Signal Processing 91 (2011) 1334–1338

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast communication

Some empirical advances in matrix completion

A. Majumdar*, R.K. Ward

Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada V6T1Z4

ARTICLE INFO

Article history:
Received 7 July 2010
Accepted 13 December 2010
Available online 21 December 2010

Keywords:
Matrix completion
Non-convex optimization
Blue-noise sampling

ABSTRACT

Solving the matrix completion problem via rank minimization is NP hard. Recent studies have shown that this problem can be addressed as a convex nuclear-norm minimization problem, albeit at an increase in the required number of samples. This paper proposes a non-convex optimization problem (a variant of convex nuclear-norm minimization) for the solution of matrix completion. It also develops a fast numerical algorithm to solve the optimization. This empirical study shows that significant improvement can be achieved by the proposed method compared to the previous ones.

The number of required samples is also dependent on the type of sampling scheme used. This work shows that blue-noise sampling schemes yield more accurate matrix completion results compared to the popular uniform random sampling.

1. Introduction

In the matrix completion problem, only a subset of all the entries in the matrix is known. The problem is to recover the matrix from its partially known samples. This problem arises in a variety of situations like collaborative filtering, system identification [1], direction-of-arrival estimation [2], etc.

A full rank matrix $X_{n\times n}$ (of rank n) has n^2 degrees of freedom. It is thus not possible to estimate all its entries from its partially sampled ones. However, when the matrix has a low rank (r) the degrees of freedom is only $r(2n-r)$, which is much lower than n^2 if r is small. It is possible to recover a low rank matrix from its partial samples via a rank minimization problem of the following form:

$$\min \text{rank}(X) \text{ subject to } Y = M_\Omega(X) \tag{1}$$

where X is the matrix to be recovered, Ω is the set of indices of the sampled entries, M_Ω is a masking operator that selects the entries of X that are within Ω, and Y is the collected sample.

Solving (1) is NP hard and hence is not practical. Recent theoretical studies [3–5] prove that the following convex relaxation (1) can recover the matrix with a high probability:

$$\min ||X||_* \text{ subject to } Y = M_\Omega(X) \tag{2}$$

where $||X||_*$ is called the nuclear norm and is defined as the sum of its singular values.

When the entries are sampled randomly following a uniform distribution, the number of entries that should be known for perfectly recovering the matrix is of the following order:

$$m \geq C\mu^2 rn(\log n)^\alpha \tag{3}$$

where μ is the coherence factor, which depends on the nature of the matrix to be recovered and α is a positive number.

The form (3) is extremely generalized. Different theoretical studies define μ differently. They also prove the results for different values of α.

Generically the rank and the nuclear-norm minimization of a matrix X can be expressed as $||diag(\Sigma)||_p$ ($p=0$ for rank, $p=1$ for nuclear norm) where Σ is the singular value of the matrix X. We see that as the value of p increases from zero (1) to unity (2), the number of samples required to

* Corresponding author.
E-mail addresses: angshulm@ece.ubc.ca (A. Majumdar), rababw@ece.ubc.ca (R.K. Ward).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.12.005

complete the matrix also increases. If we keep the value of p between zero and one, intuitively we feel that the number of samples required to solve the matrix completion problem will also be intermediate between the two extremes. However for fractional values of p, the optimization problem will be non-convex. However this kind of non-convex optimization problems can be solved as efficiently as their convex counterparts.

This work proposes to solve the matrix completion problem via the following non-convex variant of (2)

$$\min ||X||_{p*} \text{ subject to } Y = M_\Omega(X) \tag{4}$$

where $||X||_{p*}$ is the l_p-norm of the singular values of X.

This new and improved approach for solving the matrix completion problem is motivated by recent developments in Compressed Sensing. It is difficult to theoretically prove the improvement brought about by (4) over (3); therefore we validate our claim by thorough experimentation.

This work also shows that instead of using uniform random sampling for selecting the entries of the matrix, blue-noise sampling schemes like Quasi-Random sampling, Quasi-Crystal sampling and Farthest Point sampling yield more accurate recovery results.

In Section 2, we discuss the motivation of this work. Section 3 discusses the proposed non-convex algorithm for solving (4). In Section 4 the different blue-noise sampling strategies are explained. The experimental results are shown in Section 5. Finally in Section 6 the conclusions of this work are discussed.

