

Two stories of matrix optimization: Maximum-Entropy Sampling & Rank-Sparsity Decomposition

Jon Lee

Industrial and Operations Engineering Department
University of Michigan



Ann Arbor, Michigan

PGMO Seminar — June 2014

Toward Optimal Rank-Sparsity Decomposition

Jon Lee, Bai Zou. Optimal rank-sparsity decomposition. *Journal of Global Optimization*, 2013.

- Define the problem
- Motivation for the problem
- Motivating references
- Branch-and-bound toward global optimization
- Computational work
- A framework for full global optimization

The problem

- Given a real matrix C , our problem is to decompose C as $C = A + B$, where A is sparse and B has low rank.

The problem

- Given a real matrix C , our problem is to decompose C as $C = A + B$, where A is sparse and B has low rank.
- Additionally, *and critically*, we treat the situation where we are given convex sets of matrices \mathcal{A}, \mathcal{B} , which we require A and B to be chosen from.

The problem

- Given a real matrix C , our problem is to decompose C as $C = A + B$, where A is sparse and B has low rank.
- Additionally, *and critically*, we treat the situation where we are given convex sets of matrices \mathcal{A}, \mathcal{B} , which we require A and B to be chosen from.
- We assume that \mathcal{A} and \mathcal{B} are specified in convenient forms (e.g., via linear matrix inequalities, and for additional convenience, we prefer to assume that these sets are compact).

The problem

- Given a real matrix C , our problem is to decompose C as $C = A + B$, where A is sparse and B has low rank.
- Additionally, *and critically*, we treat the situation where we are given convex sets of matrices \mathcal{A}, \mathcal{B} , which we require A and B to be chosen from.
- We assume that \mathcal{A} and \mathcal{B} are specified in convenient forms (e.g., via linear matrix inequalities, and for additional convenience, we prefer to assume that these sets are compact).
- Also, for purely pedagogical purposes, we assume that our matrices are square.

The problem

- Given a real matrix C , our problem is to decompose C as $C = A + B$, where A is sparse and B has low rank.
- Additionally, *and critically*, we treat the situation where we are given convex sets of matrices \mathcal{A}, \mathcal{B} , which we require A and B to be chosen from.
- We assume that \mathcal{A} and \mathcal{B} are specified in convenient forms (e.g., via linear matrix inequalities, and for additional convenience, we prefer to assume that these sets are compact).
- Also, for purely pedagogical purposes, we assume that our matrices are square.
- Finally, for specificity, we look at the (NP-hard) version:







$$\min\{\gamma\|A\|_0 + r(B) : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\},$$

where γ is a parameter that can be varied.

Motivation

- Such decomposition problems arise in a number of settings, with the interpretation of the sparse and low-rank matrices depending on the application.
- In statistical model selection, the sparse matrix can correspond to a Gaussian graphical model, and the low-rank matrix summarizes the effect of a small number of latent, unobserved variables (corresponding to some systematic noise).
- In computational complexity theory, rigidity of a matrix is the least number of entries that must be changed in order to reduce the rank of the matrix below some given constant.
- In system identification, the low-rank matrix represents a system with a small “model order”, and the sparse matrix represents a system with a sparse “impulse response”.

References

-  Fazel, Maryam. Matrix rank minimization with applications. Ph.D. thesis, Stanford University, 2002.
-  Parrilo, Pablo A. The convex algebraic geometry of rank minimization, ISMP 2009 (Chicago), plenary talk, August 2009.
-  Recht, Benjamin; Fazel, Maryam; Parrilo, Pablo A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review* 52 (2010), no. 3, 471–501.
-  Mohan, Karthik; Fazel, Maryam. Iterative reweighted algorithms for matrix rank minimization, *Journal of Machine Learning Research* 13 (2012) 3441–3473.
-  Parrilo, Pablo A. Rank/sparsity minimization and latent variable graphical model selection, **IPAM Workshop (UCLA), October 2010**.
-  Chandrasekaran, Venkat; Sanghavi, Sujay; Parrilo, Pablo A.; Willsky, Alan S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization* 21 (2011), no. 2, 572–596.

Convex approximation

$$\min\{\gamma\|A\|_0 + r(B) : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

Convex approximation

$$\min\{\gamma\|A\|_0 + r(B) : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

$$= \min\{\gamma\|A\|_0 + \|\sigma(B)\|_0 : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

Convex approximation

$$\min\{\gamma\|A\|_0 + r(B) : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

$$= \min\{\gamma\|A\|_0 + \|\sigma(B)\|_0 : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

\approx

$$\min\{\gamma\|A\|_1 + \|\sigma(B)\|_1 : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

$$= \min\{\gamma\|A\|_1 + \|B\|_* : A + B = C, A \in \mathcal{A}, B \in \mathcal{B}\}$$

- $\|A\|_1 := \sum_{i,j} |a_{ij}|$ (the usual entrywise 1-norm)
- $\|B\|_* := \sum_k \sigma_k(B)$ (the nuclear norm)
- The nuclear norm is also known as the trace norm, the Ky Fan n -norm, and the Schatten 1-norm. It has the alternative definition (for a real matrix) as $\|B\|_* := \text{Tr}(\sqrt{B^t B})$

Semidefinite programming

Recast as

$$\begin{aligned} \min_{A, B, W_1, W_2, Z} \quad & \gamma e^t Z e + \frac{1}{2} (\text{Tr}(W_1) + \text{Tr}(W_2)) \\ \text{s.t.} \quad & \begin{pmatrix} W_1 & B \\ B^t & W_2 \end{pmatrix} \succeq 0, \\ & -Z \leq A \leq Z, \\ & A + B = C, \\ & A \in \mathcal{A}, B \in \mathcal{B}, \end{aligned}$$

where e is the n -vector of all ones.

