First-Order Methods for Differentiable "Nonsmooth" Convex Optimization: A Tale of Frank-Wolfe and Multiplicative-Gradient

Renbo Zhao

MIT Operations Research Center

Department of Mathematical Sciences
Rensselaer Polytechnic Institute

December, 2022

Binary Classification

- Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples

Binary Classification

- Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples
  - For $i \in [m]$, $x_i \in \mathbb{R}^n$ is the feature vector and $y_i \in \{-1, 1\}$ is the (binary) label

Binary Classification

- Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples
  - For $i \in [m]$, $x_i \in \mathbb{R}^n$ is the feature vector and $y_i \in \{-1, 1\}$ is the (binary) label
  - We wish to build/train a statistical model $\mathsf{M}(\cdot;\theta)$ with input x, output y and model parameter θ

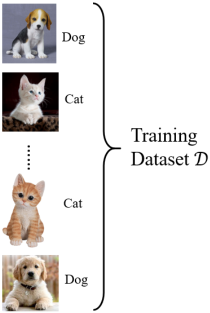
Binary Classification

- Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples
  - For $i \in [m]$, $x_i \in \mathbb{R}^n$ is the feature vector and $y_i \in \{-1, 1\}$ is the (binary) label
  - We wish to build/train a statistical model $\mathsf{M}(\cdot;\theta)$ with input x, output y and model parameter θ
  - Given $x_{\text{new}} \in \mathbb{R}^n$, $\hat{y} = \mathsf{M}(x_{\text{new}};\theta)$ is the classified label

Binary Classification

▷ Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples

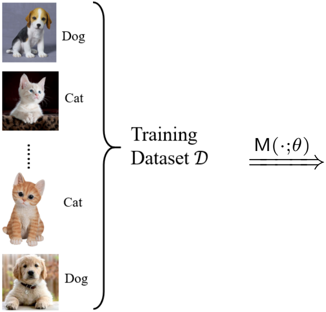
- For $i \in [m]$, $x_i \in \mathbb{R}^n$ is the feature vector and $y_i \in \{-1, 1\}$ is the (binary) label
- We wish to build/train a statistical model $\mathsf{M}(\cdot;\theta)$ with input x, output y and model parameter θ
- Given $x_{\text{new}} \in \mathbb{R}^n$, $\hat{y} = \mathsf{M}(x_{\text{new}};\theta)$ is the classified label



Binary Classification

▷ Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples

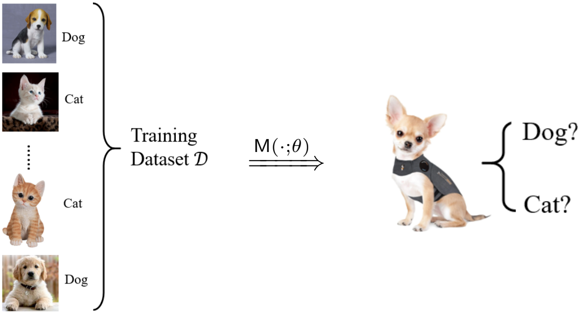
- For $i \in [m]$, $x_i \in \mathbb{R}^n$ is the feature vector and $y_i \in \{-1, 1\}$ is the (binary) label
- We wish to build/train a statistical model $\mathsf{M}(\cdot;\theta)$ with input x, output y and model parameter θ
- Given $x_{\text{new}} \in \mathbb{R}^n$, $\hat{y} = \mathsf{M}(x_{\text{new}};\theta)$ is the classified label



Binary Classification

▷ Given a training dataset $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^m$ with m samples

- For $i \in [m]$, $x_i \in \mathbb{R}^n$ is the feature vector and $y_i \in \{-1, 1\}$ is the (binary) label
- We wish to build/train a statistical model $\mathsf{M}(\cdot;\theta)$ with input x, output y and model parameter θ
- Given $x_{\text{new}} \in \mathbb{R}^n$, $\hat{y} = \mathsf{M}(x_{\text{new}};\theta)$ is the classified label



Model Training via Gradient Methods (GMs)

Model Training via Gradient Methods (GMs)

▷ Training $\mathsf{M}(\cdot;\theta)$ typically involves solving the "model fitting" problem

$$\min_{\theta\in\Theta} \; f_{\mathsf{M},\mathcal{D}}(\theta) \qquad \texttt{(Training)}$$

Model Training via Gradient Methods (GMs)

- Training $\mathsf{M}(\cdot;\theta)$ typically involves solving the "model fitting" problem

$$\min_{\theta\in\Theta} \; f_{\mathsf{M},\mathcal{D}}(\theta) \qquad \texttt{(Training)}$$

- Gradient methods (GMs) are appealing in solving (Training):

Model Training via Gradient Methods (GMs)