2. Connection with compressed sensing

The problem of matrix completion is similar to Compressed Sensing (CS). While matrix completion recovers a rank-deficient matrix, i.e. a matrix with only a few non-zero singular values, Compressed Sensing (CS) solves a system of under-determined equations where the solution is sparse, i.e. has only a few non-zeroes. Consider

$$y_{m\times 1} = A_{m\times n} x_{n\times 1}, \quad m < n \tag{5}$$

where x is k-sparse (k non-zeroes only).

Ideally the inverse problem (5) can be solved by an exhaustive search, which can be expressed as

$$\min ||x||_0 \text{ subject to } y = Ax \tag{6}$$

where $||x||_0$ is the number of non-zeroes in x.

This is an NP-hard problem, as it has been proved [6] that the sparse solution can be recovered from a convex relaxation of (6)

$$\min ||x||_1 \text{ subject to } y = Ax \tag{7}$$

where $||x||_1$ is the sum of the absolute values of x.

This is a strong result, which guarantees perfect recovery when the NP-hard optimization problem is replaced by a convex problem that can be solved by linear programming. The price to be paid is an increase in the number of samples/data. When the inverse problem (5) is solved via (6), the number of equations required is only $m \geq 2k$, but when solved by (7), the number of equations is much larger $m \geq Ck\log n$.

For matrix completion, ideally we would like to solve an NP-hard problem (1). However as in the case of CS, it turns out that the convex relaxation (2) of the NP-hard problem (1) can recover the matrix accurately. The price to be paid for convex relaxation is the increased number of equations (3) required to solve the problem. Now the question is can we improve upon this?

In CS, it has been shown that non-convex optimization of the following form (8) requires much less number of equations than (7):

$$\min ||x||_p \text{ subject to } y = Ax, \quad 0 < p \leq 1 \tag{8}$$

where $||x||_p = (\sum_i x_i^p)^{1/p}$

The number of equations needed to solve (5) is now about $m \geq C_1 k + pC_2 k\log n$ [7]. When p is small, the second term almost vanishes and the number of equations increases approximately linearly with k.

Our present work is motivated by this theoretical result in CS. We see from (6)–(8) that as the value of p increases, the number of required equations also increases. Intuitively, the same will be true for Matrix Completion. Therefore instead of solving the problem by convex optimization (2), this paper proposes to solve it by non-convex optimization (4).

On a related note, an empirical study in CS has shown that better recovery results are obtained when sophisticated blue-noise sampling schemes are employed instead of uniform random sampling [8]. This work shows that the same holds true for matrix completion as well.

3. Optimization algorithm

The problem is to solve (4). However, we relax the problem slightly and solve the following instead:

$$\min ||X||_{p*} \text{ subject to } ||Y - M_\Omega(X)||_F^2 \leq \varepsilon \tag{9}$$

where $||.||_F$ is the Frobenius norm of the matrix.

When $p=1$, the problem is convex, and for fractional values of p, it is non-convex. The solution from (9) can be made to reach that of (4) by making the mismatch ε small enough.

The following unconstrained Lagrangian version of (9) is easier to solve:

$$\min ||Y - M_\Omega(X)||_F^2 + \lambda ||X||_{p*} \tag{10}$$

The parameters λ and ε are related but in general there exists no analytical relation between them. In this work, we develop an optimization algorithm to solve (10). Later, we will show how to solve (9) iteratively via (10) by cooling λ.

A more convenient way to represent (10) is

$$\min ||y - Mx||_2^2 + \lambda ||X||_{p*} \tag{11}$$

where $x_{n^2}\times 1$ is the vectorized version of the matrix $X_{n\times n}$ formed by row/concatenation; $M : \Re^{n^2} \to \Re^m$ is a restriction operator, which has diagonal elements as ones at the sampling locations; $y_{m\times 1}$ is the vector of measurements.

We propose a novel non-convex optimization problem (9). Therefore there is no fast algorithm to solve it. The rest of Section 3 is devoted in deriving a fast algorithm to solve the said problem.

3.1. Optimization transfer

The problem (11) does not have a closed form solution and needs to be solved iteratively. At each iteration, instead of solving (11) we minimize the following:

$$G_k(x) = ||y-Mx||_2^2 + (x-x_k)^T(I-M^TM)(x-x_k) + \lambda||X||_{p*}$$

where $x_k = x_{k-1} + M^T(y-Mx_{k-1})$

This technique is called Majorization–Minimization (MM). It replaces the original function (11) by another function that is easier to minimize, and is also guaranteed to reach the solution of (11) iteratively. The MM approach has been used for deriving CS optimization algorithms [9]. Owing to lack of space, we skip the intermediate steps. The interested reader can follow a derivation similar to this in [10].