Semidefinite programming

Recast as

$$\begin{aligned} \min_{A, B, W_1, W_2, Z} \quad & \gamma e^t Z e + \frac{1}{2} (\text{Tr}(W_1) + \text{Tr}(W_2)) \\ \text{s.t.} \quad & \begin{pmatrix} W_1 & B \\ B^t & W_2 \end{pmatrix} \succeq 0, \\ & -Z \leq A \leq Z, \\ & A + B = C, \\ & A \in \mathcal{A}, B \in \mathcal{B}, \end{aligned}$$

where e is the n -vector of all ones.

- CAUTION! convex *approximation* \neq convex *relaxation*.
- The 0-norm is not a norm! e.g., it is *invariant* under scaling.

Bounds to get bounds

We assume that:

- The maximum-norm $\|A\|_{\max} := \max\{|a_{ij}|\}$ is bounded on \mathcal{A} .
That is, we have a scalar $\alpha > 0$, so that

$$\|A\|_{\max} \leq \alpha, \quad \forall A \in \mathcal{A}.$$

Bounds to get bounds

We assume that:

- The maximum-norm $\|A\|_{\max} := \max\{|a_{ij}|\}$ is bounded on \mathcal{A} .
That is, we have a scalar $\alpha > 0$, so that

$$\|A\|_{\max} \leq \alpha, \quad \forall A \in \mathcal{A}.$$

Then we have

$$\frac{1}{\alpha} \|A\|_1 \leq \|A\|_0, \quad \forall A \in \mathcal{A}.$$

Bounds to get bounds

We assume that:

- The maximum-norm $\|A\|_{\max} := \max\{|a_{ij}|\}$ is bounded on \mathcal{A} . That is, we have a scalar $\alpha > 0$, so that

$$\|A\|_{\max} \leq \alpha, \quad \forall A \in \mathcal{A}.$$

Then we have

$$\frac{1}{\alpha} \|A\|_1 \leq \|A\|_0, \quad \forall A \in \mathcal{A}.$$

- The spectral norm $\sigma_1(B)$ is bounded on \mathcal{B} . That is, we have a scalar $\beta > 0$, so that

$$\sigma_1(B) \leq \beta, \quad \forall B \in \mathcal{B}.$$

Bounds to get bounds

We assume that:

- The maximum-norm $\|A\|_{\max} := \max\{|a_{ij}|\}$ is bounded on \mathcal{A} . That is, we have a scalar $\alpha > 0$, so that

$$\|A\|_{\max} \leq \alpha, \quad \forall A \in \mathcal{A}.$$

Then we have

$$\frac{1}{\alpha} \|A\|_1 \leq \|A\|_0, \quad \forall A \in \mathcal{A}.$$

- The spectral norm $\sigma_1(B)$ is bounded on \mathcal{B} . That is, we have a scalar $\beta > 0$, so that

$$\sigma_1(B) \leq \beta, \quad \forall B \in \mathcal{B}.$$

Then we have

$$\frac{1}{\beta} \|B\|_* \leq r(B), \quad \forall B \in \mathcal{B}.$$

A true convex *relaxation*

So we have the rigorous lower bound

$$\frac{\gamma}{\alpha} \|A\|_1 + \frac{1}{\beta} \|B\|_* \leq \gamma \|A\|_0 + r(B), \quad \forall A \in \mathcal{A}, B \in \mathcal{B},$$

which we can calculate by simply **tweaking** the SDP to

$$\begin{aligned} \min_{A, B, W_1, W_2, Z} \quad & \frac{\gamma}{\alpha} e^t Z e + \frac{1}{2\beta} (\text{Tr}(W_1) + \text{Tr}(W_2)) \\ \text{s.t.} \quad & \begin{pmatrix} W_1 & B \\ B^t & W_2 \end{pmatrix} \succeq 0, \\ & -Z \leq A \leq Z, \\ & A + B = C, \\ & A \in \mathcal{A}, B \in \mathcal{B}, \end{aligned}$$

where e is the n -vector of all ones.

Branch-and-bound

Now that we have a rigorous lower bound, we can build a branch-and-bound algorithm *toward* global optimization, by:

- devising an effective branching technique, compatible with the lower-bounding method;
- crafting a good upper-bounding heuristic;
- putting it all together (branching-object selection rule, subproblem selection rule, policy for running the heuristic, algorithms/software for lower and upper bound calculation).

Branching on the sparsity pattern

- Notation for a sparse matrix: For a subset S of $\{1, \dots, n\}^2$, $A_S : S \rightarrow \mathbb{R}$ is defined by $A_S(i, j) := a_{ij}$. Then we define $\|A_S\|_0 := |\{(i, j) \in S : a_{ij} \neq 0\}|$.

Branching on the sparsity pattern

- Notation for a sparse matrix: For a subset S of $\{1, \dots, n\}^2$, $A_S : S \rightarrow \mathbb{R}$ is defined by $A_S(i, j) := a_{ij}$. Then we define $\|A_S\|_0 := |\{(i, j) \in S : a_{ij} \neq 0\}|$.
- We consider partitions of $\{1, \dots, n\}^2$ into sets \mathcal{Z} (“zero”), \mathcal{N} (“nonzero”), \mathcal{U} (“unbranched”), and such a partition determines a *subproblem*:

$$\gamma|\mathcal{N}| + \min \{ \gamma\|A_{\mathcal{U}}\|_0 + r(B) : A + B = C, A \in \mathcal{A}, \\ B \in \mathcal{B}, a_{ij} = 0 \text{ for } (i, j) \in \mathcal{Z} \}.$$

- So, sparsity of A is enforced on the index set \mathcal{Z} , and nonzeros are charged for on the index set \mathcal{N} , *regardless of whether or not they are nonzero in A .*

Branching on the sparsity pattern

- Notation for a sparse matrix: For a subset S of $\{1, \dots, n\}^2$, $A_S : S \rightarrow \mathbb{R}$ is defined by $A_S(i, j) := a_{ij}$. Then we define $\|A_S\|_0 := |\{(i, j) \in S : a_{ij} \neq 0\}|$.
- We consider partitions of $\{1, \dots, n\}^2$ into sets \mathcal{Z} (“zero”), \mathcal{N} (“nonzero”), \mathcal{U} (“unbranched”), and such a partition determines a *subproblem*:

$$\gamma|\mathcal{N}| + \min \{ \gamma\|A_{\mathcal{U}}\|_0 + r(B) : A + B = C, A \in \mathcal{A}, \\ B \in \mathcal{B}, a_{ij} = 0 \text{ for } (i, j) \in \mathcal{Z} \}.$$

- So, sparsity of A is enforced on the index set \mathcal{Z} , and nonzeros are charged for on the index set \mathcal{N} , *regardless of whether or not they are nonzero in A* .
- Relax to an SDP (including $a_{ij} = 0$ for $(i, j) \in \mathcal{Z}$) as before.
- Tighten α (for subtrees) as the branching proceeds.

