Rank Minimization via Online Learning

Inderjit S. Dhillon
University of Texas at Austin

Workshop on Algorithms for Modern Massive Data Sets(MMDS)
Stanford University, CA
June 28, 2008

Joint work with Raghu Meka, Prateek Jain, Constantine Caramanis

Overview

- Rank Minimization Problem (RMP)
- Algorithm based on Multiplicative Weights (MW) Update
- Algorithm based on Online Convex Programming (OCP)
- Results
- Conclusions

Rank Minimization Problem(RMP)

$$\begin{aligned} \text{RMP}: \min \quad & \operatorname{rank}(X) \\ \text{s.t} \quad & \operatorname{tr}(A_i X) \geq b_i, \quad 1 \leq i \leq m \\ & X \in \mathcal{C}. \end{aligned}$$

Rank Minimization Problem(RMP)

$$\begin{aligned} \text{RMP}: \min \quad & \operatorname{rank}(X) \\ \text{s.t} \quad & \operatorname{tr}(A_i X) \geq b_i, \quad 1 \leq i \leq m \\ & X \in \mathcal{C}. \end{aligned}$$

- $\mathcal{C}$ is an "easy" convex set, e.g.
  - Unit ball under any L_p or Frobenius norm
  - The semi-definite cone
  - The intersection of the unit ball with the semi-definite cone
- Applications to various Machine Learning Problems
  - Low-rank Kernel Learning
  - Non-negative Matrix Factorization
  - Nonlinear Dimensionality Reduction
- NP-hard to optimize

Existing Work

- $X = UV^T$, U is $m \times r$, V is $n \times r$

Existing Work

- $X = UV^T$, U is $m \times r$, V is $n \times r$
- Drop the rank constraint
  - Take top eigenvectors of the solution
  - Random projection of the solution [Barvinok'02]

Existing Work

- $X = UV^T$, U is $m \times r$, V is $n \times r$
- Drop the rank constraint
  - Take top eigenvectors of the solution
  - Random projection of the solution [Barvinok'02]
- Relax rank constraint to a convex constraint
  - Trace-norm [Fazel, Hindi, Boyd'01] $\sum_i \sigma_i$
  - Log-det[Fazel, Hindi, Boyd'03] $\log\det(X + \delta I)$
- Typically, no guarantees on the quality of solution
- Trace-norm guarantees optimal solution when RMP restricted to equality constraints ($\text{tr}(A_i X) = b_i$) and A_i's are well-conditioned [Recht, Fazel, Parrilo'07]

Existing Work

- $X = UV^T$, U is $m \times r$, V is $n \times r$
- Drop the rank constraint
  - Take top eigenvectors of the solution
  - Random projection of the solution [Barvinok'02]
- Relax rank constraint to a convex constraint
  - Trace-norm [Fazel, Hindi, Boyd'01] $\sum_i \sigma_i$
  - Log-det[Fazel, Hindi, Boyd'03] $\log\det(X + \delta I)$
- Typically, no guarantees on the quality of solution
- Trace-norm guarantees optimal solution when RMP restricted to equality constraints ($\text{tr}(A_i X) = b_i$) and A_i's are well-conditioned [Recht, Fazel, Parrilo'07]
- Our methods:
  - Scalable
  - Work for larger class of problems
  - Have approximation guarantees

Special Case: one constraint

- RMP with only one trace constraint

$$\begin{aligned} \text{RMP1}: \min \quad & \text{rank}(X) \\ \text{s.t.} \quad & \text{tr}(AX) \geq b \\ & X \in \mathcal{C} \end{aligned}$$

- Easy to solve for various interesting $\mathcal{C}$, e.g.

$$\begin{aligned} \text{RMP1}: \min \quad & \text{rank}(X) \\ \text{s.t.} \quad & \text{tr}(AX) \geq b \\ & X \succeq 0, \|X\|_F \leq 1 \end{aligned}$$

