Optimization in Big Data

Ethan Xingyuan Fang

University Park PA, June, 2018
Outline

1 Linear Programming
2 Convex Optimization
3 Stochastic First-Order Methods
4 ADMM
What is Linear Programming?

Linear Optimization Problems
Linear Programming studies linear optimization problems of the following forms

$$\max_{x} c^T x, \text{ subject to } Ax \leq b, \ x \geq 0,$$

or

$$\max_{x} c^T x, \text{ subject to } Ax = b.$$ 

These two forms are actually equivalent. Why?
Some Geometry of Linear Programming

Polyhedron
The feasible set of linear programming problems $Ax \leq b$ is called a polyhedron. In 2D or 3D spaces, it can be visualized as following.

Efficient algorithms are developed based on this observation: Simplex method.
Extreme Points and Vertices
In mathematics, an extreme point of a convex set $S$ in a real vector space is a point in $S$ which does not lie in any open line segment joining two points of $S$. Intuitively, an extreme point is a “vertex” of $S$. 
Extreme Points and Vertices
In mathematics, an extreme point of a convex set $S$ in a real vector space is a point in $S$ which does not lie in any open line segment joining two points of $S$. Intuitively, an extreme point is a “vertex” of $S$.

Extreme Points in LP
For any LP problems with bounded optimal value, there is an extreme point of the feasible set achieves the optimal value.
Simplex Method

Consider LP problems of the following form

$$\max_x c^T x, \text{ subject to } Ax \leq b, x \geq 0,$$

where $A \in \mathbb{R}^{m \times n}$. Based on the observation of extreme points, simplex method is proposed by selecting basic variables, i.e., selecting the active constraints.
A different formulation
For the problem of our interest

\[
\max_{x} c^T x, \text{ subject to } Ax \leq b, x \geq 0.
\]

A equivalent formulation is

\[
\max_{x} c^T x, \text{ subject to } Ax + w \leq b, x, w \geq 0.
\]
Interior Point Method

A different formulation
For the problem of our interest

\[
\max_{x} c^T x, \text{ subject to } Ax \leq b, x \geq 0.
\]

A equivalent formulation is

\[
\max_{x} c^T x, \text{ subject to } Ax + w \leq b, x, w \geq 0.
\]

What if we consider a different formulation?

\[
\max_{x} c^T x + \mu \left( \sum_{i} \log x_i + \sum_{j} w_j \right), \text{ subject to } Ax + w = b, x, w \geq 0.
\]
Interior Point Method

(a) $\mu = \infty$
(b) $\mu = 1$
(c) $\mu = 0.01$
(d) central path
Brief History of Linear Programming

Simplex Method 1946
Ellipsoid Method 1980
Interior-Point Method 1986
Algorithm keeps improving...

Military Solver 1946
CPLEX 1.0 1988
Gurobi 1.0 2008
CPLEX/Gurobi 12.6/6.5 2016

Implementation keeps improving...

Computer keeps improving...
Solving LP Problems in Matlab

In Matlab, solving linear programming can be done using “linprog” that linprog(c,A,b) solves the problem

$$\min_x c^T x \text{ subject to } Ax \leq b.$$ 

There are many options could help you solve the problem more efficiently, such as adding equality ($A_2 x = b_2$) or nonnegative constraints ($x \geq 0$). Type “help linprog” in Matlab to see how.

There are more efficient commercial solvers such as CPLEX by IBM, Gurobi and MOSEK (Free to students!).
LAD Regression

In linear regression, we usually take the least square loss. In particular, suppose $y_i = x_i^T \beta^* + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$. We estimate $\beta^*$ by

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2.$$ 

What if we take a different loss function called least absolute deviation (LAD) that

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} |y_i - x_i^T \beta|.$$ 

This problem can be formulated as a linear programming problem.
Penalized LAD

In high-dimensional statistics, when sample size is much less than the number of features, we usually assume sparsity of $\beta$, i.e., we assume most of the components of $\beta$ are zeros. To encourage sparsity of the estimator, we add an $\ell_1$-norm penalization that

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} |y_i - x_i^T \beta| + \lambda \|\beta\|_1.$$ 

Again, this problem can be formulated as an LP problem.
Generalizing LAD-Quantile Regression

Sample median of samples \( \{x_1, \ldots, x_n\} \) is

\[
\hat{x} = \arg\min_a \sum_{i=1}^{n} |x_i - a|.
\]

So, the LAD regression essentially estimates the median of the linear model, which is robust to outliers. What if we care about the \( q \)-th quantile of the model? Let \( \rho_q(u) = u\{q - 1(u < 0)\} \) be the quantile loss function. \( q \)-th quantile of \( \{x_1, \ldots, x_n\} \) solves

\[
\min_a \sum_{i=1}^{n} \rho_q(x_i - a).
\]
Generalizing LAD-Quantile Regression

Quantile regression for the $q$-th quantile solves

$$
\hat{\beta}(q) = \arg\min_\beta \sum_{i=1}^n \rho_q(y_i - x_i^T \beta),
$$

where $\rho_q(u) = u \{ q - 1(u < 0) \}$. 
Dantzig Selector

Still in high-dimensional statistics, a well-known approach to estimate $\beta$ is by the Dantzig selector defined as follows.

$$\hat{\beta} = \arg\min_{\beta} \|\beta\|_1, \text{ subject to } \|X^T y - X^T X \beta\|_\infty \leq \lambda.$$  

This estimator can be again solved by LP formulation.
Local Warming

National Oceanic and Atmospheric Administration (NOAA) collects and archives weather data from thousands of collection sites around the world. 
Temperature Model

Assume daily average temperature has a sinusoidal annual variation superimposed on a linear trend:

$$\min_{\beta_0, \ldots, \beta_3} \sum_{d \in D} \left| a_0 + a_1 d + a_2 \cos\left(\frac{2\pi d}{364.25}\right) + a_3 \sin\left(\frac{2\pi d}{365.25}\right) - T_d \right|.$$ 

We see a positive signal! $\hat{a}_1 = 0.0200462$. Better model can be found here.

http://www.princeton.edu/~rvdb/ampl/nlmodels/LocalWarming/McGuireAFB/McGuire.html
Outline

1. Linear Programming
2. Convex Optimization
3. Stochastic First-Order Methods
4. ADMM
Convex Optimization

Convex Set
A set $S$ is convex if for any $x, y \in S$, then $\lambda x + (1 - \lambda)y \in S$ for any $\lambda \in [0, 1]$. 
Convex Function
A function $f(\cdot)$ is convex if for any $x, y \in \text{dom}(f)$,

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(\lambda x + (1 - \lambda) y)$$

for any $\lambda \in [0, 1]$. Note that a simple way to check if a differentiable function $f$ is convex or not is to check if its Hessian $\nabla^2 f(x)$ is always positive semidefinite.
The standard form of convex optimization problem is

\[ \min_x f(x), \]

subject to \( f_i(x) \leq 0, \]

\[ Ax = b. \]
Convex Optimization

Convex Optimization Problem
The standard form of convex optimization problem is

\[
\min_x f(x), \\
\text{subject to } f_i(x) \leq 0, \\
Ax = b.
\]

Example
However, it is sometimes case by case:

\[
\min_{x_1, x_2} x_1^2 + x_2^2, \text{ subject to } x_1/(1 + x_2^2) \leq 0, \ (x_1 + x_2)^2 = 0.
\]
Quadratic Program (QP)
QP problems are of the form

\[ \min_x x^T Px + q^T x + r, \]

subject to \( Gx \leq h, \)
\[ Ax = b, \]

where the matrix \( P \) is positive definite. This is to minimize a convex quadratic function over a polyhedron.
Quadratic Program (QP)
QP problems are of the form

$$\min_x x^T P x + q^T x + r,$$

subject to $G x \leq h,$

$$A x = b,$$

where the matrix $P$ is positive definite. This is to minimize a convex quadratic function over a polyhedron.

