform the state of the art algorithms, nuclear norm [1] and OptSpace [5], for positive data.

### 2.2 Circuits

In this section we describe how one obtains polynomials from the observed matrix, which include variables corresponding to missing entries and which must vanish, in limit of low noise, in order that the matrix has rank $r$. This will then yield in the following section a strategy for completing the missing entries in $A$.

The starting point here is the following classical theorem.

**Theorem 2.1.** *Let $A \in \mathbb{C}^{m \times n}$. Then $A$ is of rank $r$ or less, if and only if all determinants of $(r+1) \times (r+1)$-submatrices of $A$ vanish.*

Thus, given any submatrix of size $(r+1) \times (r+1)$, the polynomial given by its determinant with variables at its missing positions, must vanish, otherwise $A$ cannot have rank $\leq r$.

If, this polynomial contains only one variable, $X_s$, then the polynomial may be solved uniquely, since linear in $X_s$ (the monomials in the determinant never repeat entries). This solution may be then taken, in the noise free case as the completed entry (we approach the noisy case below).

However, in some cases, the subdeterminant may contain multiple variables. Suppose we are interested in completing just one entry $s$, i.e. solving for $X_s$. In order to make this feasible, we need to find additional polynomials containing the remaining variables. These additional polynomials, however, may in turn again contain additional variables. A solution for $X_s$ is thus only possible if a joint solution set may be obtained by this process which has zero degrees of freedom. If the number of variables only proliferates with the number of polynomials added, no solution for $X_s$ will be possible.

This observation leads us to define circuits of a given rank, which formalizes the notion of the locations on the matrix covered by this process and in particular, circuits which allow us to complete entries in principle. The term circuit is suggestive of the fact that there is a graph theoretic interpretation to this process, which is discussed in [6]; in this context, a purely algebraic formulation is possible.

First we consider collections of potential entries in the matrix which are plausible given the rank $r$ assumption

**Definition 2.2.** Let $S$ be a collection of indices, let $B_s \in \mathbb{C}, s \in S$ be an indexed collection of numbers. Then we say that the $(B_s)_{s \in C}$ is *compatible with rank* $r$, if there exists matrix $A \in \mathbb{C}^{m \times n}$, of rank $r$ or less, with $A_s = B_s$ for all $s \in C$.

**Definition 2.3.** Let $C \subseteq [m] \times [n]$ be a subset of indices. Then $C$ is called a *circuit* of rank $r$ if:

**(i)** For every proper subset $S \subsetneq C$, any collection of numbers $B_s \in \mathbb{C}, s \in S$ is compatible with rank $r$.

**(ii)** For any $e \in C$, and almost all collections of numbers $B_s \in \mathbb{C}, s \in C \setminus \{e\}$, there are at most finitely many $B_e$ yielding a collection $B_s, s \in C$ that is compatible with rank $r$.

This definition formalizes the notion, outlined above, that the polynomials we obtain from the matrix should at some point yield finitely many solutions for the missing entries. The simplest circuit of rank $r$ is the support of an $(r+1) \times (r+1)$-submatrix. (i) and (ii) hold since each sub-minor with one missing entry admits exactly one compatible completion, except in a zero set of pathological cases where there are more, e.g. when all observed entries vanish.

For an $(r+1) \times (r+1)$ sub matrix, one naturally obtains a polynomial, viz. the determinant including the missing entries as variables, which vanishes whenever the submatrix is compatible with rank $r$. It is possible to generalize this polynomial to arbitrary circuits of rank $r$. Thus, with each circuit, one may associate a unique *circuit polynomial*, which coincides with the determinant polynomial for subminors [7].