Branch-and-bound

- A list of active subproblems is maintained, each subproblem with an associated solution of its relaxation.

Branch-and-bound

- A list of active subproblems is maintained, each subproblem with an associated solution of its relaxation.
- An active subproblem, determined by a partition $\mathcal{Z}, \mathcal{N}, \mathcal{U}$ is selected, according to some rule, and a branching index $(i', j') \in \mathcal{U}$ is selected.

Branch-and-bound

- A list of active subproblems is maintained, each subproblem with an associated solution of its relaxation.
- An active subproblem, determined by a partition $\mathcal{Z}, \mathcal{N}, \mathcal{U}$ is selected, according to some rule, and a branching index $(i', j') \in \mathcal{U}$ is selected.
- Then the subproblem determined by the partition $\mathcal{Z}, \mathcal{N}, \mathcal{U}$ is replaced with two subproblems, determined by partitions: (i) $\mathcal{Z} + (i', j'), \mathcal{N}, \mathcal{U} - (i', j')$, and (ii) $\mathcal{Z}, \mathcal{N} + (i', j'), \mathcal{U} - (i', j')$.

Branch-and-bound

- A list of active subproblems is maintained, each subproblem with an associated solution of its relaxation.
- An active subproblem, determined by a partition $\mathcal{Z}, \mathcal{N}, \mathcal{U}$ is selected, according to some rule, and a branching index $(i', j') \in \mathcal{U}$ is selected.
- Then the subproblem determined by the partition $\mathcal{Z}, \mathcal{N}, \mathcal{U}$ is replaced with two subproblems, determined by partitions: (i) $\mathcal{Z} + (i', j'), \mathcal{N}, \mathcal{U} - (i', j')$, and (ii) $\mathcal{Z}, \mathcal{N} + (i', j'), \mathcal{U} - (i', j')$.
- For each of these subproblems, we solve the associated relaxation; if its optimal value is greater than our global upper bound (or if it is infeasible), then we discard the subproblem; otherwise we update our global lower bound (to the minimum objective value over all active relaxations), and we update our global upper bound (by evaluating the objective function of the solution of the relaxation, but according to the objective function of the original problem, and comparing it to the current global upper bound).

An upper-bounding heuristic

Motivated by the approach of:

- Mohan, K.; Fazel, M. Iterative reweighted algorithms for matrix rank minimization, *J. of Mach. Learn. Res.* 13 (2012) 3441–3473.

An upper-bounding heuristic

Motivated by the approach of:

- Mohan, K.; Fazel, M. Iterative reweighted algorithms for matrix rank minimization, *J. of Mach. Learn. Res.* 13 (2012) 3441–3473.

At any subproblem of a b&b search, choose $0 \leq p, q \leq 1$, and formulate the objective function

$$\min \{ \gamma \|A_{\mathcal{U}}\|_p + \|\sigma(B)\|_q : A + B = C, A \in \mathcal{A}, \\ B \in \mathcal{B}, a_{ij} = 0 \text{ for } (i, j) \in \mathcal{Z} \}.$$

An upper-bounding heuristic

Motivated by the approach of:

- Mohan, K.; Fazel, M. Iterative reweighted algorithms for matrix rank minimization, *J. of Mach. Learn. Res.* 13 (2012) 3441–3473.

At any subproblem of a b&b search, choose $0 \leq p, q \leq 1$, and formulate the objective function

$$\min \{ \gamma \|A_{\mathcal{U}}\|_p + \|\sigma(B)\|_q : A + B = C, A \in \mathcal{A}, \\ B \in \mathcal{B}, a_{ij} = 0 \text{ for } (i, j) \in \mathcal{Z} \}.$$

- For $p = q = 0$, we have the true objective.

An upper-bounding heuristic

Motivated by the approach of:

- Mohan, K.; Fazel, M. Iterative reweighted algorithms for matrix rank minimization, *J. of Mach. Learn. Res.* 13 (2012) 3441–3473.

At any subproblem of a b&b search, choose $0 \leq p, q \leq 1$, and formulate the objective function

$$\min \{ \gamma \|A_{\mathcal{U}}\|_p + \|\sigma(B)\|_q : A + B = C, A \in \mathcal{A}, \\ B \in \mathcal{B}, a_{ij} = 0 \text{ for } (i, j) \in \mathcal{Z} \}.$$

- For $p = q = 0$, we have the true objective.
- For $p = q = 1$, we have the nonsmooth but convex approximation used by Maryam, Pablo, et al.

An upper-bounding heuristic

Motivated by the approach of:

- Mohan, K.; Fazel, M. Iterative reweighted algorithms for matrix rank minimization, *J. of Mach. Learn. Res.* 13 (2012) 3441–3473.

At any subproblem of a b&b search, choose $0 \leq p, q \leq 1$, and formulate the objective function

$$\min \{ \gamma \|A_{\mathcal{U}}\|_p + \|\sigma(B)\|_q : A + B = C, A \in \mathcal{A}, \\ B \in \mathcal{B}, a_{ij} = 0 \text{ for } (i, j) \in \mathcal{Z} \}.$$

- For $p = q = 0$, we have the true objective.
- For $p = q = 1$, we have the nonsmooth but convex approximation used by Maryam, Pablo, et al.
- As a heuristic, we propose to find local optima, for choices of $0 < p, q < 1$ (smooth but nonconvex), using as a starting point the (global) optimum of the associated (convex) subproblem (having $p = q = 1$).
- Evaluate each local solution using the true objective and update the upper bound.