- Let $A = U\Lambda U^T$, $\Sigma = \Lambda_+$
- Minimum rank solution: minimum k s.t. $\sqrt{\sum_{i=1}^{k} \sigma_i^2} \geq b$

$$X = \frac{U_k \Sigma_k U_k^T}{\sqrt{\sum_{i=1}^{k} \sigma_i^2}}$$

First Method

Generalized Experts Framework

- Given: n experts, set of events $\mathcal{E}$, and penalty matrix M

$$M(i,j) = \text{penalty for } i\text{-th expert if } j\text{-th event occurs}$$

- Adversary chooses event j^t at step t
- Task: Choose a distribution on experts that minimizes expected penalty
- Penalty measured relative to the best expert

$$\text{Regret(T)} = \sum_{t=1}^{T} \sum_{\ell=1}^{n} p_\ell^t M(\ell, j^t) - \min_i \sum_{t=1}^{T} M(i, j^t)$$

Multiplicative Weights Update Algorithm

- Assign a weight w_i^t to each expert i at t-th step
- Define distribution $\mathcal{D}^t = (p_1^t, \ldots, p_n^t)$, where

$$p_i^t = \frac{w_i^t}{\sum_j w_j^t}$$

- Decrease weight of experts which incur large penalties

$$w_i^{t+1} = \begin{cases} w_i^t (1-\delta)^{M(i,j^t)/\rho} \text{ if } M(i,j^t) \geq 0 \\ w_i^t (1+\delta)^{-M(i,j^t)/\rho} \text{ if } M(i,j^t) < 0 \end{cases}$$

- $M(i,j) \in [-\rho, \rho]$, and δ is a parameter
- Let $\epsilon > 0$ be an error parameter, $\delta = \min\{\frac{\epsilon}{4\rho}, \frac{1}{2}\}$, and $T = \frac{16\rho^2 \log n}{\epsilon^2}$

$$\text{Avge Regret(T)} = \frac{\sum_{t=1}^{T} \sum_{\ell=1}^{n} p_\ell^t M(\ell, j^t)}{T} - \frac{\sum_t M(k, j^t)}{T} \leq \epsilon, \quad \forall k$$

Use of MW Update Algorithm in Optimization

- Generalized Experts Framework can be for optimization:
  - Linear Programming, packing problems [PST'91]
  - SDP [Arora, Hazan, Kale'05 and Arora, Kale'07]
  - Boosting [Freund and Schapire'95]
- Basic idea: treat each constraint as an expert
- Solve optimization problem for weighted combination of constraints
- Increase weight of most violated constraint
- Regret bound of the MW update algorithm used to prove feasibility

RMP: MW Update Algorithm

- Let each constraint be an expert

$$M(i, X) = \text{tr}(A_i X) - b_i$$

- Define Oracle $\mathcal{O}(A, b)$:

$$\begin{aligned} \text{RMP1} : \min \quad & \text{rank}(X) \\ \text{s.t.} \quad & \text{tr}(AX) \geq b \\ & X \in \mathcal{C} \end{aligned}$$

- At each step, Oracle is provided $A^t = \sum_i p_i A_i$ and $b^t = \sum_i p_i b_i$
- Oracle acts as an adversary which provides events X^t
- Use multiplicative updates for updating weight (p_i) of each constraint
- Efficient implementation of Oracle for various interesting $\mathcal{C}$

RMP: MW Update Analysis

- Case 1: Oracle returns infeasible $\Longrightarrow$ problem is infeasible
- Case 2: If Oracle returns feasible solution at each step: $\bar{X} = \frac{\sum_t X_t}{T}$
- Using regret bound for Multiplicative Weight Update:

$$\frac{\sum_{t=1}^{T} \sum_{l=1}^{n} p_\ell^t (\text{tr}(A_\ell X^t) - b_\ell)}{T} \leq \epsilon + \frac{\sum_t (\text{tr}(A_k X^t) - b_k)}{T}, \ \forall k$$

- LHS≥ 0, as oracle returns feasible solution

$$0 \leq \epsilon + (\text{tr}(A_k \bar{X}) - b_k), \ \forall k$$

- Each constraint satisfied within an error of ϵ

RMP: MW Update Analysis Contd...