**Proposition 2.4.** *Let $C \subseteq [m] \times [n]$ be a circuit of rank $r$. Then, there is a(n) (up to multiplicative constant) unique irreducible polynomial $\theta_C$ in variables $X_s, s \in [m] \times [n]$ such that: $B_s \in \mathbb{C}, s \in C$ is compatible with rank $r$ if and only if $\theta_C(B_s, s \in C) = 0$.*

One can prove the following theorem [7] which generalizes Theorem 2.1:

**Theorem 2.5.** *Let $A \in \mathbb{C}^{m \times n}$, let $E \subseteq [m] \times [n]$ be a set of observed entries. Then, the collection $A_e, e \in E$ is compatible with rank $r$ if and only if for all circuits $C \subseteq E$, the circuit polynomial evaluations $\theta_C(A_e, e \in E)$ vanish.*

# 3 Reconstruction by variance minimization

## 3.1 Reconstruction by circuits

Theorem 2.5 implies that the circuit polynomial of any circuit running through the observed entries and an unobserved entry should vanish in the low-noise limit – thus yielding a solving strategy for that single entry.

**Definition 3.1.** Let $E \subseteq [m] \times [n]$ be a set of observed entries, let $e \in [m] \times [n]$ (observed or unobserved). A circuit $C \subseteq E \cup \{e\}$ of rank-$r$ with $e \in C$ is called *solving circuit* for $e$ (w.r.t. $E$). If $\theta_C$ has degree 1 in $X_e$, we call $C$ a *unique solving circuit*.

Every unique solving circuit gives rise to a rational *solving equation* of the form [7]:

$$A_e = \frac{f_C(A_s, s \in S)}{g_C(A_s, s \in S)}, \quad \text{where } S = C \setminus \{e\}.$$

In the case of a $(r+1) \times (r+1)$-submatrix, the solving equation takes the form of monomials of the determinant not containing $X_e$ on the numerator of the right hand side, and on the denominator, monomials containing $A_e$, but with $A_e$ factored out.

In the absence of noise, it would suffice to find exactly one such equation and substitute the observed $A_s$. If noise is present, we will follow a linear variance-minimization strategy as in the pseudocode detailed as Algorithm 1, which fulfills the desideratum that an estimate should be subject to the minimum variance when a class of estimators is under consideration.

Crucial to note at this point that a variance estimate is simultaneously an *error* estimate on the prediction of *individual matrix entries* we obtain.

**Algorithm 1** variance-minimizing local completion

1: Find solving circuits $C_1, \ldots, C_m$ for $A_e$
2: Compute candidate estimates $a_1, \ldots, a_m$ via the $C_i$
3: Compute (co-)variance estimates $\sigma_1, \ldots, \sigma_m$, from $s_e$
4: return a linear combination $\widehat{A_e} = \alpha_1 a_1 + \cdots + \alpha_m a_m$ with minimal variance

We will present two prototypical algorithms employing this strategy: one for rank 1 and one for general local rank $r$ matrix completion.

### 3.2 Variance minimizing reconstruction: rank 1

For rank 1, we use a variant of the algebraic algorithm (which we refer to as *Algebraic Combinatorial Completion in Rank One* - ACCRO) presented in [6]. In that algorithm, computations are performed for logarithmic entries $a_e = \log A_s$, for which circuits become linear equations. In order to speed up computations, we will only consider circuits of length 4 or less (=determinants and the entry itself, if observed) for reconstruction, also we assume that the multiplicative noise is equal on all log-entries. An additional speed-up is possible if one neglects the correlations between circuits. We will refer to this variant as fACCRO - *fast ACCRO*. An informal description can be found in Algorithm 2. The algorithm follows a variance minimizing strategy as outlined in Algorithm 1; the weighting used is the logarithmic weighting of Equation (4). This may in principle be used to estimate the error of the estimate. The reason that fACCRO is fast is that, if multiple entries are required, Step 5. may be performed *en masse*.

**Algorithm 2** fACCRO; *input* incomplete matrix $A$, missing index $(i, j)$; *output* completed entry $\widehat{A}_{i,j}$

1: Find all $l$ where $A(i, l)$ is observed.
2: Find all pairs $A(k, l), A(k, j)$ where both are observed.
3: Compute $w_k = |A(k, l)|$.
4: Normalize the $w_k$ so that $\sum_k w_k$ =1
5: Store $b_l = \exp(\sum_k w_k(\log(A_{k,j}) - \log(A_{k,l})))$
6: Estimate $\widehat{A}^k_{i,j} = A_{i,l}b_l$
7: Compute weights $w'_l = |A(i, l)|$ and normalize $\sum_k w'_l$ =1
8: Estimate $\widehat{A}_{i,j} = \exp(\sum w'_l \log A(i, l))$

### 3.3 Variance minimizing reconstruction: rank $r$

For general low rank, it is not the case that each circuit of rank $r$ determines exactly one solution when the circuit polynomial's variables are substituted by generic entries in all but one variable $X_s$.