Example
Least Squares Minimization:

$$\min_x \|A x - b\|_2^2$$

Analytical solution is $x^* = A^{-} b$, where $A^{-}$ denotes the Moore-Penrose pseudo-inverse. We can also add linear constraints like $\ell \leq x \leq u$. 
Example

The following is a risk-averse optimization problem

$$\min_x c^T x + \gamma x^T \Sigma x$$

subject to $Gx \leq h$, $Ax = b$. 

Convex Optimization
Convex Optimization

Quadratically Constrained Quadratic Program (QCQP)

\[
\min_{x} x^T P x + q^T x + r,
\]

subject to \( x^T P_i x + q_i^T x + r_i \leq 0 \), for \( i = 1, \ldots, n \)

\( Ax = b, \)

where \( P \) is positive semidefinite, and all \( P_i \)'s are positive definite.
Second-Order Conic Programming (SOCP)

SOCP is of the form

$$\min_{x} f^T x,$$

subject to $$\|A_i x + b_i\|_2 \leq c_i^T x + d_i$$, for $$i = 1, \ldots, n$$,

$$Fx = g,$$

This is a generalization of LP and QCQP.
Support Vector Machine (SVM) is one of the most popular classification methods. To compute the hyperplane, the problem can be formulated as a QP problem:

\[
\min_{w, \beta, \xi} \frac{1}{2} w^T w + C e^T \xi, \quad \text{subject to } Y X w + \beta y + \xi \geq e, \xi \geq 0.
\]
Distance Weighted Discrimination

Distance Weighted Discrimination (DWD) minimizes the sum of the reciprocals of the residuals, perturbed by a penalized vector $\xi$, where we have

$$\min_{r, w, \beta, \xi} \sum_i (1/r_i) + Ce^T \xi, \text{ subject to } r = YXw + \beta y + \xi, \ w^T w \leq 1, \ r \geq 0, \ \xi \geq 0.$$
Convex Optimization

Distance Weighted Discrimination

Distance Weighted Discrimination (DWD) minimizes the sum of the reciprocals of the residuals, perturbed by a penalized vector $\xi$, where we have

$$\min_{r, w, \beta, \xi} \sum_i \left(\frac{1}{r_i}\right) + C e^T \xi, \quad \text{subject to} \quad r = YXw + \beta y + \xi, \quad w^T w \leq 1, \quad r \geq 0, \quad \xi \geq 0.$$

SOCP Formulation

Introducing new variables, let $r_i = \rho_i - \sigma_i$, where $\rho_i = (r_i + 1/r_i)/2$, $\sigma_i = (1/r_i - r_i)/2$. Then, $\rho_i^2 - \sigma_i^2 = 1$, and $\rho_i + \sigma_i = 1/r_i$. Thus, we have the DWD problem can be written as

$$\min_{w, \beta, \xi, \rho, \sigma, \tau} C e^T \xi + e^T \rho + e^T \sigma$$

subject to $YXw + \beta y + \xi - \rho + \sigma = 0,$

$$\|w\|_2 \leq 1,$$

$$\tau = e, \quad \| (\sigma_i, \tau_i) \|_2 \leq \rho_i, \quad \text{for} \ i = 1, \ldots, n.$$
Outline

1. Linear Programming
2. Convex Optimization
3. Stochastic First-Order Methods
4. ADMM
Subgradient/subdifferential

For a differentiable convex function $f$, we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

for any $x, y \in \mathcal{D}$. 


First Order Methods

Subgradient/subdifferential

For a differentiable convex function $f$, we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

for any $x, y \in \mathcal{D}$.

For general function $f$, a subgradient of $f$ at $x$ is a vector $g$, such that

$$f(y) \geq f(x) + g^T (y - x)$$

for all $y$. 
Subgradient/subdifferential

For a differentiable convex function $f$, we have

\[ f(y) \geq f(x) + \nabla f(x)^T (y - x) \]

for any $x, y \in D$.

For general function $f$, a subgradient of $f$ at $x$ is a vector $g$, such that

\[ f(y) \geq f(x) + g^T (y - x) \]

for all $y$.

Set of all subgradients of $f$ at $x$ is called the subdifferential of $f$ at $x$, denoted as $\partial f$. 
Subgradient Methods

This is an extremely simple algorithm to minimize nondifferentiable convex function $f$

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}.$$
First Order Methods

Subgradient Methods
This is an extremely simple algorithm to minimize nondifferentiable convex function $f$

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}.$$  

Stepsize is very important. Some popular choices

- constant stepsize: $\alpha_k = \alpha$
- constant step length: $\alpha_k = \gamma / \|g^{(k)}\|_2$
- square summable but not summable: step sizes satisfy

$$\sum_{k=1}^{\infty} \alpha_k^2 = \infty, \quad \sum_{k=1}^{\infty} \alpha_k = \infty$$
Projected Subgradient Methods

For constrained optimization problem

\[
\min f(x), \quad \text{subject to } x \in C.
\]

\(f, C\) are convex.
Projected Subgradient Methods

For constrained optimization problem

\[
\min f(x), \text{ subject to } x \in C.
\]

\(f, C\) are convex.

Projected subgradient method is given by

\[
x^{(k+1)} = \Pi_C \left( x^{(k)} - \alpha_k g^{(k)} \right),
\]

where \(\Pi_C\) is the Euclidean projection onto \(C\), and \(g^{(k)} \in \partial f(x^{(k)})\).

Example

\[
\min \|x\|_1, \text{ subject to } Ax = b.
\]
Projected Subgradient for Dual

Primal:

\[ \min f(x), \text{ subject to } f_i(x) \leq 0. \]

Solve the dual:

\[ \max g(\lambda), \text{ subject to } \lambda \geq 0, \]

through

\[ \lambda^{(k+1)} = \left( \lambda^{(k)} - \alpha_k h^{(k)} \right)_+, \quad h^{(k)} \in \partial(-g)(\lambda^{(k)}). \]

Question: What is \( h^{(k)} \) here?
Example

\[ \min x^T Px - q^T x, \text{ subject to } x_i^2 \leq 1, \]

where \( P \succ 0 \).
First Order Methods

Primal-Dual Subgradient Method

Consider the general convex optimization problem

\[ \min f(x), \quad \text{subject to } f_i(x) \leq 0, \ Ax = b. \]

A simple idea to accelerate is to use both primal and dual information to update both primal and dual variables.
Equality Constrained Problem

Consider the problem

$$\min f(x), \text{ subject to } Ax = b,$$

with variable $x$ and optimal value $p^*$. We will work with the equivalent augmented problem

$$\min f(x) + (\rho/2)\|Ax - b\|_2^2, \text{ subject to } Ax = b,$$

with $\rho > 0$. 


Augmented Lagrangian

Augmented Lagrangian is

$$\mathcal{L}(x, \nu) = f(x) + \nu^T (Ax - b) + \frac{\rho}{2} \|Ax - b\|^2$$

$(x, \nu)$ is primal-dual optimal if and only if

$$0 \in \partial_x \mathcal{L}(x, \nu) = \partial f(x) + A^T \nu + \rho A^T (Ax - b),$$

and

$$0 = -\nabla_{\nu} \mathcal{L}(x, \nu) = b - Ax.$$
Primal-Dual Methods

Let \( T(x, \nu) = (\partial_x L(x, \nu), -\nabla_\nu L(x, \nu))^T \), and let \( z = (x, \nu)^T \). We update \( z \) by

\[
    z^{(k+1)} = z^{(k)} - \alpha_k T^{(k)},
\]

where \( T^{(k)} \in T(z^{(k)}) \), and \( \alpha_k \) is step length.

