Probabilistic Low-Rank Matrix Completion with Adaptive Spectral Regularization Algorithms

François Caron

Department of Statistics, Oxford

STATLEARN 2014, Paris
April 7, 2014

Joint work with Adrien Todeschini, Marie Chavent (INRIA, U. of Bordeaux)

Outline

Matrix Completion

- Netflix prize
- 480k users and 18k movies providing 1-5 ratings
- 99% of the ratings are missing
- Objective: predict missing entries in order to make recommendations

Movies

Users

$$\begin{pmatrix} \mathbf{1} & \times & \times & \times & \mathbf{4} & \dots \\ \times & \times & \times & \mathbf{1} & \times & \dots \\ \mathbf{2} & \times & \mathbf{5} & \times & \times & \dots \\ \mathbf{3} & \mathbf{1} & \times & \mathbf{4} & \times & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

...

Matrix Completion

Objective

Complete a matrix $\boldsymbol{X}$ of size $\boldsymbol{m} \times \boldsymbol{n}$ from a subset of its entries

Applications

- Recommender systems
- Image inpainting
- Imputation of missing data

$$\begin{pmatrix} \square & \times & \times & \times & \square & \dots \\ \times & \times & \times & \square & \times & \dots \\ \square & \times & \square & \times & \times & \dots \\ \square & \square & \times & \square & \times & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

Matrix Completion

- Potentially large matrices (each dimension of order $\mathbf{10^4 - 10^6}$)
- Very sparsely observed (1%-10%)

Low rank Matrix Completion

- Assume that the complete matrix $\boldsymbol{Z}$ is of low rank

$$\underbrace{\boldsymbol{Z}}_{m\times n} \simeq \underbrace{\boldsymbol{A}}_{m\times k}\underbrace{\boldsymbol{B}^{T}}_{k\times n}$$

with $\boldsymbol{k \ll \min(m,n)}$.

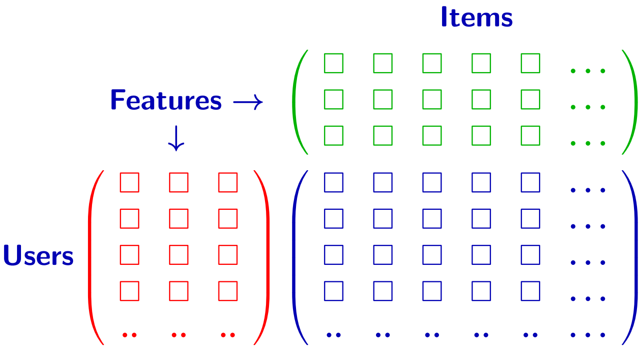
$$\begin{array}{cc} & \begin{pmatrix} \square & \square & \square & \square & \square & \dots \\ \square & \square & \square & \square & \square & \dots \\ \square & \square & \square & \square & \square & \dots \end{pmatrix} \\ \begin{pmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \vdots & \vdots & \vdots \end{pmatrix} & \begin{pmatrix} \square & \square & \square & \square & \square & \dots \\ \square & \square & \square & \square & \square & \dots \\ \square & \square & \square & \square & \square & \dots \\ \square & \square & \square & \square & \square & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots \end{pmatrix} \end{array}$$

Low rank Matrix Completion

- Assume that the complete matrix Z is of low rank

$$\underbrace{Z}_{m\times n} \simeq \underbrace{A}_{m\times k}\underbrace{B^T}_{k\times n}$$

with $k \ll \min(m, n)$.