Software and experiments

- Bai Zou implemented our methods using the open-source software CVX in conjunction with Matlab.

Software and experiments

- Bai Zou implemented our methods using the open-source software `CVX` in conjunction with `Matlab`.
- Developed and maintained by Michael Grant and Stephen Boyd, `CVX` is a `Matlab`-based modeling system for “disciplined” convex optimization. `CVX` enables `Matlab` to be used as a modeling language; so constraint and objective functions can be easily specified using standard `Matlab` syntax. `CVX` is distributed under the GNU General Public License 2.0.

Software and experiments

- Bai Zou implemented our methods using the open-source software `CVX` in conjunction with `Matlab`.
- Developed and maintained by Michael Grant and Stephen Boyd, `CVX` is a `Matlab`-based modeling system for “disciplined” convex optimization. `CVX` enables `Matlab` to be used as a modeling language; so constraint and objective functions can be easily specified using standard `Matlab` syntax. `CVX` is distributed under the GNU General Public License 2.0.
- We have gathered some evidence for assessing the quality of earlier heuristic techniques.

Increasing the lower bound (2 hr.)

- We ran our b&b scheme for 2 hours on each of 24 randomly generated test problems.
- We generated examples of the form $C := A + \gamma DE^t$, where A is $n \times n$, p_A is the probability that an entry of A is set to 0, and D and E^t are $n \times r$.
- For these tests, our branching is reducing the gap by 40% on average.
- This is a significant average gap reduction, and it is a conservative estimate as we compare to our upper bound and not a confirmed optimal solution.
- We observe a large variation of the gap reduction, from 3% to 95%.
- Generally, we reduce more gap for larger γ . This is expected as larger γ puts more weight on the sparsity, and we only branch on sparsity.

Increasing the lower bound (2 hr.)

#	p_A	r	n	γ	initial gap	final gap	% reduction
1	0.3	5	10	0.01	3.1639	1.3491	57.36
2				0.1	6.8156	3.3858	50.32
3				1	17.3964	4.0558	76.69
4				10	162.4377	8.094	95.02
5	0.5	5	10	0.01	1.608	0.6289	60.89
6				0.1	8.3568	4.1565	50.26
7				1	11.5938	4.3658	62.34
8				10	80.4045	6.5351	91.87
9	0.3	3	10	0.01	3.466	1.8541	46.51
10				0.1	7.6032	4.479	41.09
11				1	16.7836	4.4365	73.57
12				10	132.1638	7.1222	94.61

Increasing the lower bound (2 hr.), cont'd

#	p_A	r	n	γ	initial gap	final gap	% reduction
13	0.3	10	15	0.01	3.5226	3.3812	4.01
14				0.1	26.3345	25.2141	4.25
15				1	49.2981	43.8938	10.96
16				10	371.1931	268.0499	27.79
17	0.5	10	15	0.01	3.9443	3.8272	2.97
18				0.1	31.6241	29.6816	6.14
19				1	41.1424	37.522	8.80
20				10	261.5299	209.9086	19.74
21	0.3	5	15	0.01	7.8066	6.2523	19.91
22				0.1	33.3281	26.1391	21.57
23				1	50.8432	46.8181	7.92
24				10	369.5897	258.4647	30.07
							40.19

Decreasing the upper bound

- We generated 4 random matrices, all with $n = 25$, with the different combinations of $p_A = .3, .5$ and $r = 10, 15$.
- And we made 4 instances from each, by setting $\gamma = 0.01, .1, 1, 10$. So this gave us 16 instances.
- Finally, for each of the instances, we started a limit b&b search and took as a starting point for the heuristic, the subproblem with the best (i.e., minimum) upper bound after $K = 0, 5, 20, 40$ relaxation subproblems were solved. So this gave us a final set of 64 instances.
- For each of these instances, we first calculated the baseline upper bound determined by the solution (A, B) of the SDP, evaluated with the exact objective function. Then, we used that solution as a starting point for finding local optima of the nonconvex program, for choices of $0 < p, q \leq 1$. Specifically, we took all 100 combinations of $p, q \in \{0.1, 0.2, \dots, 1.0\}$.
- We used the Matlab function `fmincon()` using the ‘interior-point’ algorithmic option.

Decreasing the upper bound

#	K	p_A	r	γ	UB	% Δ UB	#	K	p_A	r	γ	UB	% Δ UB
1	0	0.3	15	0.01	31.21	3.33	33	0	0.5	10	0.01	8.18	0.00
2				0.1	63.6	0.00	34				0.1	63.6	0.00
3				1	261	0.00	35				1	173	0.00
4				10	2305	0.00	36				10	1645	0.00
5	5	0.3	15	0.01	31.19	1.22	37	5	0.5	10	0.01	31.21	2.40
6				0.1	86.7	3.11	38				0.1	86.9	4.37
7				1	257	0.00	39				1	192	0.00
8				10	2195	0.00	40				10	1665	0.00
9	20	0.3	15	0.01	31.19	1.22	41	20	0.5	10	0.01	31.21	2.40
10				0.1	86.7	3.11	42				0.1	87	7.70
11				1	257	0.00	43				1	192	0.00
12				10	2195	0.00	44				10	1665	0.00

Decreasing the upper bound, cont'd

#	K	p_A	r	γ	UB	% Δ UB	#	K	p_A	r	γ	UB	% Δ UB
13	40	0.3	15	0.01	8.19	0.00	45	40	0.5	10	0.01	23.98	0.00
14				0.1	63.9	0.00	46				0.1	86.6	6.24
15				1	242	0.00	47				1	192	0.00
16				10	2195	0.00	48				10	1665	0.00
17	0	0.3	10	0.01	9.17	0.00	49	0	0.5	15	0.01	31.21	3.75
18				0.1	75.8	0.00	50				0.1	63	0.00
19				1	233	0.00	51				1	194	0.00
20				10	2225	0.00	52				10	1625	0.00
21	5	0.3	10	0.01	31.21	1.44	53	5	0.5	15	0.01	27.21	0.00
22				0.1	86.8	0.00	54				0.1	86.6	11.20
23				1	257	0.00	55				1	184	0.00
24				10	2345	0.00	56				10	1615	0.00

