THRESHOLD SELECTION FOR NOISY MATRIX COMPLETION

Victor Solo

School of Electrical Engineering and Telecommunications
University of New South Wales
Sydney, AUSTRALIA

ABSTRACT

While noise free matrix completion has a considerable history recent interest has centered on a noisy version of the problem. We consider a nuclear norm penalised least squares formulation. And by applying the SURE method, we develop for the first time, an automatic procedure for selecting the penalty parameter. We illustrate its use with some simulation results.

Index Terms— denoising, matrix completion, model selection

1. INTRODUCTION

Matrix completion involves the recovery of a matrix of interest from possibly noisy measurements of some (hence incomplete information) its elements. Matrix completion has already a considerable history with applications in areas such as; collaborative filtering (e.g. predicting customer preferences from a subset of available expressed preferences) [1]; remote sensing e.g. filling out a correlation matrix from some of its entries [2] ; computer vision [3] and others.

An early survey is [4]. Matrix completion is an ill-conditioned inverse problem and so there are many kinds of matrix completion problems roughly corresponding to the kind of regularization used to make them solvable. Low rank and positive definiteness [5] are two of the most common. A survey of another special class of problems is in [6]; in fact Prof Hogben's web page has a wealth of references and links.

Much of the recent interest in the Electrical Engineering community derives from a new emphasis on noisy problems. And a number of interesting algorithms have been developed for the noisy problem in recent years involving application of a constraint on the nuclear norm, [7],[8],[9],[10],[11].

But resolution of an ill-conditioned inverse problem invariably involves the choice of a tuning parameter such as a penalty parameter in a penalised least squares formulation. And it is a well known feature of ill-conditioned inverse problems that the penalty parameter has a dramatic effect on the estimate. Thus automatic choice of the penalty parameter is important in practice. And so far there does not appear to be any work on choosing such penalty parameters for matrix completion problems. In this paper we develop for apparently the first time, an automatic procedure for selecting such a penalty parameter. Our approach is based on the SURE methodology [12].

The remainder of the paper is organised as follows. In section 2 we review the nuclear norm penalised least squares matrix completion problem describing a particular algorithm due to [10]. In section 3 we develop the SURE selector. An illustrative simulation is given in section 4. Conclusions are in section 5.

Notation. In the sequel we will deal with $m \times n$ matrices and denote $R = \{(i,j) : 1 \leq i \leq m; 1 \leq j \leq n\}$.
R_o will be a subset of R of $'$observed$'$ indices
of dimension k_o.
$R_\rho = R - R_o$ consists of remaining indices
of dimension k_ρ.
Given an $m \times n$ matrix A we specify that,
A^o is A but entries with indices in R_ρ are set to zero.
A^ρ is A but entries with indices in R_o are set to zero.
Thus we can write $A = A^o + A^\rho$.
Also $A_{i,j}$ is the (i,j) element of A.
Frobenius norm: $\| A \|_F^2 = trace(A^T A) = \Sigma_{i,j} A_{i,j}^2$.
SVD = singular value decomposition.

Heaviside step function: $H(x) = \left\{ \begin{smallmatrix} 1, x \geq 0 \\ 0, x < 0 \end{smallmatrix} \right\}$; Dirac Delta $\delta(x)$;
Kronecker Delta $\delta_{u,v} = \left\{ \begin{smallmatrix} 1, u = v \\ 0, u \neq v \end{smallmatrix} \right\}$.

2. NOISY MATRIX COMPLETION

We suppose available a noisy measurement $Y_{m \times n}$ of a matrix $M_{m \times n}$ of low rank $r << min(m,n)$ according to the model,

$$Y_{i,j} = M_{i,j} + \epsilon_{i,j}, (i,j) \in R_o$$

where $\epsilon_{m \times n}$ is a matrix whose entries are independent Gaussian white noises of variance σ^2.

The measurement Y is incomplete in that no measurments are available for indices outside R_o i.e. for the $'$completing$'$ indices in R_ρ. Thus $Y = Y^o, Y^\rho = 0$.

The matrix completion problem is to provide estimates for the unobserved entries M^ρ in M thereby completing the matrix M. This is possible because of the low rank assumption

on M. Indeed under certain conditions [13] among others has shown this is possible with high probability.