Low rank Matrix Completion

- Let $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ be the subset of observed entries
- For $(i, j) \in \Omega$

$$X_{ij} = Z_{ij} + \varepsilon_{ij},\ \varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

where $\sigma^2 > 0$

Low rank Matrix Completion

- Optimization problem

$$\underset{Z}{\text{minimize}} \quad \underbrace{\frac{1}{2\sigma^2}\sum_{(i,j)\in\Omega}(X_{ij}-Z_{ij})^2}_{-\text{ loglikelihood}} + \underbrace{\lambda \operatorname{rank}(Z)}_{\text{penalty}}$$

  where $\lambda > 0$ is some regularization parameter.
- Non-convex
- Computationally hard for general subset Ω

Low rank Matrix Completion

- Matrix completion with nuclear norm penalty

$$\underset{Z}{\text{minimize}} \quad \underbrace{\frac{1}{2\sigma^2} \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2}_{-\text{ loglikelihood}} + \underbrace{\lambda \|Z\|_*}_{\text{penalty}}$$

  where $\|Z\|_*$ is the nuclear norm of Z, or the sum of the singular values of Z.

- Convex relaxation of the rank penalty optimization

[Fazel, 2002, Candès and Recht, 2009, Candès and Tao, 2010]

Low rank Matrix Completion

Soft-Impute algorithm

- Start with an initial matrix $Z^{(0)}$
- At each iteration $t = 1, 2, \ldots$
  - Replace the missing elements in X with those in $Z^{(t-1)}$
  - Perform a soft-thresholded SVD on the completed matrix, with shrinkage λ to obtain the low rank matrix $Z^{(t)}$

[Mazumder et al., 2010]

Low rank Matrix Completion

Soft-Impute algorithm

- Soft-thresholded SVD yields a low-rank representation
- Each iteration decreases the value of the nuclear norm objective function towards its minimum
- Various strategies proposed to scale the algorithm to problems where n, m of order 10^6
- Same shrinkage applied to all singular values

[Mazumder et al., 2010]

Contributions

- Probabilistic interpretation of the nuclear norm objective function
  - Maximum A Posteriori estimation assuming exponential priors on the singular values
  - Soft-Impute = Expectation-Maximization algorithm
- Construction of alternative non-convex objective functions building on hierarchical priors
  - Bridge the gap between the rank penalty and the nuclear norm penalty
  - EM: Adaptative algorithm that iteratively adjusts the shrinkage coefficients for each singular value
  - Similar to adaptive lasso in multivariate regression
  - Numerical results show the interest of the approach on various datasets

Outline

Nuclear Norm penalty

- Complete matrix $\boldsymbol{X}$
- Nuclear norm objective function

$$\underset{\boldsymbol{Z}}{\text{minimize}} \quad \frac{1}{2\sigma^2}||\boldsymbol{X} - \boldsymbol{Z}||_F^2 + \lambda\, ||\boldsymbol{Z}||_*$$

  where $||\cdot||_F^2$ is the Frobenius norm

- Global solution given by a soft-thresholded SVD

$$\widehat{\boldsymbol{Z}} = \mathbf{S}_{\lambda\sigma^2}(\boldsymbol{X})$$

  where $\mathbf{S}_\lambda(\boldsymbol{X}) = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{D}}_\lambda\widetilde{\boldsymbol{V}}^T$ with
  $\widetilde{\boldsymbol{D}}_\lambda = \text{diag}((\widetilde{d}_1 - \lambda)_+, \ldots, (\widetilde{d}_r - \lambda)_+)$
  and $t_+ = \max(t, 0)$.

[Cai et al., 2010, Mazumder et al., 2010]

Nuclear Norm penalty

- Maximum A Posteriori (MAP) estimate

$$\widehat{Z} = \arg\max_{Z} \left[\log p(X|Z) + \log p(Z)\right]$$

under the prior

$$p(Z) \propto \exp\left(-\lambda \left\|Z\right\|_*\right)$$

where $Z = UDV^T$ with $D = \text{diag}(d_1, d_2, \ldots, d_r)$, and

$$U, V \overset{\text{iid}}{\sim} \text{Haar uniform prior on unitary matrices}$$
$$d_i \overset{\text{iid}}{\sim} \text{Exp}(\lambda)$$