However, when the circuit is a unique solving circuit, and there is noise on the matrix, then $\theta_C(A_s) \approx 0$, where $A$ is the true matrix.

In this higher rank case, the variance of the estimate given by a circuit $C$ must be approximated. To do this we perform a Taylor expansion of $\theta_C(A_s)$ around the solution for the exact underlying matrix in Section A of the Appendix.

The considerations of this section and the previous section lead us to define the following algorithm: For any missing entry at position $s$, require that any determinant of a $r + 1 \times r + 1$ minor through that entry is 0. For each such minor $k$, estimate the variance of the estimate by Equation (4), $w_s$ and the solutions given by the solving the minor determinant equations by $\hat{A}^k_s$. Then average these estimates by Algorithm 1, to yield an estimate of minimum variance. For large matrices, a set number of sub minors should be chosen at random for computational gains. If the observation probability is

low, then one decreases $r$ until sufficient minors are present for a stable solution. See Algorithm 3.

**Algorithm 3** vm-Closure; *input* A, missing position (i,j) in almost complete $r+1 \times r+1$ subminors;*output* estimate $\widehat{A}_{i,j}$ of $A_{i,j}$

1: Find minors of $A$ including $A_{i,j}$ with all entries but one missing: $B^k$ for $k = 1 \dots$ `iterations` of size $\widehat{r}+1 \times \widehat{r}+1$ s.t. $B^k_{r+1,r+1} = A_{i,j}$
2: Set $B^k_0$ to be $B^k$ with a zero in the bottom corner, and $B^k_1$ a 1.
3: Set $a_1 = \det(B^k_1)$ and $a_0 = \det(B^k_0)$
4: Set $\delta B_k = \frac{1}{|a_1 - a_0|} + \frac{|a_0|}{(a_1-a_0)^2}$
5: Define a probability measure $q(k)$ over $k$ by normalizing $\sum_{k'} \frac{1}{(\delta B_{k'})^2} \delta(k-k')$
6: Complete $\widehat{A}_{i,j} = \frac{1}{\texttt{iterations}} \sum_k q(k)(1 - \frac{a_0}{a_1 - a_0})$

## 3.4 Spectral meta-Algorithms

In the example depicted in Figure 1, we generate a sample from model of Equation 1, in the noise free case, where the true matrix is of rank 2. We then apply the ACCRO algorithm to this matrix. Although this algorithm is motivated by the assumption of a rank 1 truth in its solution strategies, we observe that the singular values (left) of the estimated matrices nevertheless reveal the rank 2 structure of the true matrix. In the right hand panel, we see, moreover, that the second singular vector well approximates the singular true singular vector. We compare the same graphics (red), but for completion with the mean as per [2] ( [2] proposes a simple algorithm for matrix completion: truncation of the SVD with 0s for missing entries). We see that the singular values in this case do not reveal a clear rank 2 structure, and the 2nd singular vector provides a poorer approximation to the true 2nd singular vector. Our meta-Algorithms 4 and 5 use the fact that we may obtain the rank 2 upwards singular vectors via the algebraic algorithms in rank 1. Both obtain the singular vectors using the output of one of the previous algorithms. The *Spectral matrix completion bootstrap* (SMCB), Algorithm 4 completes the full matrix by solving linearly for each row of $A$. Optspace [5] is an approach which first fills 0s into the missing entries, performs certain trimming operations, and then truncates the SVD of this coarse completion; this truncation is then fed into an optimization routine which further refines the estimate. Our *meta-OptSpace* (mOS), Algorithm 5, initializes the OptSpace optimization, instead, with the output of any of the previous algorithms. Thus we obtain algorithm instances such as mOS(SMCB(ACCRO)).

These algorithms are justified by the considerable gains in accuracy over competing spectral methods such as OptSpace observed in our simulations (Sections 4.2) and application (Section 5). Further work will aim at understanding their asymptotics.