We consider the following penalised least squares approach,

$$min_M J(M) : J(M) = \frac{1}{2} \parallel Y^o - M^o \parallel_F^2 + h \parallel M \parallel_* \tag{2.1}$$

where, to be clear, $\parallel Y^o - M^o \parallel_F^2 = \Sigma_{(i,j)\in R_o}(Y_{i,j} - M_{i,j})^2$ and where $\parallel M \parallel_*$ is the so-called nuclear norm and is just the sum of the singular values of M. It is known that this optimization problem is convex [14].

A number of algorithms have been developed in recent years for solving this problem [8] particularly using semi-definite programming but these run into problems with high dimensional data. Most interesting then, is the very simple and very fast algorithm developed recently by [10]. It is based on the following remarkable result of [7].

If $X_{m\times n}$ has rank r and SVD, $X = U_{m\times r}D_{r\times r}V^T$, with $D = diag(d_t)$ then the solution to,

$$min_M \frac{1}{2} \parallel X - M \parallel_F^2 + h \parallel M \parallel_*$$

is $M^* = S_h(X) = UD_hV^T$ where, $D_h = diag((d_j - h)^+)$ and using the heaviside step function $H(\cdot)$,

$$(d_j - h)^+ = \max(d_j - h, 0) = (d_j - h)H(d_j - h) \tag{2.2}$$

Note the presence of soft thresholding of the singular values. Thus h can be interpreted as a thresholding parameter.

Using this result [10] develop the following algorthim for solving (2.1) which they call SOFT-IMPUTE (SI). Initialize $M_{old} = 0$.

a Compute $M_{new} = S_h(Y^o + M^\rho_{old})$

b If $\parallel M_{new} - M_{old} \parallel_F / \parallel M_{old} \parallel_F < tol$ stop.

c Else set $M_{old} = M_{new}$; go to a.

We use the SI algorthm in our computations below.

Denote the converged value by $\hat{M}$. Then at convergence,

$$\hat{M} = S_h(Y^o + \hat{M}^\rho) = S_h(Y + \hat{M}^\rho) \tag{2.3}$$

This is in fact an optimality condition and any convergent algorithm must satisfy it at convergence.

3. PENALTY PARAMETER SELECTION WITH SURE

Here we develop a SURE (Stein$^{'}$s unbiassed risk estimator) procedure for choosing h. Common methods for tuning parameter selection [15] such as AIC,BIC are not obviously applicable here since they require the tuning parameter be discrete valued e.g. a model order or dimension; whereas here the tuning parameter h is continuous valued. We thus consider the SURE method suggested in two special cases by [16],[17] and then for general use by this author [12] followed by numerous applications (see references in [18]). Other more recent papers using SURE include [19],[20],[21].

The idea behind SURE is that ideally we would like to choose h to minimize the risk $R_h = E \parallel M^o - \hat{M}^o \parallel_F^2$. But this cannot be computed since we don$^{'}$t know M^o. However it is possible to find an empirically computable unbiassed estimator of R_h, namely $\hat{R}_h$ and we minimize that instead. Because $\hat{R}_h$ is unbiassed it$^{'}$s minimizer should, on average, produce a good value for h. In practice we plot $\hat{R}_h$ for a minimum in h.

The SURE, $\hat{R}_h$ has the traditional form for a tuning parameter selector i.e. residual sum of squares plus complexity penalty, and is given by the general expression [12]

$$\hat{R}_h = \parallel E^o \parallel_F^2 + 2\sigma^2\tau_h \tag{3.1}$$

where E^o is the residual $E^o = Y^o - \hat{M}^o$ and,

$$\tau_h = \Sigma_{(u,v)\in R_o} \frac{\partial \hat{M}^o_{u,v}}{\partial Y_{u,v}} \tag{3.2}$$

Our task now is to compute the derivatives in τ_h; but this is not straightforward, there being no explicit expression for $\hat{M}$.

3.1. Getting τ_h

Instead we differentiate through the optimality equation (2.3). To facilitate that we rewrite it as,

$$\hat{M} = S_h(A), A = Y^o + \hat{M}^\rho \tag{3.3}$$

Then using the chain rule we find,

$$\frac{\partial \hat{M}}{\partial Y_{u,v}} = \Sigma_{k,l} W^{k,l} \frac{\partial A_{k,l}}{\partial Y_{u,v}}, (u,v) \in R_o$$

where we have introduced, $W^{k,l} = \frac{\partial S_h(A)}{\partial A_{k,l}}$. We discuss its computation below. Now using (3.3) we find,