Hierarchical adaptive spectral penalty

- Each singular value has its own random shrinkage coefficient
- Hierarchical model, for each singular value $i = 1, \dots, r$

$$\begin{aligned} d_i | \gamma_i &\sim \mathrm{Exp}(\gamma_i) \\ \gamma_i &\sim \mathrm{Gamma}(a, b) \end{aligned}$$

- Marginal distribution over d_i:

$$p(d_i) = \int_0^\infty \mathrm{Exp}(d_i; \gamma_i)\, \mathrm{Gamma}(\gamma_i; a, b) d\gamma_i = \frac{ab^a}{(d_i + b)^{a+1}}$$

Pareto distribution with heavier tails than exponential distribution

[Todeschini et al., 2013]

Hierarchical adaptive spectral penalty

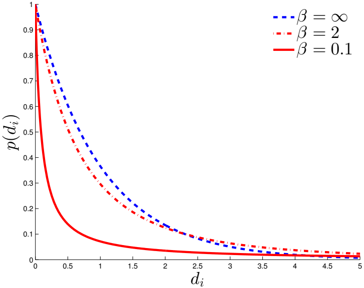


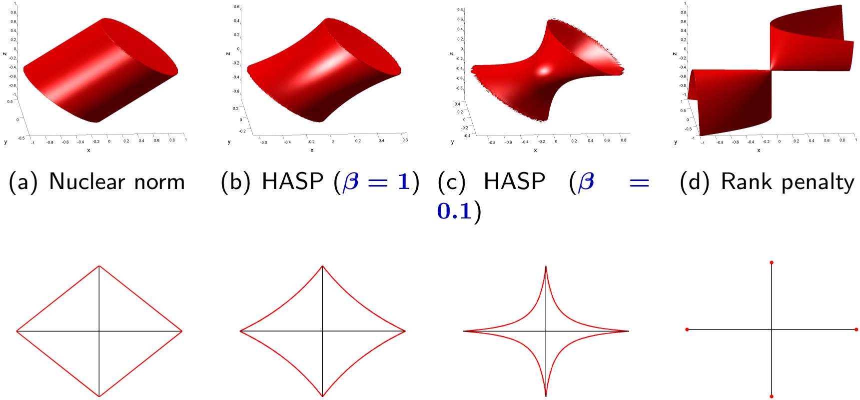
Figure: Marginal distribution $p(d_i)$ with $a = b = \beta$

- HASP penalty

$$pen(Z) = -\log p(Z) = \sum_{i=1}^{r} (a+1)\log(b + d_i)$$

- Admits as special case the nuclear norm penalty $\lambda||Z||_*$ when $a = \lambda b$ and $b \to \infty$.

Hierarchical adaptive spectral penalty



(a) Nuclear norm (b) HASP ($\beta = 1$) (c) HASP ($\beta = 0.1$) (d) Rank penalty

(e) ℓ_1 norm (f) HAL ($\beta = 1$) (g) HAL ($\beta = 0.1$) (h) ℓ_0 norm

Figure: Top: Manifold of constant penalty, for a symmetric 2×2 matrix $Z = [x, y; y, z]$ for (a) the nuclear norm, hierarchical adaptive spectral penalty with $a = b = \beta$ (b) $\beta = 1$ and (c) $\beta = 0.1$, and (d) the rank penalty. Bottom: contour of constant penalty for a diagonal matrix $[x, 0; 0, z]$, where one recovers the classical (e) lasso, (f-g) hierarchical lasso and (h) ℓ_0 penalties.

EM algorithm for MAP estimation

Expectation Maximization (EM) algorithm to obtain a MAP estimate

$$\widehat{Z} = \arg\max_{Z} \left[\log p(X|Z) + \log p(Z)\right]$$

i.e. to minimize

$$L(Z) = \frac{1}{2\sigma^2} \|X - Z\|_F^2 + \sum_{i=1}^{r} (a+1) \log(b + d_i)$$

EM algorithm for MAP estimation

- Latent variables: $\gamma = (\gamma_1, \ldots, \gamma_r)$
- E step:

$$\begin{aligned} Q(Z, Z^*) &= \mathbb{E}\left[\log(p(X, Z, \gamma)) | Z^*, X\right] \\ &= C - \frac{1}{2\sigma^2} \|X - Z\|_F^2 - \sum_{i=1}^r \omega_i d_i \end{aligned}$$

where $\omega_i = \mathbb{E}[\gamma_i | d_i^*] = \frac{a+1}{b+d_i^*}$.