Question: How to generalize to inequality constrained problem?
Primal-Dual Methods

Let \( T(x, \nu) = (\partial_x \mathcal{L}(x, \nu), -\nabla_\nu \mathcal{L}(x, \nu))^T \), and let \( z = (x, \nu)^T \). We update \( z \) by

\[
z^{(k+1)} = z^{(k)} - \alpha_k T^{(k)},
\]

where \( T^{(k)} \in T(z^{(k)}) \), and \( \alpha_k \) is step length. More explicitly,

\[
x^{(k+1)} = x^{(k)} - \alpha_k (g^{(k)} + A^T \nu^{(k)} + \rho A^T (Ax^{(k)} - b)),
\]

and

\[
\nu^{(k+1)} = \nu^{(k)} + \alpha_k (Ax^{(k)} - b).
\]
First Order Methods

Primal-Dual Methods
Let \( T(x, \nu) = (\partial_x \mathcal{L}(x, \nu), -\nabla_\nu \mathcal{L}(x, \nu))^T \), and let \( z = (x, \nu)^T \). We update \( z \) by

\[
  z^{(k+1)} = z^{(k)} - \alpha_k T^{(k)},
\]

where \( T^{(k)} \in T(z^{(k)}) \), and \( \alpha_k \) is step length.

More explicitly,

\[
  x^{(k+1)} = x^{(k)} - \alpha_k \left( g^{(k)} + A^T \nu^{(k)} + \rho A^T (Ax^{(k)} - b) \right),
\]

and

\[
  \nu^{(k+1)} = \nu^{(k)} + \alpha_k (Ax^{(k)} - b).
\]

Question: How to generalize to inequality constrained problem?
First Order Methods

Inequality Constraints

$$\min f(x), \text{ subject to } f_i(x) \leq 0,$$

Augmented Lagrangian and Optimality

Augmented Lagrangian is

$$L(x, \lambda) = f_0(x) + \lambda^T F(x) + \left(\frac{\rho}{2}\right) \|F(x)\|^2_2,$$

$$(x, \lambda)$$ is primal-dual optimal if and only if

$$0 \in \partial x L(x, \lambda),$$

and

$$0 = -\nabla \lambda L(x, \lambda).$$
First Order Methods

Inequality Constraints

\[ \min f(x), \text{ subject to } f_i(x) \leq 0, \]

its equivalent augmented problem

\[ \min f(x) + (\rho/2)\|F(x)\|_2^2, \text{ subject to } F(x) \leq 0, \]

where \( F(x) = (f_1(x)_+, \ldots, f_m(x)_+)^T, \rho > 0. \)

Augmented Lagrangian and Optimality

Augmented Lagrangian is

\[ \mathcal{L}(x, \lambda) = f_0(x) + \lambda^T F(x) + (\rho/2)\|F(x)\|_2^2, \]

\((x, \lambda)\) is primal-dual optimal if and only if

\[ 0 \in \partial_x \mathcal{L}(x, \lambda), \text{ and } 0 = -\nabla_\lambda \mathcal{L}(x, \lambda). \]
Bregman Divergence

Let $h$ be a differentiable convex function, then its associated Bregman divergence is

$$D_h(y, x) = h(y) - h(x) - \nabla h(x)^T (y - x).$$

Mirror Subgradient

- get subgradient $g^{(k)} \in \partial f(x^{(k)})$
- update

$$x^{(k+1)} = \arg\min_{x \in C} \left\{ x^T g^{(k)} + \frac{1}{\alpha_k} D_h(x, x^{(k)}) \right\}$$
Bregman Divergence

Let $h$ be a differentiable convex function, then its associated Bregman divergence is

$$D_h(y, x) = h(y) - h(x) - \nabla h(x)^T (y - x).$$

Mirror Subgradient

- get subgradient $g^{(k)} \in \partial f(x^{(k)})$
- update

$$x^{(k+1)} = \arg\min_{x \in C} \left\{ x^T g^{(k)} + \frac{1}{\alpha_k} D_h(x, x^{(k)}) \right\}$$

This generalizes projected subgradient decent (why?).
First Order Methods

Example

Consider a constrained robust regression problem

$$\min f(x) = \|Ax - b\|_1 = \sum_{i=1}^{n} |a_i^T x - b_i|, \quad \text{subject to } e^T x = 1, \ x \geq 0.$$ 

Let’s take $h$ as the negative entropy

$$h(x) = \sum_{j=1}^{d} x_j \log x_j.$$
First Order Methods

Proximal Operator

Proximal operator of a function \( f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \) is

\[
\text{prox}_{\lambda, f}(v) = \arg\min_x \left\{ f(x) + \frac{1}{2\lambda} \|x - v\|_2^2 \right\},
\]

with parameter \( \lambda > 0 \).

The motivation of considering the proximal operator is that it may admit close-form simple solutions.

Note
This is a generalization of projection, why?
Example

Suppose $f = \| \cdot \|_1$. What is $\text{prox}_{\lambda, f}(v)$?
Connection to Fixed Points

The point $x^*$ minimizes $f$ if and only if $x^*$ is a fixed point that:

$$x^* = \text{prox}_{\lambda f}(x^*).$$

This provides a link between proximal operators and fixed point theory. Thus, many proximal algorithm can be viewed as methods for finding fixed points of appropriate operators.
First Order Methods

Infimal Convolution
The infimal convolution of functions $f$ and $g$, denoted as $f \circ g$, is defined as

$$(f \circ g)(v) = \inf_x \{ f(x) + g(v - x) \}.$$ 

Moreau-Yosida Regularization
Moreau-Yosida Regularization of $f$ is

$$M_{\lambda,f}(v) = \inf_x \{ f(x) + \frac{1}{2\lambda} \| x - v \|^2 \}.$$ 

This is a smoothed or regularized form of $f$:

- always has full domain
- always continuously differentiable
- has the same minimizers as $f$

Thus, we can minimize $M_{\lambda,f}$ instead of $f$, though $M_f$ could be hard to evaluate in practice.
Example

What is the Moreau-Yosida regularization of $|\cdot|$?
Example
What is the Moreau-Yosida regularization of $| \cdot |$?

Some More Interpretation
If $z = \text{prox}_{\lambda, f}(x)$, then

$$z = \arg\min_u \left\{ f(u) + \frac{1}{2\lambda} \| u - x \|^2_2 \right\}.$$  

This can be written as

$$0 \in \partial_z \left\{ f(z) + \frac{1}{2\lambda} \| z - x \|^2_2 \right\},$$

which holds if and only if

$$z \in (I + \lambda \partial f)^{-1}(x).$$
First Order Methods

Proximal Point Algorithm

\[ x^{k+1} = \text{prox}_{\lambda, f}(x^k). \]

This is the simplest proximal method, can be interpreted as

- gradient method applies to \( M_{\lambda, f} \)
- simple iteration for finding fixed point of \( \text{prox}_{\lambda, f} \)

If \( f(x) = (1/2)x^T P x - b^T x \), what is the algorithm doing?
**First Order Methods**

**Operator Splitting**
Suppose the objective is of the form

\[
\min_x f(x) + g(x).
\]

\(f\) is smooth and convex, and \(g\) is convex.