Main Result 1:

$$\text{rank}(\bar{X}) \leq \sum \text{rank}(X_t) \leq k^* T = O\left(\frac{\Delta^2 D^2 \log nk^*}{\epsilon^2}\right),$$

$$\text{Running time: } O\left(\frac{\Delta^2 D^2 \log n}{\epsilon^2}(T_{\mathcal{O}} + mn^2)\right),$$

$$\text{where} \quad \Delta = \max\{\|A_i\|_F + |b_i| : 1 \leq i \leq m\}$$

$$D = \max\{\|X\|_F : X \in \mathcal{C}\}$$

$$T_{\mathcal{O}} = \text{ Oracle's running time}$$

Second Method

Online Convex Programming (OCP)

- Models useful online learning problems, e.g. network routing
- Offline Convex Programming vs Online Convex Programming:

$$\min_{z} \quad f(z) \qquad\qquad \min_{z_1, z_2, \ldots, z_T} \quad \sum_{t=1}^{T} f_t(z_t)$$

$$\text{s.t.} \quad z \in K \qquad\qquad \text{s.t.} \quad z_t \in K, \forall t$$

- For finding z_t, only $f_1, f_2, \ldots, f_{t-1}$ known

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(z_t) - \min_{z \in K} \sum_{t=1}^{T} f_t(z).$$

- [Zinkevich'03] introduced Projected Gradient Procedure, and showed

$$\text{Regret}(T) \leq G \cdot \|K\| \sqrt{T},$$

$\|K\| = \max_{z_1, z_2 \in K} \|z_1 - z_2\|, \quad G = \max_{z \in K, t \in \{1, \ldots\}} \| \bigtriangledown f_t(z)\|$

Rank Minimization: Online Convex Programming

- Use OCP for rank minimization
- Set convex set K as space of convex combinations of constraints

$$K = \left\{ \sum_i \lambda_i (A_i, b_i) : \sum_i \lambda_i = 1, \lambda_i \geq 0 \ \forall i \right\}$$

- Associate cost functions f_X with feasible points of RMP:

$$f_X(A, b) = \text{tr}(AX) - b$$

- Algorithm:
  - Obtain X^t using Oracle $\mathcal{O}(A^t, b^t)$
  - Define function $f^t(A, b) = \text{tr}(AX^t) - b$
  - Set (A^{t+1}, b^{t+1})=Projected Gradient$((A^t, b^t), f^t(A, b), K, t)$

RMP: OCP Analysis

Using regret bound:

$$\frac{\sum_{t=1}^{T}(\text{tr}(A^t X^t) - b^t)}{T} \leq \frac{\min_{(A,b)\in K}\sum_{t=1}^{T}(\text{tr}(AX^t) - b)}{T} + \frac{\Delta D}{\sqrt{T}},$$

where, $\Delta = \max\{\|A_i\|_F + |b_i| : 1 \leq i \leq m\}$,
$D = \| \bigtriangledown f^t(z)\| = \max\{\|X\|_F : X \in \mathcal{C}\}$

- If oracle returns a feasible X^t, $\forall t$: LHS ≥ 0
- For $T = \Delta^2 D^2/\epsilon^2$ and $\overline{X} = \sum_t X^t/T$,

$$\text{tr}(A\overline{X}) \geq b - \epsilon, \text{for all } (A, b) \in K$$

RMP: OCP Analysis Contd...

Main Result 2:

$$\boxed{\begin{array}{l}\text{rank}(X) \leq \sum \text{rank}(X_t) \leq k^* T = \frac{4\Delta^2 D^2 k^*}{\epsilon^2}, \\ \text{Running time: } O(\frac{\Delta^2 D^2}{\epsilon^2}(T_{OCP} + T_{\mathcal{O}} + mn^2))\end{array}}$$

$$\begin{aligned}\text{where} \quad \Delta &= \max\{\|A_i\|_F + |b_i| : 1 \leq i \leq m\} \\ D &= \max\{\|X\|_F : X \in \mathcal{C}\} \\ T_{\mathcal{O}} &= \text{Oracle's running time} \\ T_{OCP} &= \text{Projected Gradient's running time}\end{aligned}$$

- Tigher approximation guarantees than MW Update Algorithm
- Slower than MW Update Algorithm

Experimental Results

Results: Synthetic Datasets

$$\begin{aligned} \text{RMP}: \min \quad & \text{rank}(X) \\ \text{s.t} \quad & \text{tr}(A_i X) \geq b_i \quad 1 \leq i \leq m \\ & X \succeq 0, \|X\|_F \leq 1. \end{aligned}$$

- $A_i \in S^n$ generated randomly, $\epsilon = 0.05$
- $\Delta = \max_i\{\|A_i\|_F\}, D = 1$