EM algorithm for MAP estimation

- M step:

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2\sigma^2} \|X - Z\|_F^2 + \sum_{i=1}^{r} \omega_i d_i \tag{1}$$

(1) is an adaptive spectral penalty regularized optimization problem, with weights $\omega_i = \frac{a+1}{b+d_i^*}$.

$$d_1^* \geq d_2^* \geq \ldots \geq d_r^*$$

$$\Rightarrow 0 \leq \omega_1 \leq \omega_2 \leq \ldots \leq \omega_r \tag{2}$$

Given condition (2), the solution is given by a weighted soft-thresholded SVD

$$\widehat{Z} = \mathbf{S}_{\sigma^2\omega}(X) \tag{3}$$

where $\mathbf{S}_\omega(X) = \widetilde{U}\widetilde{D}_\omega\widetilde{V}^T$ with
$\widetilde{D}_\omega = \text{diag}((\widetilde{d}_1 - \omega_1)_+, \ldots, (\widetilde{d}_r - \omega_r)_+)$.

[Gaïffas and Lecué, 2011]

EM algorithm for MAP estimation

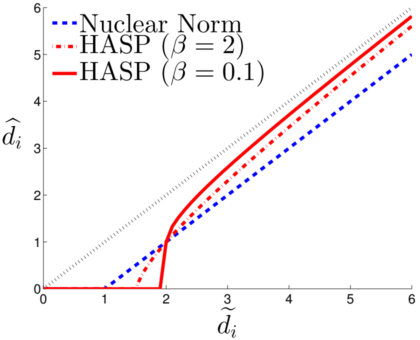


Figure: Thresholding rules on the singular values $\widetilde{d}_i$ of X

The weights will penalize less heavily higher singular values, hence reducing bias.

Low rank estimation of complete matrices

Hierarchical Adaptive Soft Thresholded (HAST) algorithm for low rank estimation of complete matrices

Initialize $Z^{(0)}$. At iteration $t \geq 1$

- For $i = 1, \ldots, r$, compute the weights $\omega_i^{(t)} = \frac{a+1}{b+d_i^{(t-1)}}$
- Set $Z^{(t)} = \mathrm{S}_{\sigma^2 \omega^{(t)}}(X)$
- If $\frac{L(Z^{(t-1)}) - L(Z^{(t)})}{L(Z^{(t-1)})} < \varepsilon$ then return $\widehat{Z} = Z^{(t)}$

- Admits soft-thresholded SVD operator as a special case when $a = b\lambda$ and $b = \beta \to \infty$.

Settings

- Parametrization:
  - We set $\boldsymbol{b} = \boldsymbol{\beta}$ and $\boldsymbol{a} = \boldsymbol{\lambda\beta}$ where $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ are tuning parameters that can be chosen by cross-validation.
  - Possible to estimate $\boldsymbol{\sigma}$ within the EM algorithm.
- Initialization:
  - Initialization with the soft thresholded SVD with parameter $\boldsymbol{\sigma^2\lambda}$

Outline

Matrix completion

- Only a subset $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ of the entries of the matrix X is observed.
- Subset operators:

$$P_\Omega(X)(i,j) = \begin{cases} X_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

$$P_\Omega^\perp(X)(i,j) = \begin{cases} 0 & \text{if } (i,j) \in \Omega \\ X_{ij} & \text{otherwise} \end{cases}$$

- Same prior over Z
- MAP estimate is obtained by minimizing

$$L(Z) = \frac{1}{2\sigma^2} \|P_\Omega(X) - P_\Omega(Z)\|_F^2 + (a+1) \sum_{i=1}^{r} \log(b + d_i)$$