**Proximal Gradient Method**

\[
 x^{k+1} = \text{prox}_{\lambda,g}(x^k - \nabla f(x^{(k)})).
\]

This is a generalization of projected gradient, why?
Example: Lasso

Lasso problem is

$$\min_{\beta} \frac{1}{2} \| y - X \beta \|_2^2 + \lambda \| \beta \|_1.$$ 

What is the proximal gradient method?
Example: Matrix Completion

Given a matrix $M \in \mathbb{R}^{p \times q}$, and only observe entries $Y_{ij}, (i, j) \in \Omega$. Suppose we want to fill in missing entries. This is the matrix completion problem. Based on a nuclear norm relaxation, the problem is

$$
\min_{B \in \mathbb{R}^{p \times q}} \frac{1}{2} \sum_{(i,j) \in \Omega} (Y_{ij} - B_{ij})^2 + \lambda \|B\|_*.
$$

Here $\|B\|_*$ denotes the nuclear (or trace) norm of $B$, where

$$
\|B\|_* = \sum_{i=1}^{r} \sigma_i(B),
$$

where $r = \text{rank}(B)$ and $\sigma_1(B) \geq \sigma_2(B) \geq \cdots \geq \sigma_r(X) > 0$ are the singular values.
First Order Methods

Acceleration

We can accelerate proximal gradient descent to achieve a better complexity. Let \( x^0 = x^{-1} \in \mathbb{R}^n \), at the \( k \)-th iteration, we add an intermediate step before doing the proximal step:

\[
\begin{align*}
v &= x^{(k-1)} + \frac{k - 2}{k + 1} \left\{ x^{(k-1)} - x^{(k-2)} \right\}, \\
x^k &= \text{prox}_{\alpha,g} \left( v - \alpha \nabla f(v) \right).
\end{align*}
\]

Note that the first step when \( k = 1 \) is just usual proximal gradient update. Then, the auxiliary step \( v = x^{k-1} + \frac{k - 2}{k + 1} \left\{ x^{k-1} - x^{k-2} \right\} \) carries some “momentum” or “weight” from previous iterations.

Remark

Note that this also gives the accelerated gradient method, why?
Projection might cause trouble...

Consider the constrained optimization problem

$$\min_x f(x), \text{ subject to } x \in C,$$

where $f$ is convex and $C$ is convex. Suppose the projection step $P_C(x)$ is expensive. Projected gradient method might not be efficient.
Frank-Wolfe Method

Frank-Wolfe (Conditional Gradient) Method:

- Choose a feasible starting point \( x^{(0)} \in C \)
- At the \( k \)-th iteration, compute

\[
s^k \in \arg\min_{s \in C} \nabla f(x^k)^T s
\]

- Update the solution

\[
x^{k+1} = (1 - \gamma_k)x^k + \gamma_k s^k
\]
First Order Methods

Frank-Wolfe Method
Frank-Wolfe (Conditional Gradient) Method:

- Choose a feasible starting point \( x^{(0)} \in C \)
- At the \( k \)-th iteration, compute

\[
s^k \in \arg\min_{s \in C} \nabla f(x^k)^T s
\]

- Update the solution

\[
x^{k+1} = (1 - \gamma_k) x^k + \gamma_k s^k
\]

Note
There is no projection. Choosing \( \gamma_k \in [0, 1] \) guarantees the feasibility. You may view the update as

\[
x^{k+1} = x^k + \gamma_k (s^k - x_k)
\]

Standard choice of step size: \( \gamma_k = 2/(k + 1) \), i.e., we are moving less and less in the direction of the linearization minimizer as the algorithm proceeds.
Frank-Wolfe Method

When $C = \{x : \|x\| \leq t\}$ for an arbitrary norm. Then

$$s \in \arg\min_{\|s\| \leq t} \nabla f(x^k)^T s.$$ 

What is it?

Note

This can often be simpler or cheaper than projection onto $C = \{x : \|x\| \leq t\}$. 
First Order Methods

Examples

\[
\min_x f(x), \text{ subject to } \|x\|_1 \leq t,
\]
First Order Methods

Examples

\[
\begin{align*}
\min_x f(x), \ &\text{subject to } \|x\|_1 \leq t, \\
\min_x f(x), \ &\text{subject to } \|x\|_p \leq t, \ \text{for } p \in [1, \infty].
\end{align*}
\]
First Order Methods

Examples

\[ \min_{x} f(x), \text{ subject to } \|x\|_1 \leq t, \]

\[ \min_{x} f(x), \text{ subject to } \|x\|_p \leq t, \text{ for } p \in [1, \infty]. \]

\[ \min_{X} f(X), \text{ subject to } \|X\|_* \leq t, \]
Suboptimality Gap

There is a natural suboptimality gap that

\[ f(x^k) - f^* \geq \max_{s \in C} \nabla f(x^k)^T (x^k - s). \]

Sometimes, this is called a duality gap, why?
First Order Methods

Line Search
Instead of fixing $\gamma_k = 2/(k + 1)$, use exact line search for the step sizes

$$\gamma_k = \arg\min_{\gamma \in [0,1]} f(x^k + \gamma(s^k - x^k))$$

This can be done through golden-section method. Or, we can do backtracking here.
An Important Variant

At each iteration, we compute an away step that

\[ a^k = \arg\max_{a \in C} \nabla f(x^k)^T a, \]

Then, we choose \[ v = s^k - x^k \] OR \[ v = x^k - a^k, \] and let

\[ x^{k+1} = x^k + \gamma_k v. \]
Path Following

Consider the problem

$$\min_x f(x), \text{ subject to } \|x\| \leq t.$$ 

Frank-Wolfe algorithm can be used for path following, i.e., can produce an approximate solution path $\hat{x}(t)$ for all $t \geq 0$. Beginning at $t_0 = 0$, $x^*(0) = 0$, we fix parameters $\epsilon, m > 0$, then repeat for $k = 1, 2, \ldots$:

- Compute

$$t_k = t_{k-1} + \frac{(1 - 1/m)\epsilon}{\|\nabla f(\hat{x}(t_{k-1}))\|_*}$$

and set $\hat{x}(t) = \hat{x}(t_{k-1})$ for all $t \in [t_{k-1}, t_k)$.

- Compute $\hat{x}(t_k)$ by running Frank-Wolfe at $t = t_k$, terminating when the duality gap is $\leq \epsilon m$. 

First Order Methods

Theoretical Guarantee
With this strategy, we have

$$f(\hat{x}(t)) - f(x^*(t)) \leq \epsilon,$$

for all $t$. 
Outline

1. Linear Programming
2. Convex Optimization
3. Stochastic First-Order Methods
4. ADMM
Augmented Lagrangian Function

Lagrangian Dual Problem
Consider linearly constraint convex optimization problem.

\[
\min_x f(x), \text{ subject to } Ax = b.
\]

Lagrangian dual function is

\[
L(x, \gamma) = f(x) + \gamma^T (Ax - b).
\]

Dual objective:

\[
\max_{\gamma} Q(\gamma) := \min_x L(x, \gamma).
\]

Given \( \gamma^* \), the corresponding primal solution is

\[
x^* = \arg\min_x L(x, \gamma^*).
\]
Augmented Lagrangian Function

Dual Ascent

Lagrangian dual problem:

\[
\max_{\gamma} Q(\gamma) := \min_x L(x, \gamma).
\]

Based on looking at the Lagrangian dual function, we design dual ascent algorithm. At the \((t + 1)\)-th iteration

\[
x^{t+1} = \text{argmin}_x L(x, \gamma^t),
\]

\[
\gamma^{t+1} = \gamma^t + \alpha_{t+1}(Ax^{t+1} - b).
\]
Augmented Lagrangian

Augmented Lagrangian Function

Here we introduce the augmented Lagrangian function:

\[ L_{\rho}(x, \gamma) = f(x) + \gamma^T (Ax - b) + \left( \frac{\rho}{2} \right) \|Ax - b\|^2. \]

Dual algorithm:

\[ x^{t+1} = \arg\min_x L_{\rho}(x, \gamma^t), \]

\[ \gamma^{t+1} = \gamma^t + \rho (Ax^{t+1} - b). \]

This algorithm guarantees convergence under weaker assumptions than Lagrangian dual ascent.
The algorithm