Method\n	50	75	100	200	300
RMP-MW	23.25	11.25	7.3	2	2
RMP-OCP	12.8	7.5	5.3	2	2
Trace-norm	6.8	6.7	6.5	-	-
LogDet	5	4.2	4.0	-	-

Table: Rank of solution

Application: Low-rank Kernel Learning

$$
\begin{aligned}
\min_K \quad & \|K - K_0\|_F \\
\text{s.t.} \quad & \mathrm{tr}(S_i K) \leq \ell \quad \forall\, i \in \mathcal{S} \\
& \mathrm{tr}(D_j K) \geq u \quad \forall\, j \in \mathcal{D} \\
& K \succeq 0 \\
& \mathrm{rank}(K) \leq r,
\end{aligned}
$$

Application: Low-rank Kernel Learning

$$\begin{array}{ll} \min_K & \|K - K_0\|_F \\ \text{s.t.} & \text{tr}(S_i K) \leq \ell \quad \forall\, i \in \mathcal{S} \\ & \text{tr}(D_j K) \geq u \quad \forall\, j \in \mathcal{D} \\ & K \succeq 0 \\ & \text{rank}(K) \leq r, \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \min_K & \text{rank}(K) \\ \text{s.t.} & \text{tr}(S_i K) \leq \ell \quad \forall i \\ & \text{tr}(D_j K) \geq u \quad \forall j \\ & \text{tr}(K K_0) \geq \beta \\ & \|K\|_F \leq 1, \quad K \succeq 0 \end{array}$$

- $D = 1, \Delta = O\left(\ell + u + \beta + \|K_0\|_F\right)$
- Oracle:

$$\min_K \ \text{rank}(K) : \ \text{tr}(AK) \geq b, \|K\|_F \leq 1, \ K \succeq 0$$

  - Implementation: $A = U \Lambda U^T$, $\Sigma = |\Lambda|_+$
  - Solution: minimum k s.t. $\sqrt{\sum_{i=1}^{k} \sigma_i^2} \geq b$

Results: UCI Datasets

Dataset\Method	GK	MW	OCP	BK
Musk	80.80	93.11	**98.15**	81.51
	(476)	(44.1)	(61.2)	(61.2)
Heart	77.44	91.05	**91.13**	83.91
	(267)	(46.8)	(39.5)	(39.5)
Ionosphere	90.34	**91.26**	91.17	90.67
	(350)	(40)	(27.9)	(27.9)
Cancer	90.12	93.14	91.46	**93.38**
	(569)	(82)	(94)	(94)
Scale	66.34	**73.78**	72.46	72.11
	(607)	(146)	(91)	(91)

- GK: Baseline Gaussian kernel
  BK: Burg Kernel Method [Kulis, Sustik, Dhillon'06]
  MW: Multiplicative Weights update method
  OCP: Online Convex Programming method

Conclusions

- Considered a general Rank Minimization Problem
- Provide hardness result for approximation
- Provide efficient algorithms using two online learning paradigms
  - Multiplicative Weights Update Algorithm
  - Online Convex Programming
- Provide approximation guarantees for both of our algorithms
- Application to the problem of Kernel Learning
- Future Work
  - Analysis of both MW and OCP algorithms from optimization perspective
  - Obtain better approximation factor for RMP or prove matching lower bound
  - Application to other Machine Learning problems like Semi-definite Embedding

RMP: Computational Complexity

- NP-hard problem to optimize
- Even hard to approximate within a logarithmic factor
- Assuming $\mathrm{NP} \not\subseteq \mathrm{DTIME}(n^{poly \log n})$, RMP not approximable within a factor of

$$2^{\log^{1-\delta} \Delta} \text{ for every } \delta > 0$$

$$\Delta = \max\{\|A_i\|_F + |b_i| : 1 \leq i \leq m\}$$

- Proof technique similar to one used by [Almadi, Kann'98]

Special cases of RMP

- NNMA:

$$\begin{aligned} \min_X \quad & \|X - X_0\|_F \\ \text{s.t.} \quad & X_{ij} \geq 0 \quad \forall\ i, j \\ & \text{rank}(X) \leq r \end{aligned}$$