Decreasing the upper bound, cont'd

#	K	p_A	r	γ	UB	% Δ UB	#	K	p_A	r	γ	UB	% Δ UB
25	20	0.3	10	0.01	31.21	1.44	57	20	0.5	15	0.01	23.98	0.00
26				0.1	86.7	0.00	58				0.1	86.6	11.20
27				1	257	0.00	59				1	184	0.00
28				10	2345	0.00	60				10	1615	0.00
29	40	0.3	10	0.01	31.19	1.80	61	40	0.5	15	0.01	8.21	0.00
30				0.1	86.7	0.46	62				0.1	82.6	3.51
31				1	257	0.00	63				1	184	0.00
32				10	2345	0.00	64				10	1615	0.00

Improvements for 17 out of the 64 instances

Rank

- To get to a full specification of a global-optimization algorithm, we need to develop a global-optimization method for pure-rank subproblems.

Rank

- To get to a full specification of a global-optimization algorithm, we need to develop a global-optimization method for pure-rank subproblems.
- We can control the rank of B directly by writing B as the product of two matrices. That is, $B = DE^t$, where D and E have n rows but a small number of columns.

Rank

- To get to a full specification of a global-optimization algorithm, we need to develop a global-optimization method for pure-rank subproblems.
- We can control the rank of B directly by writing B as the product of two matrices. That is, $B = DE^t$, where D and E have n rows but a small number of columns.
- The cost is quadratic nonconvexity!

Rank

- To get to a full specification of a global-optimization algorithm, we need to develop a global-optimization method for pure-rank subproblems.
- We can control the rank of B directly by writing B as the product of two matrices. That is, $B = DE^t$, where D and E have n rows but a small number of columns.
- The cost is quadratic nonconvexity!
- Concretely, suppose that B has rank at most r , and we write the SVD $B = U\Sigma V^t$, with $U = [u_1 | \cdots | u_r]$ and $V = [v_1 | \cdots | v_r]$ both being $n \times r$ matrices having orthonormal columns, and $\Sigma = \text{Diag}(\sigma_1, \dots, \sigma_r)$ and $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$.

Rank

- To get to a full specification of a global-optimization algorithm, we need to develop a global-optimization method for pure-rank subproblems.
- We can control the rank of B directly by writing B as the product of two matrices. That is, $B = DE^t$, where D and E have n rows but a small number of columns.
- The cost is quadratic nonconvexity!
- Concretely, suppose that B has rank at most r , and we write the SVD $B = U\Sigma V^t$, with $U = [u_1 | \cdots | u_r]$ and $V = [v_1 | \cdots | v_r]$ both being $n \times r$ matrices having orthonormal columns, and $\Sigma = \text{Diag}(\sigma_1, \dots, \sigma_r)$ and $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$.
- Let $\bar{U} := U\sqrt{\Sigma}$ and $\bar{V} = V\sqrt{\Sigma}$, so that $B = \bar{U}\bar{V}^t = \sum_{l=1}^r \bar{u}_l \bar{v}_l^t$.

Rank

- To get to a full specification of a global-optimization algorithm, we need to develop a global-optimization method for pure-rank subproblems.
- We can control the rank of B directly by writing B as the product of two matrices. That is, $B = DE^t$, where D and E have n rows but a small number of columns.
- The cost is quadratic nonconvexity!
- Concretely, suppose that B has rank at most r , and we write the SVD $B = U\Sigma V^t$, with $U = [u_1 | \cdots | u_r]$ and $V = [v_1 | \cdots | v_r]$ both being $n \times r$ matrices having orthonormal columns, and $\Sigma = \text{Diag}(\sigma_1, \dots, \sigma_r)$ and $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$.
- Let $\bar{U} := U\sqrt{\Sigma}$ and $\bar{V} = V\sqrt{\Sigma}$, so that $B = \bar{U}\bar{V}^t = \sum_{l=1}^r \bar{u}_l \bar{v}_l^t$.
- Now, the restriction of $\sigma_1(B) \leq \beta$ relaxes to $\|u_l\|_2 \leq \sqrt{\beta}$ and $\|v_l\|_2 \leq \sqrt{\beta}$, for $l = 1, \dots, r$.

Rank, cont'd.

- We can control the rank more directly using indicator variables $y_l \in \{0, 1\}$, for $l = 1, \dots, r$, and having $\sum_{l=1}^r y_l \leq 1$.

Rank, cont'd.

- We can control the rank more directly using indicator variables $y_l \in \{0, 1\}$, for $l = 1, \dots, r$, and having $\sum_{l=1}^r y_l \leq 1$.
- The meaning of $y_l = 1$ is that the rank of B is restricted to be no more than l . So $y_1 = \dots = y_r = 0$ corresponds to B being all zero, and $y_r = 1$ corresponds to B being allowed to have rank up to r .

Rank, cont'd.

- We can control the rank more directly using indicator variables $y_l \in \{0, 1\}$, for $l = 1, \dots, r$, and having $\sum_{l=1}^r y_l \leq 1$.
- The meaning of $y_l = 1$ is that the rank of B is restricted to be no more than l . So $y_1 = \dots = y_r = 0$ corresponds to B being all zero, and $y_r = 1$ corresponds to B being allowed to have rank up to r .
- Then we impose the rank restriction on \bar{U} and \bar{V} , via

$$\|u_l\|_2 \leq \sqrt{\beta} \sum_{k=l}^r y_k, \quad l = 1, \dots, r;$$

$$\|v_l\|_2 \leq \sqrt{\beta} \sum_{k=l}^r y_k, \quad l = 1, \dots, r,$$

and we force $B = \bar{U} \bar{V}^t$ via the bilinear equations

$$b_{ij} = \sum_{l=1}^r \bar{u}_{il} \bar{v}_{jl}, \quad i, j = 1, \dots, n.$$

Rank, cont'd.