$$\begin{aligned} \frac{\partial \hat{M}}{\partial Y_{u,v}} &= \Sigma_{k,l} W^{k,l} (\delta_{k,u}\delta_{v,l} + \frac{\partial \hat{M}^\rho_{(k,l)}}{\partial Y_{u,v}}) \\ &= W^{u,v} + \Sigma_{(a,b)\in R_\rho} W^{a,b} \frac{\partial \hat{M}^\rho_{(a,b)}}{\partial Y_{u,v}} \end{aligned}$$

where $\delta_{k,i}$ is Kronecker$^{'}$s delta (see notation). In component form this becomes,

$$\frac{\partial \hat{M}_{k,l}}{\partial Y_{u,v}} = W^{u,v}_{k,l} + \Sigma_{(a,b)\in R_\rho} W^{a,b}_{k,l} \frac{\partial \hat{M}^\rho_{(a,b)}}{\partial Y_{u,v}}, (k,l) \in R \tag{3.4}$$

Restricting the components to R_ρ gives,

$$\frac{\partial \hat{M}^\rho_{(c,d)}}{\partial Y_{u,v}} = W^{u,v}_{c,d} + \Sigma_{(a,b)\in R_\rho} W^{a,b}_{c,d} \frac{\partial \hat{M}^\rho_{(a,b)}}{\partial Y_{u,v}}, (c,d) \in R_\rho \tag{3.5}$$

and these equations can be solved to deliver $\frac{\partial \hat{M}^{\rho}_{(a,b)}}{\partial Y_{u,v}}$ as we shortly describe. Then from (3.4),(3.2) we also have,

$$\begin{aligned}\frac{\partial \hat{M}^{o}_{u,v}}{\partial Y_{u,v}} &= W^{u,v}_{u,v} + \Sigma_{(a,b)\in R_\rho} W^{a,b}_{u,v} \frac{\partial \hat{M}^{\rho}_{(a,b)}}{\partial Y_{u,v}}, (u,v) \in R_o \\ \Rightarrow \tau_h &= \Sigma_{(u,v)\in R_o}(W^{u,v}_{u,v} + \Sigma_{(a,b)\in R_\rho} W^{a,b}_{u,v} \frac{\partial \hat{M}^{\rho}_{(a,b)}}{\partial Y_{u,v}})\end{aligned} \tag{3.6}$$

Returning to (3.5) we vectorise the equations. Put

$$\nabla^{u,v} = vec(\frac{\partial \hat{M}^{\rho}_{(a,b)}}{\partial Y_{u,v}}), (a,b) \in R_\rho$$

$$W^{u,v}_{\rho} = vec(W^{u,v}_{c,d}), (c,d) \in R_\rho$$

Then (3.5) can be rewritten,

$$\nabla^{u,v} = W^{u,v}_{\rho} + K\nabla^{u,v} \Rightarrow \nabla^{u,v} = (I-K)^{-1} W^{u,v}_{\rho} \tag{3.7}$$

where $K_{k_\rho \times k_\rho}$ is obtained from $W^{a,b}_{c,d}$ by running through the indices in the order a,b,c,d for $(a,b) \in R_\rho$ and $(c,d) \in R_\rho$.

To sum up, we use (3.7) to get $\nabla^{u,v}$ which is then used in (3.6) to get τ_h and so SURE=$\hat{R}_h$ from (3.1).

3.2. Getting $W^{k,l}$

Recall that if A has SVD, $A = QDP^T = \Sigma \sigma_t q_t p_t^T$ where q_t, p_u are the columns respectively of Q, P then $S_h(A) = \Sigma \sigma^*_t q_t p_t^T$ where from (2.2) $\sigma^*_t = (\sigma_t - h)H(\sigma_t - h)$. So,

$$\frac{\partial S_h(A)}{\partial A_{k,l}} = \Sigma[\frac{\partial \sigma^*_t}{\partial A_{k,l}} q_t p_t^T + \sigma^*_t \frac{\partial q_t}{\partial A_{k,l}} p_t^T + \sigma^*_t q_t \frac{\partial p_t^T}{\partial A_{k,l}}]$$

We next note that, since $\frac{d}{dx}H(x) = \delta(x)$,

$$\begin{aligned}\frac{\partial \sigma^*_t}{\partial A_{k,l}} &= \frac{\partial}{\partial A_{k,l}}(\sigma_t - h)H(\sigma_t - h) \\ &= \frac{\partial \sigma_t}{\partial A_{k,l}}(H(\sigma_t - h) + (\sigma_t - h)\delta(\sigma_t - h)) \\ &= \frac{\partial \sigma_t}{\partial A_{k,l}} H(\sigma_t - h)\end{aligned}$$