EM algorithm for MAP estimation

- Latent variables: γ and $P_\Omega^\perp(X)$

Hierarchical Adaptive Soft Impute (HASI) algorithm for matrix completion

Initialize $Z^{(0)}$. At iteration $t \geq 1$

- For $i = 1, \ldots, r$, compute the weights $\omega_i^{(t)} = \frac{a+1}{b+d_i^{(t-1)}}$
- Set $Z^{(t)} = \mathbf{S}_{\sigma^2\omega^{(t)}}\left(P_\Omega(X) + P_\Omega^\perp(Z^{(t-1)})\right)$
- If $\frac{L(Z^{(t-1)})-L(Z^{(t)})}{L(Z^{(t-1)})} < \varepsilon$ then return $\widehat{Z} = Z^{(t)}$

EM algorithm for MAP estimation

- HASI algorithm admits the Soft-Impute algorithm as a special case when $a = \lambda b$ and $b = \beta \to \infty$. In this case, $\omega_i^{(t)} = \lambda$ for all i.
- When $\beta < \infty$, the algorithm adaptively updates the weights so that to penalize less heavily higher singular values.

Initialization

- Non-convex objective function - different initializations may lead to different modes.
- We set $\boldsymbol{a = \lambda b}$ and $\boldsymbol{b = \beta}$ and initialize the algorithm with the Soft-Impute algorithm with regularization parameter $\boldsymbol{\sigma^2 \lambda}$.

Scaling

- Similarly to the Soft-Impute algorithm, the computationally bottleneck is the computation of the weighted soft-truncated SVD

$$\mathbf{S}_{\boldsymbol{\sigma^2\omega^{(t)}}}\left(\boldsymbol{P_\Omega(X) + P_\Omega^\perp(Z^{(t-1)})}\right)$$

- For large matrices, one can resort to the PROPACK algorithm.
- Efficiently computes the truncated SVD of the "sparse + low rank" matrix

$$\boldsymbol{P_\Omega(X) + P_\Omega^\perp(Z^{(t-1)})} = \underbrace{\boldsymbol{P_\Omega(X) - P_\Omega(Z^{(t-1)})}}_{\text{sparse}} + \underbrace{\boldsymbol{Z^{(t-1)}}}_{\text{low rank}}$$

and can thus handle large matrices.

[Larsen, 2004]

Outline

Simulated data

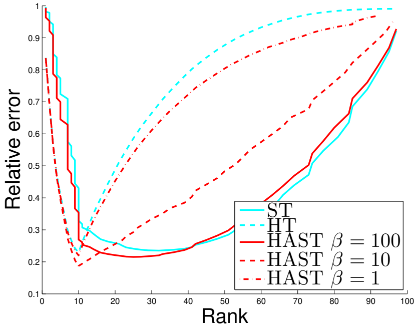
Procedure

- We generate matrices $Z = AB^T$ of low rank q where A and B are Gaussian matrices of size $m \times q$ and $n \times q$, $m = n = 100$ and add Gaussian noise with $\sigma = 1$.
- The signal to noise ratio is defined as $\text{SNR} = \sqrt{\frac{\text{var}(Z)}{\sigma^2}}$.
- For the HASP penalty, we set $a = \lambda\beta$ and $b = \beta$.
- Grid of 50 values of the regularization parameter λ
- Metric

$$err = \frac{||\widehat{Z} - Z||_F^2}{||Z||_F^2} \qquad \text{and} \qquad err_{\Omega^\perp} = \frac{||\widehat{P}_\Omega^\perp(\widehat{Z}) - P_\Omega^\perp(Z)||_F^2}{||P_\Omega^\perp(Z)||_F^2}$$