- We can control the rank more directly using indicator variables $y_l \in \{0, 1\}$, for $l = 1, \dots, r$, and having $\sum_{l=1}^r y_l \leq 1$.
- The meaning of $y_l = 1$ is that the rank of B is restricted to be no more than l . So $y_1 = \dots = y_r = 0$ corresponds to B being all zero, and $y_r = 1$ corresponds to B being allowed to have rank up to r .
- Then we impose the rank restriction on \bar{U} and \bar{V} , via

$$\|u_l\|_2 \leq \sqrt{\beta} \sum_{k=l}^r y_k, \quad l = 1, \dots, r;$$

$$\|v_l\|_2 \leq \sqrt{\beta} \sum_{k=l}^r y_k, \quad l = 1, \dots, r,$$

and we force $B = \bar{U} \bar{V}^t$ via the bilinear equations

$$b_{ij} = \sum_{l=1}^r \bar{u}_{il} \bar{v}_{jl}, \quad i, j = 1, \dots, n.$$

- We even can directly express $r(B)$ as $\sum_{l=1}^r l y_l$.

Rank, cont'd.

- We can control the rank more directly using indicator variables $y_l \in \{0, 1\}$, for $l = 1, \dots, r$, and having $\sum_{l=1}^r y_l \leq 1$.
- The meaning of $y_l = 1$ is that the rank of B is restricted to be no more than l . So $y_1 = \dots = y_r = 0$ corresponds to B being all zero, and $y_r = 1$ corresponds to B being allowed to have rank up to r .
- Then we impose the rank restriction on \bar{U} and \bar{V} , via

$$\|u_l\|_2 \leq \sqrt{\beta} \sum_{k=l}^r y_k, \quad l = 1, \dots, r;$$

$$\|v_l\|_2 \leq \sqrt{\beta} \sum_{k=l}^r y_k, \quad l = 1, \dots, r,$$

and we force $B = \bar{U} \bar{V}^t$ via the bilinear equations

$$b_{ij} = \sum_{l=1}^r \bar{u}_{il} \bar{v}_{jl}, \quad i, j = 1, \dots, n.$$

- We even can directly express $r(B)$ as $\sum_{l=1}^r l y_l$.
- We can treat the bilinear equations via standard spatial b&b (e.g., McCormick envelopes).

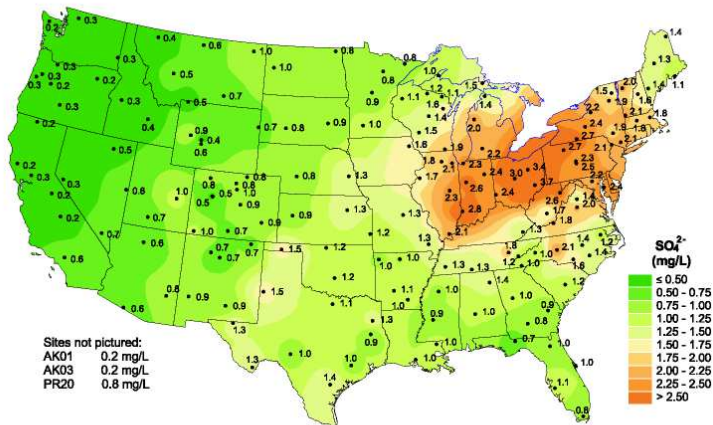
Maximum-Entropy Sampling

(with Ko and Queyranne; with Anstreicher, Fampa and Williams; with Burer; with Hoffman (and Williams); with \emptyset)

- Motivation: Environmental monitoring
- Define *entropy* and the problem: *Maximum-Entropy Sampling*
- Properties of entropy
- Branch-and-...
- Bounds
- Some references

Motivation: Environmental Monitoring

Sulfate ion concentration, 1994



National Atmospheric Deposition Program/National Trends Network
<http://nadp.sws.uiuc.edu>

Information = Disorder

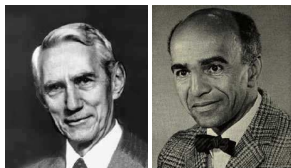
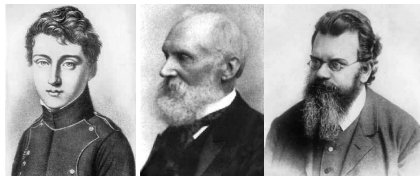
“Chance and chance alone has a message for us. Everything that occurs out of necessity, everything expected, repeated day in and day out, is mute. Only chance can speak to us. We read its message much as gypsies read the images made by coffee grounds at the bottom of a cup.”

- Milan Kundera (*The Unbearable Lightness of Being*)



Entropy

- R. Clausius (1865) — “entropy” (also Carnot and Kelvin in their versions of the 2nd law of thermodynamics), **arrow of time** (*“What then is time? If no one asks me, I know what it is. If I wish to explain it to him who asks, I do not know.” — St. Augustine*)
- L. Boltzmann (1877) — statistical mechanics
- C. Shannon (1948) — information theory
- D. Blackwell (1951) — statistics



Maximum-Entropy Sampling

$$N = \{1, 2, \dots, n\}$$

Random $Y_N = \{Y_j : j \in N\}$ with continuous density g_N

Goal: Choose $S \subset N$, with $|S| = s$, so that observing Y_S maximizes the “information” obtained about Y_N .

Entropy: $h(S) := -E[\ln g_S(Y_S)]$.