▷ Training $\mathsf{M}(\cdot;\theta)$ typically involves solving the "model fitting" problem

$$\min_{\theta\in\Theta}\ f_{\mathsf{M},\mathcal{D}}(\theta) \qquad \texttt{(Training)}$$

▷ Gradient methods (GMs) are appealing in solving (Training):

- In modern applications, both m and n can be huge

Model Training via Gradient Methods (GMs)

▷ Training $\mathsf{M}(\cdot;\theta)$ typically involves solving the "model fitting" problem

$$\min_{\theta\in\Theta} \; f_{\mathsf{M},\mathcal{D}}(\theta) \qquad \texttt{(Training)}$$

▷ Gradient methods (GMs) are appealing in solving (Training):

- In modern applications, both m and n can be huge
- Gradient methods only involve computing and manipulating gradients of $f_{\mathsf{M},\mathcal{D}}(\cdot)$, hence have low-computational cost per iteration

Model Training via Gradient Methods (GMs)

▷ Training $\mathsf{M}(\cdot;\theta)$ typically involves solving the "model fitting" problem

$$\min_{\theta\in\Theta} \; f_{\mathsf{M},\mathcal{D}}(\theta) \qquad \texttt{(Training)}$$

▷ Gradient methods (GMs) are appealing in solving (Training):

- In modern applications, both m and n can be huge
- Gradient methods only involve computing and manipulating gradients of $f_{\mathsf{M},\mathcal{D}}(\cdot)$, hence have low-computational cost per iteration
- Gradient methods have reasonably fast convergence rate to achieve low-to-medium accuracy

Model Training via Gradient Methods (GMs)

▷ Training $\mathsf{M}(\cdot;\theta)$ typically involves solving the "model fitting" problem

$$\min_{\theta\in\Theta} \; f_{\mathsf{M},\mathcal{D}}(\theta) \qquad \texttt{(Training)}$$

▷ Gradient methods (GMs) are appealing in solving (Training):

- In modern applications, both m and n can be huge
- Gradient methods only involve computing and manipulating gradients of $f_{\mathsf{M},\mathcal{D}}(\cdot)$, hence have low-computational cost per iteration
- Gradient methods have reasonably fast convergence rate to achieve low-to-medium accuracy

▷ If $f_{\mathsf{M},\mathcal{D}}(\cdot)$ is non-differentiable, (Training) can be solved by subgradient methods

Canonical Model: Logistic Regression

Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1)$$

Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1-\beta}\right)\right]$$

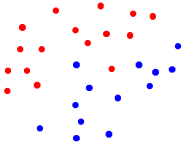
Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1-\beta}\right)\right]$$



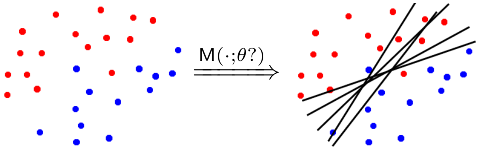
Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1-\beta}\right)\right]$$



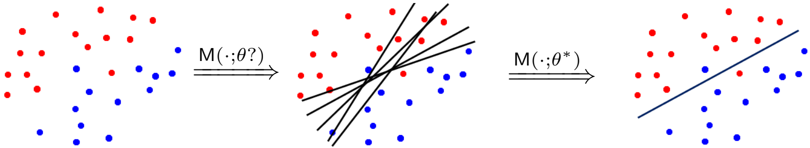
Canonical Model: Logistic Regression

▷ In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1 - \beta}\right) \right]$$



Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1-\beta}\right)\right]$$

- Given the training dataset $\mathcal{D}$, we determine the parameter $\theta = (w, b)$ via maximum-likelihood estimation, which turns out to be:

$$f^*_{\mathrm{LR}} := \min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \tfrac{1}{m} \textstyle\sum_{i=1}^m \ln\left(1 + \exp(-y_i(w^\top x_i + b))\right)\right\} \quad (\texttt{LR})$$

Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1-\beta}\right)\right]$$

- Given the training dataset $\mathcal{D}$, we determine the parameter $\theta = (w, b)$ via maximum-likelihood estimation, which turns out to be:

$$f^*_{\mathrm{LR}} := \min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \frac{1}{m} \sum_{i=1}^m \ln\left(1 + \exp(-y_i(w^\top x_i + b))\right)\right\} \quad (\texttt{LR})$$

- A critical observation: $f_{\mathrm{LR}}(\cdot)$ is convex and "smooth" on $\mathbb{R}^{n+1}$.