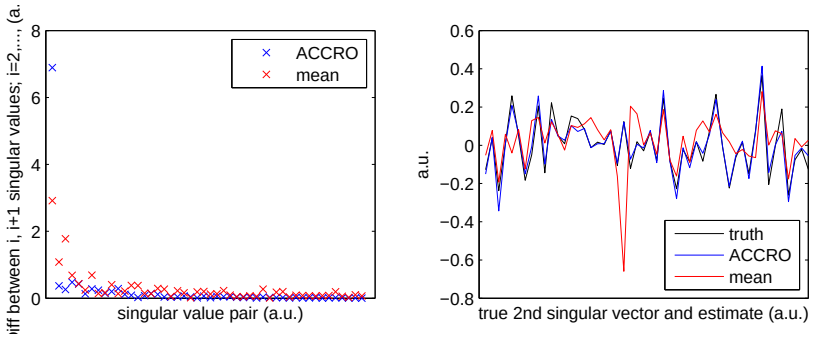


Figure 1: The figure displays in the left hand panel the difference between the 2nd and 3rd, 3rd and 4th, ... singular values on the matrix completed, by the mean (red) and by the ACCRO algorithm (blue), from left to right, when the true matrix has rank $r = 2$ and the noise level, $\epsilon = 0$, $p = 0.5$, $N = M = 50$. The ACCRO exhibits a prominent separation between the second and remaining eigenvalues, whereas, the completion by the mean, poor separation. This is again reflected in the singular vectors (right hand panel), where the ACCRO's output (blue) well approximates the true 2nd singular vector(black) and whereas the mean completion approach approximates only poorly (red).

**Algorithm 4** SMCB; *input* incomplete matrix $A$, initial estimate $A_{init}$; *output* completed estimated $\widehat{A}$

1: Let $USV^\top$ be the SVD of $A_{init}$
2: **for** i = rows of $A$ **do**
3: Let $Y = [V_1, \ldots, V_r, A_i^\top]^\top$
4: Rearrange the rows of $Y$ to give $Y = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ where $\begin{bmatrix} A_{11} & A_{12} \end{bmatrix}$ contain the singular vectors, $A_{22}$ corresponds to the unobserved entries and $A_{21}$ the observed entries of $A_i$ ($i^{\text{th}}$ row of $A$)
5: Set $A_{22} = A_{21}(A_{11})^+ A_{12}$ where $Z^+$ is the Moore-Penrose pseudo inverse of a matrix $Z$.
6: Complete the corresponding entries of $A_i$ using $A_{22}$
7: **end for**

**Algorithm 5** meta-OptSpace; *input* incomplete matrix $A$, initial estimate $A_{init}$; *output* completed estimated $\widehat{A}$

1: Initialize $\widehat{A}_1$ as $A_{init}$, using any of the algebraic algorithms.
2: Perform an SVD truncation of $\widehat{A}_1$ to rank $r$.
3: Perform the optimization loop of OptSpace starting at $\widehat{A}_1$, to output $\widehat{A}$

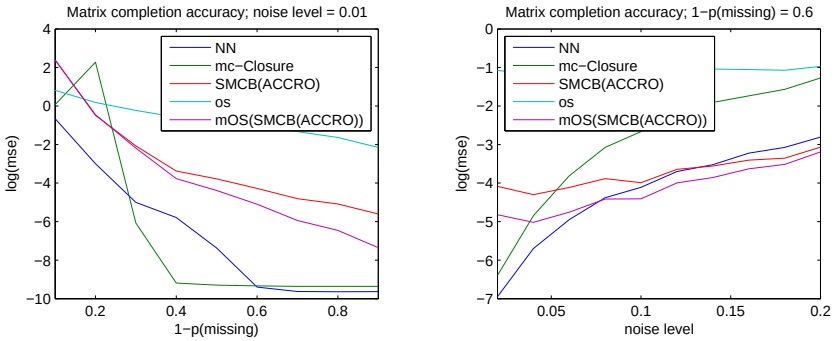


Figure 2: The figure displays the results of the simulation described in Section 4.2. The left hand panel displays the results for additive noise.