Consider the following optimization problem:

\[
\min_{x \in \mathcal{X}, y \in \mathcal{Y}} f(x) + g(y), \text{ subject to } Ax + By = c.
\]

Augmented Lagrangian:

\[
L_{\rho}(x, y, \lambda) = f(x) + g(y) + \gamma^T(Ax + By - c) + (\rho/2)\|Ax + By - c\|^2.
\]

Alternating Direction Method of Multipliers (ADMM):

\[
x^{t+1} = \arg\min_{x \in \mathcal{X}} L_{\rho}(x, y^t, \gamma^t),
\]

\[
y^{t+1} = \arg\min_{y \in \mathcal{Y}} L_{\rho}(x^{t+1}, y, \gamma^t),
\]

\[
\gamma^{t+1} = \gamma^t + \rho(Ax^{t+1} + By^{t+1} - b).
\]
Some History
ADMM gets quite popular over the past 5-6 years:

However, it was actually developed in 1970’s (Not Modern Optimization at all!), which was proposed by Gabay, Mercier, Glowinski, Marrocco in 1976 explained as a special case Douglas-Rachford Splitting Method (DRSM). Eckstein and Bertsekas explained ADMM as an application of the proximal point algorithm in 1992.
Example: Lasso

ADMM form:

\[
\min_{\beta, z} \frac{1}{2} \|X\beta - y\|^2 + \lambda \|z\|_1, \text{ subject to } \beta - z = 0.
\]

ADMM:

\[
\beta^{t+1} = (X^T X + \rho I)^{-1} (X^T y + \rho z^t - \gamma^t)
\]
\[
z^{t+1} = S_{\lambda/\rho} (\beta^{t+1} + \gamma^t / \rho)
\]
\[
\gamma^{t+1} = \gamma^t + (\beta^{t+1} - z^{t+1})
\]
Example: Graphical Lasso

For sparse inverse covariance matrix estimation: Let $\hat{\Sigma}$ be the sample covariance matrix of $X \sim N(0, \Sigma)$. To estimate $\Sigma^{-1}$, assuming sparsity, we compute

$$\min_{\Omega} \text{Trace}(\hat{\Sigma}\Omega) - \log \det(\Omega) + \lambda \|\Omega\|_1.$$

ADMM form:

$$\min_{\Omega, Z} \text{Trace}(\hat{\Sigma}\Omega) - \log \det(\Omega) + \lambda \|Z\|_1, \text{ subject to } \Omega - Z = 0.$$

ADMM:

$$\Omega^{t+1} = \arg\min_{\Omega} \text{Trace}(\hat{\Sigma}\Omega) - \log \det(\Omega) + (\rho/2) \|\Omega - Z^t + U^t\|_F^2,$$

$$Z^{t+1} = S_{\lambda/\rho}(\Omega^{t+1} + U^t),$$

$$U^{t+1} = U^t + (\Omega^{t+1} - Z^{t+1}).$$
Ω-Update Step

The Ω-update step can be computed efficiently: We first compute the eigendecomposition of \( \rho(Z^t - U^t) = Q\Lambda Q^T \).

Then, we compute a diagonal matrix \( \hat{\Lambda} \)

\[
\hat{\Lambda}_{jj} = \frac{\lambda_j + \sqrt{\lambda_j^2 + 4\rho}}{2\rho}.
\]

We get

\[
\Omega^{t+1} = Q\hat{\Lambda}Q^T.
\]

Note that here the computational burden is an eigendecomposition (not efficient for large dimension).
**ADMM**

**Linearized ADMM**

For problem

\[
\min_{x \in X, y \in Y} f(x) + g(y), \text{ subject to } Ax + By = c,
\]

if \( A \in \mathbb{R}^{n \times p} \) and \( n < p \), we might want to consider the following algorithm for the \( x \)-update step:

\[
x^{t+1} = \arg\min_{x \in X} f(x) - x^T A^T \gamma^t + \rho/2 \|Ax + By^t - c\|^2 + \|x - x^t\|^2_{G/2}.
\]

For example, if \( G = \tau I - \rho A^T A \) with \( \tau > \rho \|A^T A\| \), we have the \( x \)-update step reduces to a proximal step:

\[
x^{t+1} = \arg\min_{x \in X} f(x) + \tau \left\| x - \frac{1}{\tau} \left\{ (\tau I - \rho A^T A)x^t - \rho A^T By^t + A^T \gamma^t + \rho A^T b \right\} \right\|^2.
\]
Example: Dantzig Selector

Dantzig Selector problem:

\[
\min_{\beta} \| \beta \|_1, \text{ subject to } \| X^T (X \beta - y) \|_\infty \leq \delta,
\]

where \( \delta > 0 \) is a tuning parameter and \( \| \cdot \|_\infty \) is the infinity norm.

ADMM form:

\[
\min_{\beta \in \mathbb{R}^d, z \in Z} \| \beta \|_1, \text{ subject to } X^T (X \beta - y) - z = 0, \ Z = \{ z : \| z \|_\infty \leq \delta \}.
\]

ADMM for \( \beta \)-update step:

\[
\beta^{t+1} = \arg \min_{\beta} \| \beta \|_1 + \rho \| X^T (X \beta - y) - z^t - \gamma^t / \rho \|^2 / 2.
\]

This step has no closed-form solution when \( d > n \) \( (X \in \mathbb{R}^{n \times d}) \).
Example: Dantzig Selector

Linearized ADMM:

\[
\beta^{t+1} = \arg\min_{\beta} \|\beta\|_1 + \rho \tau \|\beta - v^t\|^2 / 4, \text{ for some } v^t \text{ can be computed efficiently,}
\]

\[
= S_{2 / \rho \tau} (\beta^t - 2v^t / \rho),
\]

\[
z^{t+1} = \min \left\{ \max \left\{ X^T X \beta^{t+1} - y - \gamma^t / \rho, -\delta \right\}, \delta \right\}
\]

\[
\gamma^{t+1} = \gamma^t - \rho \left( X^t (X \beta^{t+1} - y) - z^{t+1} \right)
\]

There are many other variants of ADMM, linearize \( f \) or \( g \), online ADMM, Stochastic ADMM, Parallel ADMM,...
A Nice Review Paper

*Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* by Stephen Boyd and others, with a website with many examples with codes:
http://stanford.edu/~boyd/papers/admm_distr_stats.html

This paper has been highly influential, has been cited more than 3,900 times in 5 years.
Applications
Matrix completion finds applications in collaborative filtering, image denoising, sensor network localization and so on.
Matrix Completion

Netflix Problem

Completing the user-movie matrix will help Netflix company to make better personalized recommendation to the users.
Matrix Completion

Structural Assumption

In general, matrix completion is not possible.
In general, matrix completion is not possible. However, in many practical applications, we assume the matrix is of low-rank.
Matrix Completion

Data
Random index set and noisy observations:

\[ \Omega = \{(i_t, j_t) : t = 1, \ldots, n\}, \quad \text{and} \quad Y_{i_t, j_t} = M_{i_t, j_t}^0 + \sigma \xi_t. \]

Low-Rank Estimator

\[
\hat{M}_0 = \arg\min_{M \in \mathbb{R}^{d_1 \times d_2}} \frac{1}{n} \sum_{t=1}^{n} (Y_{i_t, j_t} - M_{i_t, j_t})^2 \\
\text{subject to rank}(M) \leq r.
\]
Nuclear-Norm Approach

Nuclear-Norm Relaxation

The rank-constrained estimator is not computationally tractable. A convex (or $\ell_1$) relaxation results the following estimator

$$\hat{M}_1 = \arg\min_{M \in \mathbb{R}^{d_1 \times d_2}} \frac{1}{n} \sum_{t=1}^{n} (Y_{it,jt} - M_{it,jt})^2$$

subject to $\|M\|_* \leq \lambda$,

where $\|M\|_*$ denotes the nuclear-norm of $M$, and is defined as the sum of singular values of $M$. 
Nuclear-Norm Approach

Theoretical Guarantee
Under some regularity conditions, assuming each entry is equally likely to be drawn, the nuclear-norm approach obtains a fast rate of convergence. Specifically, we have, with high probability,

\[
\frac{1}{d_1 d_2} \| \hat{M}_1 - M^0 \|_F^2 = O\left( \frac{rd \log d}{n} \right),
\]

where \( d = d_1 + d_2 \).