Nice Properties of Entropy

- **Submodularity:** $h(S \cup T) + h(S \cap T) \leq h(S) + h(T)$
[Another Talk on Approximation Algorithms]
- The **Gaussian** distribution **maximizes** the **entropy** for a given covariance matrix C
- Gaussian case: $h(S) = k_s + k \ln \det C[S, S]$
- **Conditional Additivity:**

$$h(N) = \overbrace{h(S)}^{\max} + \overbrace{h(N \setminus S|S)}^{\min}$$

(justifies our objective function)

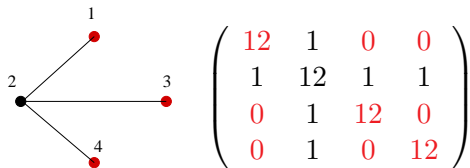
- **Change coordinate systems:** Entropy **difference** is $\log \det(\text{Jacobian of transformation})$
- **Complementation:**
 $\ln \det C[S, S] = \ln \det C + \ln \det C^{-1}[N \setminus S, N \setminus S]$

Not-So-Nice Property

Proposition [Ko, Lee, Queyranne]. The maximum-entropy sampling problem is **NP-Hard** (even for the Gaussian diagonally-dominant case)

Proof:

- **INDEPENDENT SET:** Does a simple undirected graph G on n vertices have an independent set of vertices of cardinality s ?
- Let $C := A(G) + 3nI$



(KLQ) Branch ...

- **Fixing j out of S :**

⇒ Strike out row and column j : $C[N, N] \rightarrow$

$$C[N - j, N - j]$$

- **Fixing j in S :**

⇒ Schur complement of $C[j, j]$: $C[N, N] \rightarrow$

$$C[N - j, N - j] - C[N - j, j]C^{-1}[j, j]C[j, N - j]$$

(and solution/bounds are shifted by $\ln C[j, j]$).

... and Bound

- Lower bounds: Greedy, local-search, rounding heuristics
- Upper bounds:
 - ▶ Spectral based bounds
 - ★ Ko, Lee, Queyranne '95 (original B&B and spectral bound)
 - ★ Lee '98 (extension to side constraints)
 - ★ Hoffman, Lee & Williams '01 (spectral partition bounds)
 - ★ Lee, Williams '03 (tightening HLW via ILP and matching)
 - ★ Anstreicher, Lee '04 (generalization of HLW)
 - ★ Burer, Lee '07 (another approach to computing the AL bound)
 - ▶ NLP relaxation
 - ★ Anstreicher, Fampa, Lee & Williams '96 (continuous NLP relaxation and parallel B&B)

Complementary Bounds (Anstreicher, Fampa, Lee, Williams)

$$\ln \det C[S, S] = \ln \det C + \ln \det C^{-1}[N \setminus S, N \setminus S]$$

- So a maximum entropy s -subset of N with respect to C is the **complement** of a maximum entropy $(n - s)$ -subset of N with respect to C^{-1}
- So a bound on the complementary problem plus the entropy of the entire system is a bound on the original problem
- These complementary bounds can be quite effective

NLP Bound (AFLW)

$$\max f(x) := \ln \det \left(\text{Diag}(x_j^{p_j}) C \text{Diag}(x_j^{p_j}) + \text{Diag}(d_j^{x_j} - d_j x_j^{p_j}) \right)$$

$$\text{subject to } \sum_{j \in N} a_{ij} x_j \leq b_i, \forall i; \quad \Leftarrow \text{CONSTRAINTS}$$

$$\sum_{j \in N} x_j = s;$$

$$0 \leq x_j \leq 1, \forall j,$$

where the constants $d_j > 0$ and $p_j \geq 1$ satisfy $d_j \leq \exp(p_j - \sqrt{p_j})$, and $\text{Diag}(d_j) - C[N, N] \succeq 0$.

NLP Bound, cont'd

$$\max f(x) := \ln \det \left(\text{Diag}(x_j^{p_j}) C \text{Diag}(x_j^{p_j}) + \text{Diag}(d_j^{x_j} - d_j x_j^{p_j}) \right)$$

For $(\overbrace{1, 1, \dots, 1}^S, \overbrace{0, 0, \dots, 0}^{N \setminus S})$

- $\text{Diag}(d_j^{x_j} - d_j x_j^{p_j}) = \text{Diag}(\overbrace{0, 0, \dots, 0}^S, \overbrace{1, 1, \dots, 1}^{N \setminus S})$.
- $\text{Diag}(x_j^{p_j}) C \text{Diag}(x_j^{p_j}) = \left(\begin{array}{c|c} C[S, S] & 0 \\ \hline 0 & 0 \end{array} \right)$

NLP Bound: Properties

- **Concavity:** Assume $D \succeq C$, $p_j \geq 1$, $0 < d_j \leq \exp(p_j - \sqrt{p_j})$. Then f is concave for $0 < x \leq e$
- **Dominance:** Assume that p and d satisfy the above, and $p' \geq p$. Let f' be defined as above, but using p' for p . Then $f'(x) \geq f(x) \forall 0 < x \leq e$
- **Scaling C by γ adds $s \ln(\gamma)$ to the obj.** Let
$$f_\gamma(x) := \ln \det \left(\gamma X^{p/2} (C - D) X^{p/2} + (\gamma D)^x \right) - s \ln(\gamma)$$
 - ▶ **Scaling:** Assume $I \succeq D \succeq C$, $p = e$. Then $f_\gamma(x) \geq f(x) \forall 0 \leq x \leq e$, $e^T x = s$ and $0 < \gamma \leq 1$
 - ▶ Assume $D \succeq C$, $D \succeq I$. Then $f_\gamma(x) \geq f(x) \forall 0 < x \leq e$, $e^T x = s$ and $\gamma \geq 1$, where p is chosen as above

These results give us some **guidance for choosing the p_j , d_j and γ**

Spectral Bound (KLQ)

$$z \leq \sum_{l=1}^s \ln \lambda_l(C)$$

- Determinant = product of eigenvalues.
- Eigenvalue interlacing.

$$\left(\begin{array}{c} \\ \\ \\ \boxed{} \\ \\ \end{array} \right) \quad \begin{array}{l} \lambda_1 \geq \lambda'_1 \\ \lambda_2 \geq \lambda'_2 \\ \lambda_3 \geq \lambda'_3 \\ \vdots \\ \lambda_s \geq \lambda'_s \end{array}$$