With re-arrangement of terms, $G_k(x)$ can be expressed as

$$G_k(x) = ||x-x_k||_2^2 - x_k^Tx_k + y^Ty + x_{k-1}^T(I-M^TM)x_{k-1} + \lambda||X||_{p*}$$

As all the terms apart from the first term are independent of x, we can thus minimize the following instead:

$$G_k'(x) = ||x-x_k||_2^2 + \lambda||X||_{p*} \tag{12}$$

Now, x and x_k are vectorized forms of matrices. The following property of singular value decomposition holds in general:

$$||v_{n^2\times 1}||_2^2 = ||V_{n\times n}||_F^2 = ||\Sigma_{r\times r}||_F^2 = ||s_{r\times 1}||_2^2$$

where v is the vectorized version of the matrix V, Σ is the singular value matrix of V, and s is a vector formed by the diagonal elements of Σ (singular values of V). Now both x and x_k have the same right and left singular vectors. Using this property, minimizing (12) is the same as minimizing the following:

$$G_k''(s) = ||s-s_k||_2^2 + \lambda||s||_p^p \tag{13}$$

where s and s_k are the singular values of the matrices corresponding to x and x_k, respectively.

Expression (13) is decoupled as

$$G_k''(s) = \sum_i (s(i)-s(i)_k)^2 + \lambda s(i)^p \tag{14}$$

It is now easy to minimize (14) by differentiating it term-wise. Skipping the mathematical manipulations for lack of space, it can be shown that (14) is minimized by

$$s = signum(s_k)\max\left(0, |s_k| - \frac{\lambda}{2}p|s_k|^{p-1}\right) \tag{15}$$

This is actually a modified version of the famous iterative soft thresholding algorithm used in Compressed Sensing for l_1-norm minimization.

The shrinkage algorithm for the unconstrained optimization problem (11) is shown below.

3.1.1. Shrinkage algorithm

1. $x_k = x_{k-1} + M^T(y-Mx_{k-1})$.
2. Form the matrix X_k by reshaping x_k.
3. SVD: $X_k = U\Sigma V^T$.
4. Soft threshold the singular values: $\hat{\Sigma} = soft(diag(\Sigma), (\lambda/2)p|s_k|^p)$.
5. $X_{k+1} = U\hat{\Sigma}V^T$. Form x_{k+1} by vectorizing X_{k+1}.
6. Update: $k=k+1$ and return to step 1.

3.2. Constrained optimization via cooling

Our target is to solve the constrained problem (9). So far, we have discussed how to solve the unconstrained version (10). Although the parameters λ and σ are related, the relation is not analytical and is nearly impossible to find for all practical cases. In this paper, we solve this problem by adopting a cooling technique as employed by [11].

3.2.1. Cooling algorithm

Initialize: $x_0=0$; $\lambda < \max(M^Tx)$
Choose a decrease factor (*DecFac*) for cooling λ
Outer loop: while[1] $||y-Mx||_2 > \varepsilon$
Inner loop: while[2] $J_k - J_{k+1}/J_k + J_{k+1} \geq Tol$
Compute objective function for current iterate: $J_k = ||y-Mx_k||_2^2 + \lambda||X_k||_{p*}$
Minimize J_k by the shrinkage algorithm described earlier.
Compute objective function for next iterate: $J_{k+1} = ||y-Mx_{k+1}||_2^2 + \lambda||X_{k+1}||_{p*}$
End while[2] (inner loop ends)
$\lambda = \lambda \times DecFac$
End while[1] (outer loop ends)

The cooling algorithm consists to two loops. The inner loop is for minimizing the unconstrained problem (11) for a particular value of λ. This loop either runs for a fixed number of iterations or exits when the relative change in the objective function $(J_k - J_{k+1})/(J_k + J_{k+1})$ is less than the tolerance level. The outer loops reduce the value of λ. The outer loop continues as long as the value of λ does not reach a minimum value or as long as $||y-Mx||_2 > \varepsilon$.

4. Blue-noise sampling

In [2–5], the number of entries required for perfect estimation of the rank-deficient matrix was based on the assumption that the samples from the matrix have been selected uniformly at random. Ideally one needs a sampling strategy that does not completely omit any row or column of the matrix. Uniform random sampling satisfies this property up to a logarithmic factor.

In this work, our quest is to find better sampling strategies for matrix completion strategies than the popular uniform random sampling. It has been found that compressed sensing techniques for seismic imaging [8] yield better results when blue-noise sampling schemes (farthest point sampling and Poisson disk sampling) are used instead of uniform random sampling. Inspired by the empirical results in [8], we investigate the performance of three blue-noise sampling strategies (Quasi-Random,

Quasi-Crystal and Farthest Point) over the popular uniform Random sampling, for the matrix completion problem.