Special cases of RMP

- NNMA:

$$\begin{array}{rl} \min_X & \|X - X_0\|_F \\ \text{s.t.} & X_{ij} \geq 0 \quad \forall\ i,j \\ & \text{rank}(X) \leq r \end{array} \quad \Leftrightarrow \quad \begin{array}{rl} \min_X & \text{rank}(X) \\ \text{s.t.} & \text{tr}(X\mathbf{e_j}\mathbf{e_i}^T) \geq 0 \quad \forall i,j \\ & \text{tr}(XX_0) \geq \beta,\ \|X\|_F \leq \|X_0\|_F \end{array}$$

Special cases of RMP

- NNMA:

$$\begin{array}{ll} \min_X & \|X - X_0\|_F \\ \text{s.t.} & X_{ij} \geq 0 \quad \forall\ i,j \\ & \text{rank}(X) \leq r \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \min_X & \text{rank}(X) \\ \text{s.t.} & \text{tr}(X\mathbf{e_j}\mathbf{e_i}^T) \geq 0 \quad \forall i,j \\ & \text{tr}(XX_0) \geq \beta,\ \|X\|_F \leq \|X_0\|_F \end{array}$$

- Sparse PCA:

$$\begin{array}{ll} \min_A & \|A - A_0\|_F \\ \text{s.t.} & \text{tr}(|A|\mathbf{1}\mathbf{1}^T) \leq k \\ & A \succeq 0, \\ & \text{rank}(A) \leq r \end{array}$$

Special cases of RMP

- NNMA:

$$\begin{array}{ll} \min_X & \|X - X_0\|_F \\ \text{s.t.} & X_{ij} \geq 0 \quad \forall\ i,j \\ & \text{rank}(X) \leq r \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \min_X & \text{rank}(X) \\ \text{s.t.} & \text{tr}(X\mathbf{e_j}\mathbf{e_i}^T) \geq 0 \quad \forall i,j \\ & \text{tr}(XX_0) \geq \beta,\ \|X\|_F \leq \|X_0\|_F \end{array}$$

- Sparse PCA:

$$\begin{array}{ll} \min_A & \|A - A_0\|_F \\ \text{s.t.} & \text{tr}(|A|\mathbf{1}\mathbf{1}^T) \leq k \\ & A \succeq 0, \\ & \text{rank}(A) \leq r \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \min_A & \text{rank}(A) \\ \text{s.t.} & \text{tr}(|A|\mathbf{1}\mathbf{1}^T) \leq k \\ & A \succeq 0, \\ & \text{tr}(AA_0) \geq \beta,\ \|A\|_F \leq \|A_0\|_F \end{array}$$

Special Cases of RMP Contd...

- Semi-definite Embedding:

$$\begin{aligned} \min_K \quad & -\operatorname{tr}(K) \\ \text{s.t.} \quad & \operatorname{tr}(K\mathbf{e}_{ij}\mathbf{e}_{ij}^T) = \operatorname{tr}(G\mathbf{e}_{ij}\mathbf{e}_{ij}^T) \\ & \operatorname{tr}(K\mathbf{1}\mathbf{1}^T) = 0 \\ & K \succeq 0 \\ & \operatorname{rank}(K) \leq r, \end{aligned}$$

where, $\mathbf{e}ij = \mathbf{e}_i - \mathbf{e}_j$

Special Cases of RMP Contd...

- Semi-definite Embedding:

$$\begin{array}{ll} \min_K & -\operatorname{tr}(K) \\ \text{s.t.} & \operatorname{tr}(K\mathbf{e}_{ij}\mathbf{e}_{ij}^T) = \operatorname{tr}(G\mathbf{e}_{ij}\mathbf{e}_{ij}^T) \\ & \operatorname{tr}(K\mathbf{1}\mathbf{1}^T) = 0 \\ & K \succeq 0 \\ & \operatorname{rank}(K) \leq r, \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \min_K & \operatorname{rank}(K) \\ \text{s.t.} & \operatorname{tr}(K\mathbf{e}_{ij}\mathbf{e}_{ij}^T) = \operatorname{tr}(G\mathbf{e}_{ij}\mathbf{e}_{ij}^T) \\ & \operatorname{tr}(K\mathbf{1}\mathbf{1}^T) = 0 \\ & K \succeq 0 \\ & \operatorname{tr}(K) \leq \beta \end{array}$$

where, $\mathbf{e}ij = \mathbf{e}_i - \mathbf{e}_j$