ADMM Formulation
We consider a dual formulation

$$\min_M \|M\|_*, \text{ subject to } \|M_\Omega - Y_\Omega\|_F \leq \delta.$$ 

ADMM Form:

$$\min_{M,Z \in Z} \|M\|_*, \text{ subject to } M - Z = 0, \ Z \in Z = \{Z : \|Z_\Omega - M_\Omega\|_F \leq \delta\}.$$
ADMM Implementation

ADMM can be implemented efficiently:

\[
Z^{t+1} = \begin{cases} 
Y_{jk} & \text{if } (j, k) \in \Omega, \\
M_{jk}^{t} - \frac{1}{\rho} \Gamma_{jk}^{t} & \text{otherwise}.
\end{cases}
\]

Let

\[
A^{t+1} = Z^{t+1} + \frac{1}{\rho} \Gamma^{t}, \quad \text{and} \quad A^{t+1} = U^{t+1} \Sigma^{t+1} (V^{t+1})^{T},
\]

with \( \Sigma^{t+1} = \text{diag}(\sigma_{1}^{t+1}, ..., \sigma_{r,t+1}^{t+1}) \). We have

\[
M^{t+1} = U^{t+1} \hat{\Sigma}^{t+1} (V^{t+1})^{T}, \quad \text{where} \quad \hat{\Sigma}^{t+1} = \text{diag}\left(\{\sigma_{j}^{t+1} - 1/\rho\}_{+}\right)
\]

and \((x)_{+} = \max\{0, x\}\).
Uniform Sampling Scheme
By the uniform sampling assumption, we expect to observe something like this...

<table>
<thead>
<tr>
<th></th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie 3</th>
<th>Movie 4</th>
<th>Movie 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>5</td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>User 2</td>
<td></td>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>User 3</td>
<td>1</td>
<td></td>
<td>4</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>User 4</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>User 5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Real Data
A data collected from five PhD students....

<table>
<thead>
<tr>
<th></th>
<th>Prison Break</th>
<th>House of Cards</th>
<th>Game of Thrones</th>
<th>True Detective</th>
<th>Big Bang Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Juan</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Junwei</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Yang</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Cagin</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Contributions

- New estimator which achieves fast rates of convergence under both uniform and non-uniform sampling schemes.

- Scalable algorithm to compute the estimator.
Max-Norm Approach

Max-Norm

It is unclear if the nuclear-norm is the best candidate for convex relaxation. Another candidate to replace $\text{rank}(M)$ is the matrix max-norm

$$\|M\|_{\text{max}} = \min_{M=UV^T} \|U\|_{2,\infty} \|V\|_{2,\infty},$$

where $\|U\|_{2,\infty} = \max_{k=1,...,d_1} \left(\sum_{\ell=1}^{d} U_{k,\ell}^2\right)^{1/2}$. 
Max-Norm Approach

Max-Norm
It is unclear if the nuclear-norm is the best candidate for convex relaxation. Another candidate to replace \( \text{rank}(M) \) is the matrix max-norm

\[
\| M \|_{\text{max}} = \min_{M = UV^T} \| U \|_{2,\infty} \| V \|_{2,\infty},
\]

where \( \| U \|_{2,\infty} = \max_{k=1,...,d_1} \left( \sum_{\ell=1}^{d} U_{k,\ell}^2 \right)^{1/2} \).

Comparison
Nuclear-norm:

\[
\| M \|_* = \inf \left\{ \sum_j |\sigma_j| : M = \sum_j \sigma_j u_j v_j^T, \| u_j \| = \| v_j \| = 1 \right\}.
\]
Max-Norm Approach

Max-Norm
It is unclear if the nuclear-norm is the best candidate for convex relaxation. Another candidate to replace \( \text{rank}(M) \) is the matrix max-norm

$$
\|M\|_{\text{max}} = \min_{M = UV^T} \|U\|_{2,\infty} \|V\|_{2,\infty},
$$

where \( \|U\|_{2,\infty} = \max_{k=1,\ldots,d_1} \left( \sum_{\ell=1}^{d} U_{k\ell}^2 \right)^{1/2} \).

Comparison
Nuclear-norm:

$$
\|M\|_* = \inf \left\{ \sum_j |\sigma_j| : M = \sum_j \sigma_j u_j v_j^T, \|u_j\| = \|v_j\| = 1 \right\}.
$$

Max-norm:

$$
\|M\|_{\text{max}} \asymp \inf \left\{ \sum_j |\sigma_j| : M = \sum_j \sigma_j u_j v_j^T, \|u_j\|_{\infty} = \|v_j\|_{\infty} = 1 \right\}.
$$
Max-Norm Estimator

Cai & Zhou (2013) proposed the following estimator

$$\hat{M}_{\text{max}} = \arg\min_{M \in \mathbb{R}^{d_1 \times d_2}} \frac{1}{n} \sum_{t=1}^{n} (Y_{i_t,j_t} - M_{i_t,j_t})^2, \text{ subject to } M \in \mathcal{K}(\alpha, R),$$

where $\mathcal{K}(\alpha, R) = \{ M \in \mathbb{R}^{d_1 \times d_2} : \|M\|_\infty \leq \alpha, \|M\|_{\text{max}} \leq R \}$ with $\alpha$ a prespecified upper bound for the elementwise $\ell_\infty$-norm of $M^0$ and $R > 0$ a tuning parameter.
Max-Norm Approach

Model
Suppose that a random index set

$$\Omega = \{(i_1, j_1), (i, j), \ldots, (i_n, j_n)\} \subset ([d_1] \times [d])^n$$

is drawn independently according to $$\Pi = \{\pi_{k\ell}\}_{1 \leq k \leq d_1, 1 \leq \ell \leq d},$$ with replacement, i.e.,

$$\mathbb{P}\{(i_t, j_t) = (k, \ell)\} = \pi_{k\ell} \text{ for all } t = 1, \ldots, n.$$ 

Given $$\Omega,$$ we observe noisy entries $$\{Y_{i_t, j_t}\}_{t=1}^n$$:

$$Y_{i_t, j_t} = M_{i_t, j_t}^0 + \sigma \xi_t, \quad t = 1, \ldots, n,$$

for some $$\sigma > 0,$$ and the noise $$\xi_t$$'s are independent with $$\mathbb{E}(\xi_t) = 0$$ and $$\mathbb{E}(\xi_t^2) = 1.$$
Max-Norm Approach

Max-Norm Recovery

(Cai & Zhou, 2013) Assume \( \pi_{k,\ell} \geq \frac{1}{\nu d_1 d_2} \), and \( M^0 \in \mathcal{K}(\alpha, R) \). Then, we have, with high probability,

\[
\frac{1}{d_1 d_2} \| \hat{M}_{\text{max}} - M^0 \|_F^2 \leq \mathcal{O} \left( \sqrt{\frac{R^2 d}{n}} \right),
\]

where \( d = d_1 + d_2 \).
Max-Norm Approach

Max-Norm Recovery

(Cai & Zhou, 2013) Assume $\pi_{k,\ell} \geq \frac{1}{\nu d_1 d_2}$, and $M^0 \in K(\alpha, R)$. Then, we have, with high probability,

$$\frac{1}{d_1 d_2} \| \hat{M}_{\text{max}} - M^0 \|_F^2 \leq O\left(\sqrt{\frac{R^2 d}{n}}\right),$$

where $d = d_1 + d_2$.