1. *Random sampling*. It aims at global variability. It can be implemented either from a 2D uniform random distribution or as a random walk on a unit square. It is the most popular sampling method in the signal reconstruction literature.
2. *Quasi-Random sampling*. It is similar to random sampling but leads to less cluttered samples that are more evenly distributed in space. It can be efficiently generated by a Halton or Sobol sequence. It has been used sometimes for signal reconstruction problems mainly as an alternative to Shannon sampling.
3. *Quasi-Crystal sampling*. It aims at local regularity, but is not a popular choice owing to its high computational requirement: it has been shown to be more optimal than random sampling for signal reconstruction problems.
4. *Farthest Point sampling*. It aims at spatial uniformity. The main idea is to repeatedly place the next sample point in the middle of the least known area of the sampling domain. In this work, fast marching algorithm is used to generate the samples.

5. Experimental results

The proposed non-convex matrix completion algorithm is compared with two state-of-the-art convex matrix completion algorithms – Singular Value Thresholding (SVT) [12] and the Fixed Point Continuation (FPC) [13]. The size of the matrix is fixed as 100×100 for all experiments. The value of p for the proposed non-convex algorithm is fixed at 0.1 if not mentioned otherwise. First, we show empirically that the algorithm converges quite fast.

Fig. 1 plots how the objective function decreases as the number of iterations increases. It is shown for two scenarios. In the first case, the rank of the matrix is 5 and in the second case it is 10. In both cases, 30% of the matrix entries have been sampled.

Tables 1 and 2 show the comparative results between the three algorithms (SVT, FPC, and Proposed). Normalized Mean Squared Error (NMSE) is used as the metric for comparison. Uniform random sampling is used to select the matrix entries. All the experiments are repeated 100 times and the average results are reported.

Tables 1 and 2 conclusively prove that our non-convex matrix completion algorithm is superior to the convex algorithms. The results can be understood intuitively. The matrix has 10,000 entries. The number of degrees of freedom for a rank 10 matrix is 1900. At 10% sampling the number of samples is less than the number of degrees of freedom; hence the recovery results are extremely poor. At 20% sampling, the number of samples is nearly the same as the degree of freedom; one cannot really expect good results at this sampling ratio. When the sampling ratio is 30% or higher, our proposed algorithm shows marked improvement (around 10 times) over the others.

From the beginning of this work we have been emphasizing on the development of 'fast' matrix completion algorithm. To show how our algorithm fairs over those of the others we give the times (in seconds) required to complete a typical matrix completion problem of rank 10 with 30% sampled entries.

Table 1
NMSE for different algorithms (rank=5 and p=0.1).

Algorithm	Sampling ratio (%)			
	10	20	30	40
SVT	35,479	0.6506	0.2079	0.0631
FPC	0.7654	0.3125	0.0726	0.0106
Proposed	0.7155	0.0295	0.0054	0.0000

Table 2
NMSE for different algorithms (rank=10 and p=0.1).

Algorithm	Sampling ratio (%)			
	10	20	30	40
SVT	40,162	0.9527	0.7838	0.4154
FPC	0.9335	0.7036	0.4420	0.1964
Proposed	0.9757	0.6724	0.0434	0.0022

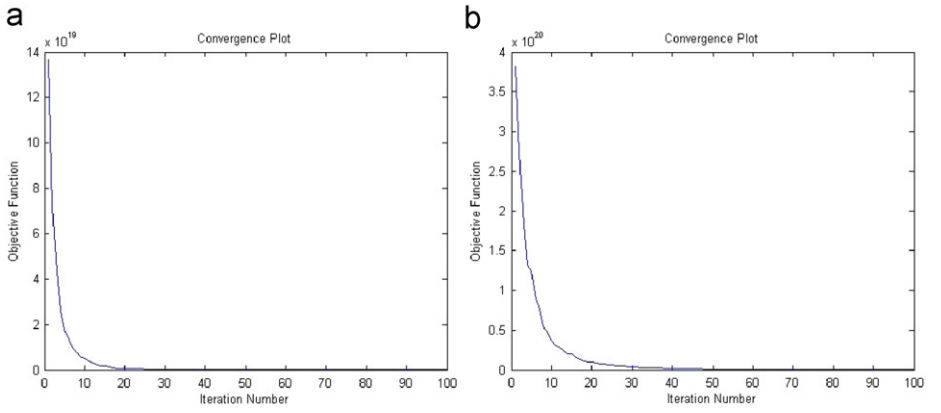

Fig. 1. Plot of objective function against number of iterations: (a) rank=5, sampling=30% and (b) rank=10, sampling=30%.