# 4 Experiments on Simulated Data

## 4.1 Simulated Data

For simulated data subject to multiplicative noise, we sample:

$$A = (UV^\top) \circ \mathcal{E} \tag{1}$$

For simulated data subject to additive noise, we sample

$$A = UV^\top + \mathcal{E} \tag{2}$$

Each entry of $U, V$ is sampled independently from $|z|$, where $z$ is a standard Gaussian. For multiplicative noise we consider each entry of $\mathcal{E}$ as sampled independently from a log-normal centered around 1 ($\exp(\epsilon z)$, where $\epsilon$ is the noise level). For additive noise we consider each entry to sampled from $\epsilon|z|$. Each entry of the mask $M$ is sampled independently from $\{0, 1\}$ from a Bernouilli distribution with parameter $p$.

## 4.2 Accuracy in the Matrix Completion Task

The aim of this simulation is to assess the accuracy of the algebraic methods, baselining against the Nuclear Norm algorithm and OptSpace.

In the first simulation, 100 matrices with additive noise and masks are realized for $\epsilon = 0.01$ and for each probability that an entry is missing, $p = 0.1, \ldots, 0.9$.

In the second simulation, 100 matrices with multiplicative noise and masks are realized for $\epsilon = 0.02, 0.04, \ldots, 0.2$ and with the probability that an entry is missing, $p = 0.6$. In both cases compare Nuclear Norm, OptSpace, SMCB(fACCRO), vm-Closure and mOS(SMCB(fACCRO).

The results are displayed in Figure 2. The left hand panel displays the results of the first simulation: the vm-Closure algorithm outperforms Nuclear at medium observation probability. Moreover, the efficient algorithms SMCB(fACCRO) and mOS(SMCB(fACCRO)

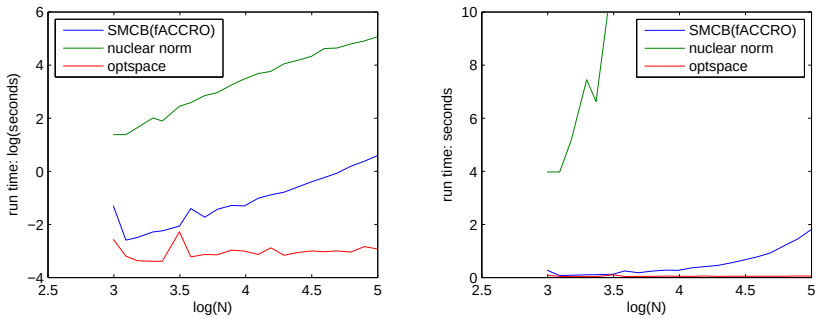


Figure 3: The figure displays the results of the simulation described in Section 4.3. The left hand panel displays computation time in log coordinates of the tested methods (legend), Nuclear Norm, OptSpace and SMCB(fACCRO). The right hand panel displays the raw computation time. The results show that SMCB(fACCRO) outperforms Nuclear Norm (while performing on a similar level in accuracy; see Section 4.2) whereas although OptSpace is the fastest, it yields the poorest matrix completion performance.

far outperform OptSpace. The right hand panel displays the results of the second simulation. Here, the meta-algorithms, initialized by algebraic-combinatorial solutions, SMCB(fACCRO) and mOS(SMCB(fACCRO), outperform Nuclear Norm for higher noise levels; all algebraic-combinatorial algorithms far outperform OptSpace.

### 4.3 Computational Efficiency

The aim of this simulation is to compare the computational efficiency of our fastest rank $r$ algorithm, SMBC(fACCRO), with the baselines nuclear norm (with cross validation) and OptSpace. Optspace, and our methods do not require cross validation since an estimate of the rank may be computed from the singular value spectrum. We generate samples from the model for $p = 0.5$, $\epsilon = 0$ and $N = M = 20, \ldots 150$ and record the computation times. The results show that SMCB(fACCRO) is considerably more efficient than Nuclear Norm, and yields competitive accuracy; SMCB(fACCRO) outperformed in efficiency by OptSpace but provides considerably greater accuracy, as seen in the previous simulation.