Canonical Model: Logistic Regression

- In logistic regression, the model parameter $\theta = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ and the underlying model is given as follows:

$$\Pr(y = 1|x) := \frac{1}{1 + \exp(-(w^\top x + b))}$$

and we output label $y = 1$ if

$$\Pr(y = 1|x) > \beta \in (0, 1) \qquad \left[\iff w^\top x + b > \ln\left(\frac{\beta}{1-\beta}\right)\right]$$

- Given the training dataset $\mathcal{D}$, we determine the parameter $\theta = (w, b)$ via maximum-likelihood estimation, which turns out to be:

$$f^*_{\mathrm{LR}} := \min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \tfrac{1}{m} \textstyle\sum_{i=1}^m \ln\left(1 + \exp(-y_i(w^\top x_i + b))\right)\right\} \quad (\texttt{LR})$$

- A critical observation: $f_{\mathrm{LR}}(\cdot)$ is convex and "smooth" on $\mathbb{R}^{n+1}$.

- By "smooth", we mean $f_{\mathrm{LR}}(\cdot)$ has *Lipschitz gradient* on $\mathbb{R}^{n+1}$:

$$\|\nabla f_{\mathrm{LR}}(\theta) - \nabla f_{\mathrm{LR}}(\theta')\| \le L\|\theta - \theta'\|, \quad \forall\, \theta, \theta' \in \mathbb{R}^{n+1} \quad (\texttt{LG})$$

where $L = \frac{1}{4m}\sum_{i=1}^m (\|x_i\|^2 + 1)$ is called the *smoothness parameter* of $f_{\mathrm{LR}}(\cdot)$.

Gradient Method for Logistic Regression

$$\boxed{\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{f_{\rm LR}(\theta) := \frac{1}{m}\sum_{i=1}^m \ln\Big(1+\exp(-y_i(w^\top x_i+b))\Big)\right\}} \qquad (\texttt{LR})$$

Gradient Method for Logistic Regression

$$\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \frac{1}{m}\sum_{i=1}^{m} \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\} \qquad (\texttt{LR})$$

- We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\mathrm{LR}}(\theta^t), \quad \forall\, t \geq 0 \qquad (\texttt{GM})$$

Gradient Method for Logistic Regression

$$\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \frac{1}{m}\sum_{i=1}^{m} \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\} \quad \text{(LR)}$$

- We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\mathrm{LR}}(\theta^t), \quad \forall\, t \geq 0 \quad \text{(GM)}$$

- Based on (LG), typically choose step-size $\alpha_t = 1/L$ for all $t \geq 0$.

Gradient Method for Logistic Regression

$$\boxed{\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \frac{1}{m}\sum_{i=1}^m \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\}} \qquad \texttt{(LR)}$$

- We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\mathrm{LR}}(\theta^t), \quad \forall\, t \ge 0 \qquad \texttt{(GM)}$$

- Based on (LG), typically choose step-size $\alpha_t = 1/L$ for all $t \ge 0$.

- Let θ^* be an optimal solution of (LR). Computational guarantee of (GM):

$$f_{\mathrm{LR}}(\theta^t) - f_{\mathrm{LR}}(\theta^*) \le 2L\|\theta^0 - \theta^*\|^2/t, \quad \forall\, t \ge 1$$

Gradient Method for Logistic Regression

$$\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\rm LR}(\theta) := \tfrac{1}{m}\sum_{i=1}^m \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\} \qquad \texttt{(LR)}$$

- We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\rm LR}(\theta^t), \quad \forall\, t \ge 0 \qquad \texttt{(GM)}$$

- Based on (LG), typically choose step-size $\alpha_t = 1/L$ for all $t \ge 0$.

- Let θ^* be an optimal solution of (LR). Computational guarantee of (GM):

$$f_{\rm LR}(\theta^t) - f_{\rm LR}(\theta^*) \le 2L\|\theta^0 - \theta^*\|^2/t, \quad \forall\, t \ge 1$$

- Several improvement available:

Gradient Method for Logistic Regression

$$\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\text{LR}}(\theta) := \frac{1}{m}\sum_{i=1}^m \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\} \qquad \texttt{(LR)}$$

▷ We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\text{LR}}(\theta^t), \quad \forall\, t \geq 0 \qquad \texttt{(GM)}$$

▷ Based on (LG), typically choose step-size $\alpha_t = 1/L$ for all $t \geq 0$.

▷ Let θ^* be an optimal solution of (LR). Computational guarantee of (GM):

$$f_{\text{LR}}(\theta^t) - f_{\text{LR}}(\theta^*) \leq 2L\|\theta^0 - \theta^*\|^2/t, \quad \forall\, t \geq 1$$

▷ Several improvement available:

- Nesterov's acceleration $\longrightarrow$ convergence rate $O(1/t^2)$

Gradient Method for Logistic Regression

$$\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\text{LR}}(\theta) := \frac{1}{m}\sum_{i=1}^{m} \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\} \quad \text{(LR)}$$

- ▷ We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\text{LR}}(\theta^t), \quad \forall\, t \geq 0 \quad \text{(GM)}$$

- ▷ Based on (LG), typically choose step-size $\alpha_t = 1/L$ for all $t \geq 0$.