Simulated data

Complete case



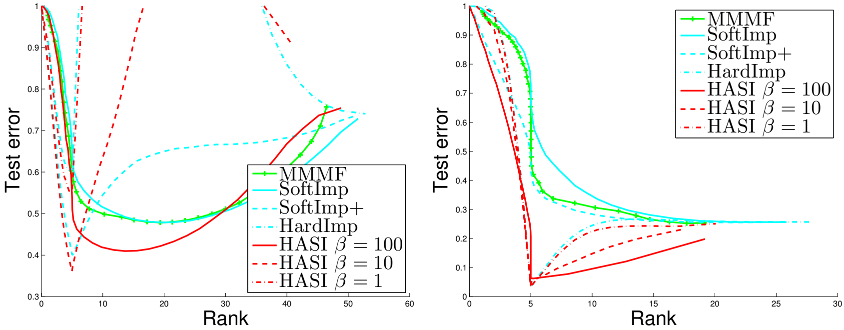
(a) SNR=1; Complete; rank=10

Figure: Test error w.r.t. the rank obtained by varying the value of the regularization parameter $\boldsymbol{\lambda}$.

- The HASP penalty provides a bridge/tradeoff between the nuclear norm and the rank penalty.
- For example, value of $\boldsymbol{\beta = 10}$ show a minimum at the true rank $\boldsymbol{q = 10}$ as HT, but with a lower error when the rank is overestimated.

Simulated data

Incomplete case



(a) SNR=1; 50% missing; rank=5 (b) SNR=10; 80% missing; rank=5

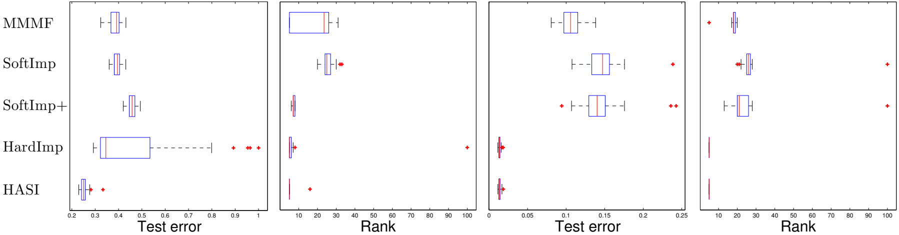
Figure: Test error w.r.t. the rank obtained by varying the value of the regularization parameter $\boldsymbol{\lambda}$, averaged over 50 replications.

- Similar behavior is observed, with the HASI algorithm attaining a minimum at the true rank $\boldsymbol{q = 5}$.

Simulated data

Incomplete case

We then remove 20% of the observed entries as a validation set to estimate the regularization parameters. We use the unobserved entries as a test set.



(a) SNR=1; 50% miss. (b) SNR=1; 50% miss. (c) SNR=10; 80% miss. (d) SNR=10; 80% miss.

Figure: Boxplots of the test error and ranks obtained over 50 replications.

Collaborative filtering examples (Jester)

Procedure

- We randomly select two ratings per user as a test set, and two other ratings per user as a validation set to select the parameters λ and β.
- The results are computed over four values $\beta = 1000, 100, 10, 1$.
- We compare the results of the different methods with the Normalized Mean Absolute Error (NMAE)

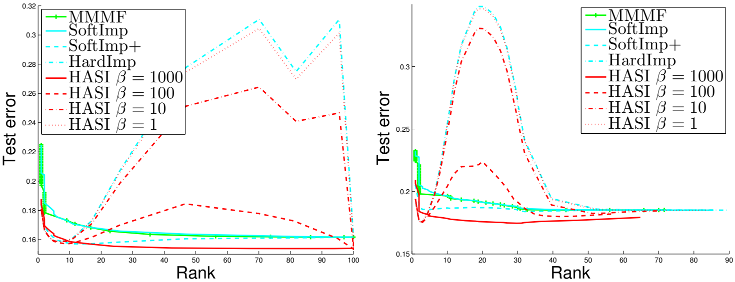
$$\text{NMAE} = \frac{\frac{1}{card(\Omega_{test})} \sum_{(i,j)\in\Omega_{test}} |X_{ij} - \widehat{Z}_{ij}|}{\max(X) - \min(X)}$$