## 5 Application of Methods to Prediction of Athletic Performance

The publicly available data of a subset of runners was obtained from `http://www.thepowerof10.info/`, which is a database cataloguing the performances of Great British runners, both professional and amateur. Each athlete in the database is tagged with information on the date, location, distance as well as the performance (in hours, minutes and seconds) over each distance.

Existing methods for the prediction of running performance have used only simple parametric models which implicitly assume that the true model has rank 1. The best known prediction (Riegel formula) predicts linearly in log space [11]:

$$T_2 = T_1 \times (D_2/D_1)^{1.06} \tag{3}$$

Here, $T_i$ refers to the times and $D_i$ to the corresponding distances.

We show that we can learn a higher rank model from the data which outperforms this rank 1 method. The existence of such a solution is tantamount to the possibility of building estimated athlete-specific information into the predictor.

The commonly attempted distances are 100m, 200m, 400m, 800m, 1500m, 1Mile, 5000m, 10000m, Half Marathon (21.1km), Marathon (42.2km), thus we obtain a matrix of size no.athletes $\times 10$. For each athlete $A$, we choose his/her best event relative to the population (the event for which the likelihood that another athlete $B$ is superior than $A$'s best performance event is lowest). $A$'s best performance is entered in seconds into the corresponding column of the matrix. In addition, we fill in the remaining columns of the matrix with times achieved over the remaining distances which occurred within 1 year of the best performance in $A$'s best event. The remaining entries are recorded as unobserved. In this paper we consider those athletes having attempted at least 7 events (no.athletes $\approx 400$).

Clearly the matrix takes positive values. Also important to note here is that *the noise is multiplicative*. This is since, a) e.g. the expected deviation in a Marathon is on the order of minutes, whereas over 100m on the order of tenths of seconds; b) slower runners are also more inconsistent.

### 5.1 Matrix Completion Performance

We test the MC performance, for 100 randomly deleted entries, of Nuclear Norm, OptSpace, mOS(SMCB(ACCRO)) and mOS(SMCB(vm-Closure rank 2)); the Riegel formula (given by Equation (3)), serves as a baseline. After deletion, we divide each column by the mean of that column, so as to bring the columns on to the same scale; otherwise the mean squared error is dominated by performances over longer distances.

The results show that a higher rank model yields a better predictor than the Riegel rank 1 predictor and that the algebraic methods significantly outperform all baselines, including nuclear norm. Intriguing is the fact that OptSpace, when improperly initialized yields the highest MSE but when properly initialized, the lowest. This is reminiscent of recent insights in deep learning, that an intelligent initialization of a deep network yields considerable performance gains.

## 6 Discussion and Conclusion

In this paper we presented algebraic combinatorial algorithms which outperform the state of the art on positive low-rank matrices in terms of accuracy and computational cost. Furthermore, the algebraic method is the only existing method which allows for

| method | mOS(SMCB(ACCRO)) | nn | mOS(SMCB(vm-Cl r2)) | Riegel | OS |
|---|---|---|---|---|---|
| MSE $\times 10^2$ | **0.35** | 0.42 | **0.35** | 2.31 | 10.3 |
| $\pm 2\sigma$ | **0.019** | 0.026 | **0.019** | 0.086 | 0.19 |

Table 1: Error given as MSE in dimensionless units (squared percentage of the mean time for any given distance). The percentiles are $\pm 2$ standard deviations estimated via a bootstrap with 1000 iterations.

the reconstruction of single entries and makes possible error estimates for them. We conjecture that the algorithms may be generalized to non-positive or complex matrices or other incomplete data imputation tasks following a different model.

# Acknowledgments

LT is supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 247029-SDModels. This research was conducted while DAJB was a guest of FK at Mathematisches Forschungsinstitut Oberwolfach, supported by FK's Oberwolfach Leibniz Fellowship. DAJB is supported by a grant from the German Research Foundation (DFG), research training group GRK 1589/1 "Sensory Computation in Neural Systems".

# References


[1] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[2] Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *arXiv preprint arXiv:1212.1247*, 2012.

[3] Shaun M Fallat, Charles R Johnson, and Ronald L Smith. The general totally positive matrix completion problem with few unspecified entries. *Electron. J. Linear Algebra*, 7:1–20, 2000.