Lagrangian Spectral Bound (Lee)

For handling linear side constraints

$$\min_{\pi \in \mathbb{R}_+^m} v(\pi)$$

where

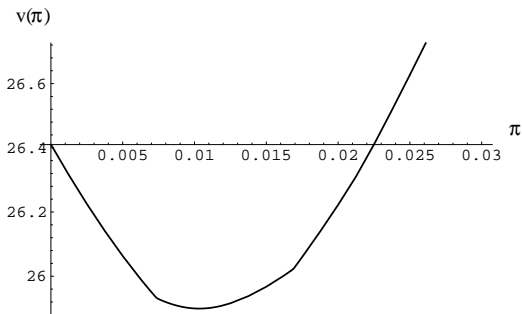
$$v(\pi) := \left\{ \sum_{l=1}^s \ln \lambda_l(D^\pi C D^\pi) + \sum_{i \in M} \pi_i b_i \right\},$$

and D^π is the diagonal matrix having

$$D_{jj}^\pi := \exp \left\{ -\frac{1}{2} \sum_{i \in M} \pi_i a_{ij} \right\}$$

Optimizing the Lagrangian Spectral Bound

- v_π is convex (in π)
- v_π is analytic when $\lambda_s(D^\pi C D^\pi) > \lambda_{s+1}(D^\pi C D^\pi)$



Optimizing the Bound, cont'd

- Let x^l be the eigenvector (of unit Euclidean norm) associated with λ_l .
- Define the continuous solution $\tilde{x} \in \mathbb{R}^N$ by $\tilde{x}_j := \sum_{l=1}^s (x_j^l)^2$, for $j \in N$.
- Define $\gamma \in \mathbb{R}^M$ by $\gamma_i := b_i - \sum_{j \in N} a_{ij} \tilde{x}_j$.
- If $\lambda_s > \lambda_{s+1}$, then γ is the gradient of f at π .
- Can incorporate this in a Quasi-Newton (or, with an expression for the Hessian, a Newton) method for finding the minimum. (Implemented using LBFGS-B (Zhu, Byrd, Nocedal) and a coarse line search)

Spectral Partition Bound (Hoffman, Lee, Willaims)

Let $\mathcal{N} = \{N_1, N_2, \dots, N_n\}$ denote a partition of N . Let $C' = 0$ except for $C'[N_k, N_k] = C[N_k, N_k]$.

$$z \leq \sum_{l=1}^s \ln \lambda_l(C')$$

- Based on “Fischer’s Inequality”
- For $\mathcal{N} = \{\{1\}, \{2\}, \dots, \{n\}\}$ we have “the diagonal bound”
- For $\mathcal{N} = \{N, \emptyset, \emptyset, \dots, \emptyset\}$ we have the ordinary spectral bound
- As we partition N , the optimal value with respect to C' cannot decrease, but the bound can decrease

ILP Bound (Lee, Williams)

Observation: Why calculate eigenvalue based bounds for small blocks of a partition? Just solve the small blocks exactly.

$x_k(i) = 1 \iff$ pick k elements from block N_i

$$\begin{aligned} g_s(\mathcal{N}) := & \max \sum_{i=1}^p \sum_{k=1}^{|N_i|} f_k(N_i) x_k(i) \\ \text{s.t.} & \sum_{k=1}^{|N_i|} x_k(i) \leq 1, \text{ for } i = 1, 2, \dots, p; \\ & \sum_{i=1}^p \sum_{k=1}^{|N_i|} k x_k(i) = s \\ & x_k(i) \in \{0, 1\}, \text{ for } i = 1, 2, \dots, p, \\ & \quad k = 1, 2, \dots, |N_i|. \end{aligned}$$

ILP Bound, cont'd

- Refines the spectral partition bound.
- Calculate via dynamic programming (assuming $|N_i|$ is bounded):

Boundary conditions:

$$v_t(j) := -\infty \text{ when } \sum_{i=1}^j |N_i| < t \leq s;$$
$$v_0(0) := 0.$$

$$v_t(j) = \max_{0 \leq k \leq \min\{|N_j|, t\}} \{f_k(N_j) + v_{t-k}(j-1)\}.$$

Then $v_s(p) = g_s(\mathcal{N})$

- Can even calculate via Edmonds' min-weight matching algorithm when $|N_i| \leq 2$.

Masked Spectral Bound (Anstreicher, Lee)

A mask is a (symmetric) $X \succeq 0$ having $\text{diag}(X) = e$. The associated masked spectral bound is

$$\xi_{C,s}(X) := \sum_{l=1}^s \ln(\lambda_l(C \circ X))$$

Special combinatorial cases:

- Spectral bound $X := E$
- Diagonal bound $X := I$
- Spectral partition bound $X := \text{Diag}_i(E_i)$

Validity

Based on

- $\det A = \prod_l \lambda_l(A)$
- “Oppenheim’s Inequality”

$$\det A \leq \det A \circ B / \prod_{j=1}^n B_{jj} ,$$

where $A \succeq 0$ and $B \succeq 0$

- the eigenvalue inequalities $\lambda_l(A) \geq \lambda_l(A')$, where $A \succeq 0$, and A' is a principal submatrix of A

Some References

- Lee. **Maximum entropy sampling**. In A.H. El-Shaarawi and W.W. Piegorsch, eds., “**Encyclopedia of Environmetrics**”. Wiley, 2001. Second Edition, pp.1570–1574, 2012.
- Lee. **Semidefinite programming in experimental design**. In H. Wolkowicz, R. Saigal and L. Vandenberghe, eds., “**Handbook of Semidefinite Programming**”, International Ser. in Oper. Res. and Manag. Sci., Vol. 27, Kluwer, 2000.
- Lee. **Techniques for Submodular Maximization**. Fields Institute Communications Series, “**Discrete Geometry and Optimization**”, (Edited by Karoly Bezdek, Yinyu Ye, and Antoine Deza). Springer, 163–178, 2013.