Collaborative filtering examples (Jester)

Table: Results on the Jester datasets, averaged over 10 replications

	Jester 1 $\mathbf{24983 \times 100}$ $\mathbf{27.5\%}$ miss.		Jester 2 $\mathbf{23500 \times 100}$ $\mathbf{27.3\%}$ miss.		Jester 3 $\mathbf{24938 \times 100}$ $\mathbf{75.3\%}$ miss.	
Method	NMAE	Rank	NMAE	Rank	NMAE	Rank
MMMF	0.161	95	0.162	96	0.183	58
Soft Imp	0.161	100	0.162	100	0.184	78
Soft Imp+	0.169	14	0.171	11	0.184	33
Hard Imp	0.158	7	0.159	6	0.181	4
HASI	**0.153**	100	**0.153**	100	**0.174**	30

Collaborative filtering examples (Jester)



(a) Jester 1 (b) Jester 3

Figure: NMAE w.r.t. the rank obtained by varying the regularization parameter $\boldsymbol{\lambda}$.

Collaborative filtering examples (MovieLens)

Procedure

- We randomly select 20% of the entries as a test set, and the remaining entries are split between a training set (80%) and a validation set (20%).
- For all the methods, we stop the regularization path as soon as the estimated rank exceeds $r_{max} = 100$.
- For the larger MovieLens 1M dataset, the precision, maximum number of iterations and maximum rank are decreased to $\epsilon = 10^{-6}$, $t_{max} = 100$ and $r_{max} = 30$.

Collaborative filtering examples (MovieLens)

Table: Results on the MovieLens datasets, averaged over 5 replications

Method	MovieLens 100k $\mathbf{943 \times 1682}$ $\mathbf{93.7\%}$ miss. NMAE	 Rank	MovieLens 1M $\mathbf{6040 \times 3952}$ $\mathbf{95.8\%}$ miss. NMAE	 Rank
MMMF	0.195	50	**0.169**	30
Soft Imp	0.197	156	0.176	30
Soft Imp+	0.197	108	0.189	30
Hard Imp	0.190	7	0.175	8
HASI	**0.187**	35	0.172	27

Outline

Conclusion and perspectives

- Conclusion:
  - Good results compared to several alternative low rank matrix completion methods.
  - Bridge between nuclear norm and rank regularization algorithms.
  - Can be extended to binary matrices
  - Non-convex optimization, but experiments show that initializing the algorithm with the Soft-Impute algorithm provides very satisfactory results.
  - Matlab code available online
- Perspectives:
  - Fully Bayesian approach
  - Larger datasets

Bibliography I

Cai, J., Candès, E., and Shen, Z. (2010).
A singular value thresholding algorithm for matrix completion.
SIAM Journal on Optimization, 20(4):1956–1982.

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

Candès, E. J. and Tao, T. (2010).
The power of convex relaxation: Near-optimal matrix completion.
Information Theory, IEEE Transactions on, 56(5):2053–2080.

Fazel, M. (2002).
Matrix rank minimization with applications.
PhD thesis, Stanford University.

Gaïffas, S. and Lecué, G. (2011).
Weighted algorithms for compressed sensing and matrix completion.
arXiv preprint arXiv:1107.1638.

Larsen, R. M. (2004).
Propack-software for large and sparse svd calculations.
Available online. URL http://sun. stanford. edu/rmunk/PROPACK.

Bibliography II

Mazumder, R., Hastie, T., and Tibshirani, R. (2010).
Spectral regularization algorithms for learning large incomplete matrices.
The Journal of Machine Learning Research, 11:2287–2322.

Todeschini, A., Caron, F., and Chavent, M. (2013).
Probabilistic low-rank matrix completion with adaptive spectral regularization algorithms.
In *Advances in Neural Information Processing Systems*, pages 845–853.