Disadvantages

- Sub-optimal for uniform sampling scheme.
- No scalable computational tools.
Hybrid Approach

A Hybrid Estimator

We propose the following estimator:

\[ \hat{M} := \arg\min_{M \in \mathbb{R}^{d_1 \times d_2}} \frac{1}{n} \sum_{t=1}^{n} (Y_{i_t,j_t} - M_{i_t,j_t})^2 + \lambda \|M\|_{\text{max}} + \mu \|M\|_*, \]

subject to \( \|M\|_\infty \leq \alpha. \)
Hybrid Approach

Exact Low-Rank Recovery

Assume \( \|M^0\|_\infty \leq \alpha \), \( \text{rank}(M^0) \leq r_0 \) and \( \xi_1, ..., \xi_n \) are i.i.d. \( N(0, 1) \) random variables. If \( \pi_{k\ell} \geq (\nu d_1 d_2)^{-1} \), for properly chosen \( \lambda \) and \( \mu \), we have, with high probability,

\[
\frac{1}{d_1 d_2} \| \hat{M} - M^0 \|_F^2 = \mathcal{O}\left( \sqrt{\frac{R^2 d}{n}} \right).
\]
Hybrid Approach

Exact Low-Rank Recovery

Assume $\|M^0\|_\infty \leq \alpha$, rank$(M^0) \leq r_0$ and $\xi_1, ..., \xi_n$ are i.i.d. $N(0, 1)$ random variables. If $\pi_{k\ell} \geq (\nu d_1 d_2)^{-1}$, for properly chosen $\lambda$ and $\mu$, we have, with high probability,

$$\frac{1}{d_1 d_2} \|\hat{M} - M^0\|_F^2 = O\left(\sqrt{\frac{R^2 d}{n}}\right).$$

If $\pi_{k\ell} = (d_1 d_2)^{-1}$, we have, with high probability,

$$\frac{1}{d_1 d_2} \|\hat{M} - M^0\|_F^2 = O\left(\frac{r_0 d \log d}{n}\right).$$
SDP Formulation

Max-Norm
Matrix max-norm is

$$\|M\|_{\text{max}} = \min_{M=UV^T} \|U\|_{2,\infty} \|V\|_{2,\infty},$$

where $$\|U\|_{2,\infty} = \max_{k=1,...,d_1} \left(\sum_{\ell=1}^{d} U_{k\ell}^2\right)^{1/2}.$$ 

An SDP formulation
The max-norm can be computed via solving an SDP problem

$$\|A\|_{\text{max}} = \min R,$$

subject to

$$\begin{pmatrix} W_1 & A \\ A^T & W_2 \end{pmatrix} \succeq 0, \quad \|\text{diag}(W_1)\|_{\infty} \leq R, \quad \|\text{diag}(W_2)\|_{\infty} \leq R.$$
SDP Formulation

Max-Norm
Matrix max-norm is

$$\|M\|_{\text{max}} = \min_{M = UV^T} \|U\|_{2,\infty} \|V\|_{2,\infty},$$

where $$\|U\|_{2,\infty} = \max_{k=1,...,d_1} \left( \sum_{\ell=1}^{d} U_{k\ell}^2 \right)^{1/2}.$$ 

An SDP formulation
The max-norm can be computed via solving an SDP problem

$$\|A\|_{\text{max}} = \min R,$$

subject to $$(W_1 \begin{bmatrix} A \\ A^T \end{bmatrix} W_2) \succeq 0, \|\text{diag}(W_1)\|_{\infty} \leq R, \|\text{diag}(W_2)\|_{\infty} \leq R.$$ 

$$\|A\|_{\ast} = \frac{1}{2} \min \text{Tr}(W_1) + \text{Tr}(W_2)$$

subject to $$(W_1 \begin{bmatrix} A \\ A^T \end{bmatrix} W_2) \succeq 0.$$
SDP Formulation

Hybrid Estimator

The proposed hybrid estimator can be computed by solving as an SDP problem:

$$\min_{Z \in \mathbb{R}^{d \times d}} \frac{1}{2} \sum_{t=1}^{n} (Y_{i_t,j_t} - Z_{i_t,j_t}^{12})^2 + \lambda \| \text{diag}(Z) \|_{\infty} + \mu \langle I, Z \rangle,$$

subject to $\| Z^{12} \|_{\infty} \leq \alpha, \ Z \succeq 0,$

where $d = d_1 + d_2,$ and

$$Z = \begin{pmatrix} Z^{11} & Z^{12} \\ (Z^{12})^T & Z^{22} \end{pmatrix}, \ Z^{11} \in \mathbb{R}^{d_1 \times d_1}, \ Z^{12} \in \mathbb{R}^{d_1 \times d_2} \text{ and } Z^{22} \in \mathbb{R}^{d \times d}.$$
A Reformulation
To efficiently solve the problem, it is important to properly reformulate the problem as

$$\min_{X,Z} \frac{1}{2} \sum_{t=1}^{n} (Y_{i_t,j_t} - Z_{i_t,j_t}^{12})^2 + \lambda \|\text{diag}(Z)\|_{\infty} + \mu \langle I, X \rangle,$$

subject to $X \succeq 0$, $Z \in \mathcal{P} = \{Z \in S^d : \|Z^{12}\|_{\infty} \leq \alpha\}$,

$X - Z = 0.$
Algorithm

Augmented Lagrangian

Denote by $\mathcal{L}(Z) = \frac{1}{2} \sum_{t=1}^{n} (Y_{i_t,j_t} - Z_{i_t,j_t}^{12})^2 + \lambda \|\text{diag}(Z)\|_{\infty}$. The augmented Lagrangian function of the reformulated problem is

$$L(X, Z; W) = \mathcal{L}(Z) + \mu \langle I, X \rangle + \langle W, X - Z \rangle + \frac{\rho}{2} \|X - Z\|_F^2,$$

where $X \in \mathcal{S}_+^d$, $Z \in \mathcal{P}$. 

ADMM

At the $t$-th iteration, we update $(X, Z; W)$ by

$$X^{t+1} = \arg\min_{X \in \mathcal{S}_+^d} \mathcal{L}(X, Z^t; W^t) = \Pi_{\mathcal{S}_+^d} \left\{ Z^t - \rho^{-1} \left( W^t + \mu I \right) \right\},$$

$$Z^{t+1} = \arg\min_{Z \in \mathcal{P}} \mathcal{L}(X^{t+1}, Z; W^t) = \arg\min_{Z \in \mathcal{P}} \mathcal{L}(Z) + \frac{\rho}{2} \|X^{t+1} - Z\|_F^2,$$

$$W^{t+1} = W^t + \tau \rho (X^{t+1} - Z^{t+1}).$$
Algorithm

Augmented Lagrangian
Denote by $\mathcal{L}(Z) = \frac{1}{2} \sum_{t=1}^{n} (Y_{i_t,j_t} - Z_{i_t,j_t}^{12})^2 + \lambda \|\text{diag}(Z)\|_\infty$. The augmented Lagrangian function of the reformulated problem is

$$L(X, Z; W) = \mathcal{L}(Z) + \mu \langle I, X \rangle + \langle W, X - Z \rangle + \frac{\rho}{2} \|X - Z\|_F^2,$$

$$X \in \mathcal{S}^d_+, \ Z \in \mathcal{P}.$$

ADMM
At the $t$-th iteration, we update $(X, Z; W)$ by

$$X^{t+1} = \arg\min_{X \in \mathcal{S}^d_+} L(X, Z^t; W^t) = \Pi_{\mathcal{S}^d_+} \{Z^t - \rho^{-1}(W^t + \mu I)\},$$