Recalling that the singular values are ordered from largest to smallest we now deduce that if $r = \#$ singular values with $\sigma_t > h$ then, $\sigma^*_t = \sigma_t - h, 1 \le u \le r$ and

$$\frac{\partial S_h(A)}{\partial A_{k,l}} = \Sigma_1^r[\frac{\partial \sigma_t}{\partial A_{k,l}} q_t p_t^T + \sigma^*_t \frac{\partial q_t}{\partial A_{k,l}} p_t^T + \sigma^*_t q_t \frac{\partial p_t^T}{\partial A_{k,l}}]$$

We now need to compute the derivatives of singular-vectors and singular values of A with respect to the elements of the matrix A. By using related formulae for such derivatives for positive definite matrices given in [22] it can be shown

$$\begin{aligned}W^{k,l} &= \frac{\partial S_h(A)}{\partial A_{k,l}} = Q_r F^{k,l} P_r^T \\ F^{k,l} &= F^{k,l}_q + (F^{k,l}_p)' + diag(\frac{\partial \sigma_t}{\partial A_{k,l}})\end{aligned}$$

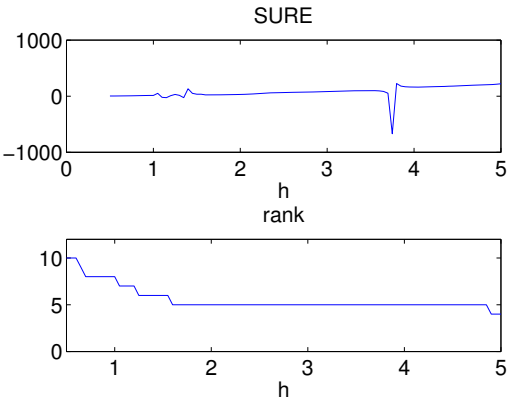

Fig. 1. SURE and Rank (case I)

where Q_r consists of the first r columns of Q and similarly for P_r; and then $F^{k,l}$ is an $r \times r$ matrix. The columns of $F^{k,l}_q$ are of the form $(\sigma_t - h)\phi^{(t)}$ and those of $F^{k,l}_p$ of the form $(\sigma_t - h)\psi^{(t)}$ where,

$$\phi^{(t)}_s = \{\frac{0}{\frac{\omega_s \alpha_s + (1-\omega_s)\beta_s}{\sigma_t - \sigma_s}}, \begin{matrix} s = t \\ s \ne t \end{matrix}\} \tag{3.8}$$

$$\psi^{(t)}_s = \{\frac{0}{\frac{\omega_s \beta_s + (1-\omega_s)\alpha_s}{\sigma_t - \sigma_s}}, \begin{matrix} s = t \\ s \ne t \end{matrix}\} \tag{3.9}$$

and, $\omega_s = \frac{\sigma_t}{\sigma_t + \sigma_s}, \alpha_s = q_{k,s}p_{l,t}, \beta_s = q_{k,t}p_{l,s}$ also, $\frac{\partial \sigma_t}{\partial A_{k,l}} = q_{k,t}p_{l,t}$.

4. SIMULATION STUDY

We use the same class of examples as in [10]. The model is $M = U_{m\times r}V^T_{n\times r}$ where the entries of $U, V, \epsilon/\sigma$ are independent Gaussian white noises of unit variance. The observed indices are selected at random leaving an average fraction f missing. The noise variance σ^2 is selected indirectly by specifiying a signal to noise variance ratio (SNVR) as $SNVR = var(M)/var(\epsilon)$ ([10] use a nonstandard definition of SNR; our SNVR is the square of their SNR).

We consider two cases each with $m = n = 30$ and case I has rank 5 while case II has rank 8. Each case has 50% missing values and SNVR = 50. ([10] considered mostly higher values of SNVR).

In Fig.1 for case I is a plot of SURE=$\hat{R}_h$ as well as the recovered rank. We see that $h = 3.75$ minimizes SURE; with corresponding correct rank of 5.

In Fig.2 for case II is a plot of SURE =$\hat{R}_h$ as well as the recovered rank. We see that $h = 4.7$ minimizes SURE with a corresponding rank of 9 very near the correct value of 8.

The spiking behaviour evident in both plots is due to the division by differences between close singular values in (3.8),(3.9).