[4] Rina Foygel and Nathan Srebro. Concentration-based guarantees for low-rank matrix reconstruction. *Arxiv preprint arXiv:1102.3923*, 2011.

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

[6] Franz J. Király and Louis Theran. Obtaining error-minimizing estimates and universal entry-wise error bounds for low-rank matrix completion. *NIPS 2013*, 2013. arXiv 1302.5337.

[7] Franz J. Király, Louis Theran, Ryota Tomioka, and Takeaki Uno. The algebraic combinatorial approach for low-rank matrix completion. *arXiv Preprint*, 2012. arXiv 1211.4116.

[8] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[9] Raghu Meka, Prateek Jain, and Inderjit S. Dhillon. Guaranteed rank minimization via singular value projection. *NIPS 2010*, 2010. Eprint arXiv:0909.5457.

[10] Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Ann. Statist.*, 39(2), 2011.

[11] Peter S Riegel. Athletic records and human endurance. *American Scientist*, 69(3):285–290, 1980.

[12] Ruslan Salakhutdinov and Nathan Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2056–2064. 2010.

[13] Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Proc. of the 18th Annual Conference on Learning Theory (COLT)*, pages 545–560. Springer, 2005.

# A Derivation of the weighting

Using the notation of Section 3.1 one can consider the first order approximation to the estimated standard deviation of an approximate solution, as follows:

$$\delta A_e = g_C^{-1} \sum_{s \in S} f_{C,s} \cdot \delta A_s - \frac{f_C}{g_C^2} \sum_{s \in S} g_{C,s} \cdot \delta A_s,$$

where we denote $f_{C,s} = \frac{\partial f_C}{\partial X_s}, g_{C,s} = \frac{\partial g_C}{\partial X_s}$ and evaluate at $A_s$. If the noise is closer to multiplicative, considering the logarithms gives a better approximation, yielding

$$\begin{aligned}
\delta \log A_e = \frac{\delta A_e}{A_e} &= f_C^{-1} \sum_{s \in S} f_{C,s} \cdot \delta A_s - g_C^{-1} \sum_{s \in S} g_{C,s} \cdot \delta A_s \\
&= \frac{\delta A_e}{A_e} = f_C^{-1} \sum_{s \in S} f_{C,s} \cdot A_s \cdot \delta \log A_s - g_C^{-1} \sum_{s \in S} g_{C,s} \cdot A_s \cdot \delta \log A_s
\end{aligned}$$

For the determinant, both expressions take a particularly simple form. Consider an $(r+1) \times (r+1)$ matrix $A$, where all entries but the bottom right entry $A_{11}$ are observed. Write $a_k$ for the determinants of $A$ where $A_{11}$ is replaced by $k$. Note that $g_C = a_1 - a_0$ and $f_C = a_0$. This yields

$$\begin{aligned}
\delta A_{11} &= (a_1 - a_0)^{-1} \sum_{s \in S} f_{C,s} \cdot \delta A_s - \frac{a_0}{(a_1 - a_0)^2} \sum_{s \in S} g_{C,s} \cdot \delta A_s, \\
\delta \log A_{11} &= (a_0)^{-1} \sum_{s \in S} f_{C,s} \cdot \delta A_s - (a_1 - a_0)^{-1} \sum_{s \in S} g_{C,s} \cdot \delta A_s \\
&= (a_0)^{-1} \sum_{s \in S} f_{C,s} \cdot A_s \cdot \delta \log A_s - (a_1 - a_0)^{-1} \sum_{s \in S} g_{C,s} \cdot A_s \cdot \delta \log A_s.
\end{aligned}$$

The sums are not easy to evaluate, even if all $\delta A_s$ are known or are of similar order of magnitude (the computation of a permanent can be obtained as a special case). However, we expect for randomly sampled data that the components of the sum lie on a single order of magnitude, therefore yielding the two approximations

$$\delta A_{11} \approx \frac{1}{|a_1 - a_0|} + \frac{|a_0|}{(a_1 - a_0)^2} \quad \text{and} \quad \delta \log A_{11} \approx \frac{1}{|a_0|} + \frac{1}{|a_1 - a_0|} \tag{4}$$