$$Z^{t+1} = \arg\min_{Z \in \mathcal{P}} L(X^{t+1}, Z; W^t) = \arg\min_{Z \in \mathcal{P}} \mathcal{L}(Z) + \frac{\rho}{2} \|Z - X^{t+1} - \rho^{-1}W^t\|_F^2,$$

$$W^{t+1} = W^t + \tau \rho (X^{t+1} - Z^{t+1}).$$
Solving the Subproblem

Denote the observed set of indices of \( M^0 \) by \( \Omega = \{(i_t, j_t)\}_{t=1}^n \). For a given matrix \( C \in \mathbb{R}^{d \times d} \), we have

\[
Z(C) = \arg\min_{Z \in \mathcal{P}} \mathcal{L}(Z) + \frac{\rho}{2} \|Z - C\|_F^2,
\]

where

\[
Z(C) = \begin{pmatrix}
Z_{11}(C) & Z_{12}(C) \\
Z_{12}(C)^T & Z_{22}(C)
\end{pmatrix}
\]

\[
Z_{12}(C) = \begin{cases}
\Pi[-\alpha, \alpha]\left(\frac{Y_{k\ell} + \rho C_{k\ell}^{12}}{1 + \rho}\right), & \text{if } (k, \ell) \in \Omega, \\
\Pi[-\alpha, \alpha](C_{k\ell}^{12}), & \text{otherwise,}
\end{cases}
\]

\[
Z_{k\ell}(C) = \begin{cases}
C_{k\ell}^{11}, & \text{if } k \neq \ell, \\
Z_{k\ell}(C) = C_{k\ell}^{22}, & \text{if } k \neq \ell,
\end{cases}
\]

\[
\text{diag}\{Z(C)\} = \arg\min_{z \in \mathbb{R}^d} \lambda \|z\|_\infty + \frac{\rho}{2} \|\text{diag}(C) - z\|^2,
\]

and \( \Pi_{[a,b]}(x) = \min\{b, \max(a, x)\} \) projects \( x \in \mathbb{R} \) to the interval \([a, b]\).
Simulation

Low-Rank Matrix
Let $M^0 = M_L M_R^T$, where $M_L, M_R \in \mathbb{R}^{d_t \times r}$, and each entry is sampled from $N(0, 1)$. Thus, $M^0$ is a rank $r$ matrix.

Non-Uniform Sampling
We conduct non-uniform sampling schemes in the following way. For each $(k, \ell)$, let $\pi_{k\ell} = p_k p_{\ell}$, where we define

$$p_k = \begin{cases} 
3p_0 & \text{if } k \leq \frac{d_t}{10} \\
9p_0 & \text{if } \frac{d_t}{10} < k \leq \frac{d_t}{5} \\
p_0 & \text{otherwise.}
\end{cases}$$

To evaluate the results, we adopt the metric of relative error (RE) that

$$\text{RE} = \frac{\| \hat{M} - M^0 \|_F}{\| M^0 \|_F}.$$
## Simulation

### Non-Uniform Sampling

<table>
<thead>
<tr>
<th>$d_t$</th>
<th>$(r, SR)$</th>
<th>Nuclear RE</th>
<th>Max RE</th>
<th>Hybrid RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>$(5, 0.20)$</td>
<td>$6.0 \times 10^{-1}$</td>
<td>$1.6 \times 10^{-1}$</td>
<td>$1.0 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$(8, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.8 \times 10^{-1}$</td>
<td>$1.3 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$(10, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.9 \times 10^{-1}$</td>
<td>$1.4 \times 10^{-1}$</td>
</tr>
<tr>
<td>750</td>
<td>$(3, 0.20)$</td>
<td>$6.0 \times 10^{-1}$</td>
<td>$1.7 \times 10^{-1}$</td>
<td>$1.2 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$(5, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.4 \times 10^{-1}$</td>
<td>$9.2 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$(8, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.5 \times 10^{-1}$</td>
<td>$1.0 \times 10^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$(10, 0.20)$</td>
<td>$6.0 \times 10^{-1}$</td>
<td>$1.5 \times 10^{-1}$</td>
<td>$1.0 \times 10^{-1}$</td>
</tr>
<tr>
<td>1000</td>
<td>$(3, 0.20)$</td>
<td>$6.0 \times 10^{-1}$</td>
<td>$2.0 \times 10^{-1}$</td>
<td>$9.4 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$(5, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.9 \times 10^{-1}$</td>
<td>$9.8 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$(8, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.5 \times 10^{-1}$</td>
<td>$7.0 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$(10, 0.20)$</td>
<td>$6.1 \times 10^{-1}$</td>
<td>$1.6 \times 10^{-1}$</td>
<td>$8.9 \times 10^{-2}$</td>
</tr>
</tbody>
</table>
Simulation

Jester Joke Data
jester-1: 24,938 users who rate 36 or more of 100 jokes;
jester-2: 23,500 users who rate 36 or more of 100 jokes.
We randomly select $n_u$ users, and we use normalized mean absolute error (NMAE) to measure the result:

$$
NMAE = \frac{\sum_{(j,k) \in \Omega} |\hat{M}_{jk} - M^0_{jk}|}{|\Omega| (r_{\text{max}} - r_{\text{min}})}
$$

$\Omega$: Available entries
$r_{\text{max}}$: Maximum rating
$r_{\text{min}}$: Minimum rating
## Simulation

### Jester Joke Data

<table>
<thead>
<tr>
<th>Example</th>
<th>$(n_u, \text{SR})$</th>
<th>Nuclear NMAE</th>
<th>Max NMAE</th>
<th>Hybrid NMAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>jester-1</td>
<td>(1500, 0.20)</td>
<td>0.143</td>
<td>0.129</td>
<td>0.129</td>
</tr>
<tr>
<td></td>
<td>(1500, 0.25)</td>
<td>0.138</td>
<td>0.124</td>
<td>0.123</td>
</tr>
<tr>
<td></td>
<td>(2000, 0.20)</td>
<td>0.142</td>
<td>0.128</td>
<td>0.127</td>
</tr>
<tr>
<td></td>
<td>(2000, 0.25)</td>
<td>0.137</td>
<td>0.124</td>
<td>0.121</td>
</tr>
<tr>
<td>jester-2</td>
<td>(1500, 0.20)</td>
<td>0.144</td>
<td>0.130</td>
<td>0.130</td>
</tr>
<tr>
<td></td>
<td>(1500, 0.25)</td>
<td>0.139</td>
<td>0.124</td>
<td>0.122</td>
</tr>
<tr>
<td></td>
<td>(2000, 0.20)</td>
<td>0.143</td>
<td>0.129</td>
<td>0.130</td>
</tr>
<tr>
<td></td>
<td>(2000, 0.25)</td>
<td>0.139</td>
<td>0.123</td>
<td>0.125</td>
</tr>
</tbody>
</table>
**Movie-100K**

Movie-100K dataset contains 100,000 ratings for 1,682 movies by 943 users.

<table>
<thead>
<tr>
<th>SR</th>
<th>Nuclear NMAE</th>
<th>Max NMAE</th>
<th>Hybrid NMAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.242</td>
<td>0.220</td>
<td>0.221</td>
</tr>
<tr>
<td>0.15</td>
<td>0.237</td>
<td>0.214</td>
<td>0.212</td>
</tr>
<tr>
<td>0.20</td>
<td>0.231</td>
<td>0.209</td>
<td>0.210</td>
</tr>
<tr>
<td>0.25</td>
<td>0.223</td>
<td>0.206</td>
<td>0.205</td>
</tr>
</tbody>
</table>