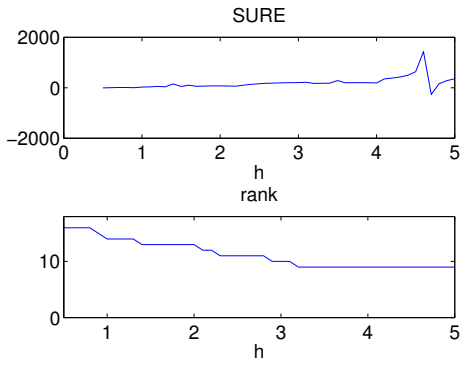

Fig. 2. SURE and Rank (case II)

5. CONCLUSION

In this paper, using the SURE methodology, we have developed for the first time, an automatic penalty parameter selector for noisy matrix completion when solved by nuclear norm penalised least squares. The computations are relatively straightforward relying on SVD and matrix inversion. The method was illustrated with two simulation examples.

6. REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. on World Wide Web (WWW 2001)*, 2001, pp. 285–295.

[2] E. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, pp. 925–936, 2010.

[3] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: Application to sfm," *IEEE Trans. PAMI*, vol. 26, pp. 1051– 1063, 2004.

[4] M. Laurent, "Matrix completion problems," in *The Encyclopedia of Optimization, vol. III*, 2001, pp. 221–229.

[5] J.Y. Choi, L.M. Dealba, L. Hogben, B.M. Kivunge, S.K. Nordstrom, and M. Shedenhelm, "The non-negative p0-matrix completion problem," *Electronic Jl. Lin. Alg.*, vol. 10, pp. 46–59, 2003.

[6] S.M. Fallat and L. Hogben, "The minimum rank of symmetric matrices described by a graph: A survey," Tech. Rep., Iowa State Univ, Dept Mathematics, 2007.

[7] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comp. Math.*, 2008.

[8] J.F. Cai, E.J. Candes, and Z Shen, "A singular value thresh- olding algorithm for matrix completion," Tech. Rep., Stanford Univ. Dept. Statistics, 2008.

[9] R.H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," Tech. Rep., Stanford Univ. Dept. Elec. Eng., 2009.

[10] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," Tech. Rep., Stanford Univ., Dept. Statistics, 2009.

[11] S. Gandy and I. Yamada, "Alternating minimization techniques for the efficient recovery of a sparsely corrupted low-rank matrix," in *Proc. IEEE ICASSP, Dallas, Texas*. IEEE, 2010, pp. 3638– 3641.

[12] V. Solo, "A SURE-fired way to choose smoothing parameters in ill-conditioned inverse problems," in *Proc. IEEE ICIP96, Lausanne, Switzerland*. IEEE, 1996, pp. vol III, pp 89–92, IEEE Press.

[13] B. Recht, "A simpler approach to matrix completion," Tech. Rep., Univ. Wisc. Madison, Dept. Comp. Sci., 2009.

[14] M. Fazel, *Matrix Rank Minimization with Applications*, Stanford Univ, PhD Thesis, Dept. Statistics, 2002.

[15] H. Linhart and W. Zucchini, *Model selection*, J. Wiley, New York, 1986.

[16] D.L. Donoho, I.M. Johnstone, J.C. Hoch, and A.S. Stern, "Maximum entropy and the nearly black object," *Jl. Roy. Stat. Soc. B*, 1992.

[17] M. Hudson and T.Lee, "Maximum likelihood reconstruction and choice of smoothing parameter in deconvolution of image data subject to poisson noise," Tech. Rep., Dept. Statistics, Macquarie Univ., 1994.

[18] MO Ulfarsson and V Solo, "Dimension estimation in noisy PCA with sure and random matrix theory.," *IEEE Trans Signal Processing*, vol. 56, pp. 5804–5816, 2008.

[19] V Solo and MO Ulfarsson, "Threshold selection for group sparsity," in *Proc IEEE ICASSP, Dallas, Texas*. IEEE, 2010, pp. 3754–3757.

[20] S. Ramani, T. Blu, and M. Unser, "Monte-carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE. Trans. Sig. Proc.*, vol. 17, pp. 1540–1554, 2008.

[21] Y. Eldar, "Generalized sure for exponential families: Applications to regularization," *IEEE. Trans. Sig. Proc.*, vol. 57, pp. 471–481, 2008.

[22] J.R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, J. Wiley, New York, 1999.