Table 3
Time (s) required by different algorithms (rank=10 and p=0.1).

Algorithm	Sampling ratio (%)			
	10	20	30	40
SVT	3.42	288.39	327.58	442.23
FPC	171.33	186.19	278.42	592.50
Proposed	16.52	14.39	12.58	8.78

Table 4
Variation in NMSE with p.

Value of p	Sampling ratio (%)			
	10	20	30	40
0.2	0.9839	0.6462	0.0642	0.0040
0.4	0.9962	0.6004	0.0640	0.0051
0.6	1.0268	0.5537	0.0719	0.0092
0.8	1.0225	0.5654	0.1586	0.0197
1.0	0.9421	0.6865	0.4686	0.2448

Table 5
NMSE for different sampling schemes.

Sampling strategy	Sampling ratio (%)			
	10	20	30	40
Random	0.9757	0.6724	0.0434	0.0022
QR	0.9484	0.6338	0.0164	0.0002
QC	0.9671	0.5810	0.0065	0.0000
FP	0.9594	0.5995	0.0133	0.0003

QR, Quasi-Random; QC, Quasi-Crystal; FP, Farthest Point.

At 10% sampling the SVT takes very less time but on the other hand gives extremely poor results (see Tables 1 and 2). For the cases where the algorithms give decent results, the proposed algorithm is over an order of magnitude faster than the others (Tables 3 and 4).

We now show how the recovery results vary when the value of p changes. The results are shown only for uniform random sampling. The rank of the matrix is 10.

The previous experiments were carried out for uniform random sampling. Now, we show that better recovery results can be obtained when blue-noise sampling schemes are employed instead of uniform random sampling. For the results in Table 5, the rank of the matrix is 10.

6. Conclusion

Theoretical studies in matrix completion are mostly based on uniform random sampling schemes followed by convex recovery algorithms. This paper goes a step beyond the current theory. We propose to solve the matrix completion problem via a non-convex optimization algorithm and show empirically that an order of magnitude improvement can be achieved over existing state-of-the-art convex optimization algorithms. Since this is a new optimization problem, there is no existing algorithm to solve it. We derive a fast numerical algorithm in Section 3. The implementation of the proposed algorithm along with a few others can be downloaded from the author's Matlab Central page [14].

This paper also shows that blue-noise sampling schemes like Quasi-Random, Quasi-Crystal, and Farthest Point give better matrix recovery results than the popular uniform random sampling.

References

[1] B. Savas, D. Lindgren, Rank reduction and volume minimization approach to state-space subspace system identification, Signal Processing 86 (11) (2006) 3275–3285.
[2] S.O. Al-Jazzara, D.C. McLernonb, M.A. Smadi, SVD-based joint azimuth/elevation estimation with automatic pairing, Signal Processing 90 (5) (2010) 1669–1675.
[3] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
[4] E.J. Candès, T. Tao, The power of convex relaxation: near-optimal matrix completion, IEEE Trans. Inf. Theory 56 (5) (2010) 2053–2080.
[5] B. Recht, A Simpler Approach to Matrix Completion arXiv:0910.0651v2.
[6] D.L. Donoho, For most large underdetermined systems of linear equations, the minimal ell-1 norm solution is also the sparsest solution, Comm. Pure and Applied Maths. 59 (6) (2006) 797–829.
[7] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, IEEE Conf. Acoust., Speech Signal Process. (2008).
[8] G. Tang, R. Shahidi, F.J. Herrmann, J. Ma, Higher dimensional blue-noise sampling schemes for curvelet-based seismic data recovery, Soc. Explor. Geophys. (2009).
[9] J. Oliveira, J. Bioucas-Dias, M. Figueiredo, Adaptive total variation image deblurring: a majorization–minimization approach, Signal Processing 89 (9) (2009) 1683–1693.
[10] 〈http://cnx.org/content/m32168/latest/〉.
[11] T. Lin, F.J. Herrmann, Compressed extrapolation, Geophysics 72 (5) (2007) 77–93.
[12] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2008) 1956–1982.
[13] S. Ma, D. Goldfarb, L. Chen, Fixed Point and Bregman Iterative Methods for Matrix Rank Minimization, arXiv:0905.1643v2.
[14] 〈http://www.mathworks.com/matlabcentral/fileexchange/26395-matrix-completion-via-thresholding〉.