- ▷ Let θ^* be an optimal solution of (LR). Computational guarantee of (GM):

$$f_{\text{LR}}(\theta^t) - f_{\text{LR}}(\theta^*) \leq 2L\|\theta^0 - \theta^*\|^2/t, \quad \forall\, t \geq 1$$

- ▷ Several improvement available:
  - Nesterov's acceleration $\longrightarrow$ convergence rate $O(1/t^2)$
  - Regime where the number of data samples m is large
    $\longrightarrow$ stochastic gradient method

Gradient Method for Logistic Regression

$$\min_{\theta=(w,b)\in\mathbb{R}^{n+1}} \left\{ f_{\mathrm{LR}}(\theta) := \tfrac{1}{m}\textstyle\sum_{i=1}^m \ln\left(1+\exp(-y_i(w^\top x_i + b))\right)\right\} \qquad \texttt{(LR)}$$

▷ We can use gradient method for solving (LR)

$$\text{Start with } \theta^0 \quad \longrightarrow \quad \theta^{t+1} := \theta^t - \alpha_t \nabla f_{\mathrm{LR}}(\theta^t), \quad \forall\, t \ge 0 \qquad \texttt{(GM)}$$

▷ Based on (LG), typically choose step-size $\alpha_t = 1/L$ for all $t \ge 0$.

▷ Let θ^* be an optimal solution of (LR). Computational guarantee of (GM):

$$f_{\mathrm{LR}}(\theta^t) - f_{\mathrm{LR}}(\theta^*) \le 2L\|\theta^0 - \theta^*\|^2/t, \quad \forall\, t \ge 1$$

▷ Several improvement available:

- Nesterov's acceleration $\longrightarrow$ convergence rate $O(1/t^2)$
- Regime where the number of data samples m is large
  $\longrightarrow$ stochastic gradient method
- Regime where dimension of θ is $n+1$ is large
  $\longrightarrow$ coordinate gradient method

Fundamental Limitation of “Stanadard” (GM)

$$(\texttt{GM}): \quad \alpha_t = \frac{1}{L}, \qquad f_{\mathrm{LR}}(\theta^t) - f_{\mathrm{LR}}(\theta^*) \le \frac{2L\|\theta^0 - \theta^*\|^2}{t}$$

The Lipschitz-gradient property plays a fundamental role in (GM):

Fundamental Limitation of "Stanadard" (GM)

$$(\texttt{GM}): \quad \alpha_t = \frac{1}{L}, \qquad f_{\text{LR}}(\theta^t) - f_{\text{LR}}(\theta^*) \le \frac{2L\|\theta^0 - \theta^*\|^2}{t}$$

The Lipschitz-gradient property plays a fundamental role in (GM):

- ▷ The smoothness parameter L appears in both *step-size* and *computational guarantees*.

Fundamental Limitation of "Stanadard" (GM)

$$(\texttt{GM}): \quad \alpha_t = \frac{1}{L}\ , \qquad f_{\mathrm{LR}}(\theta^t) - f_{\mathrm{LR}}(\theta^*) \le \frac{2L\|\theta^0 - \theta^*\|^2}{t}$$

The Lipschitz-gradient property plays a fundamental role in (GM):

- ▷ The smoothness parameter L appears in both *step-size* and *computational guarantees*.
- ▷ This property is also critical in ensuring sufficient decrease in line search.

Fundamental Limitation of "Stanadard" (GM)

$$(\texttt{GM}): \quad \alpha_t = \frac{1}{L}, \qquad f_{\rm LR}(\theta^t) - f_{\rm LR}(\theta^*) \le \frac{2L\|\theta^0 - \theta^*\|^2}{t}$$

The Lipschitz-gradient property plays a fundamental role in (GM):

- ▷ The smoothness parameter L appears in both *step-size* and *computational guarantees*.
- ▷ This property is also critical in ensuring sufficient decrease in line search.
- ▷ Without property, (GM) may fail both in *theory* and *practice*, and the same applies to its variants (e.g., accelerated, stochastic and coordinate versions).

Many Important Applications are "Non-Standard"

However, there are *many* important applications that do not have the Lipschitz-gradient property:

Many Important Applications are "Non-Standard"

However, there are *many* important applications that do not have the Lipschitz-gradient property:

- Learning of Multivariate Hawkes Process
- Positron Emission Tomography
- Poisson Image Deblurring with TV Regularization
- Nesterov's Semidefinite Relaxation of Boolean Quadratic Program (QP)
- D-optimal Design
- Quantum State Tomography
- ......

Many Important Applications are "Non-Standard"

However, there are *many* important applications that do not have the Lipschitz-gradient property:

- Learning of Multivariate Hawkes Process
- Positron Emission Tomography
- Poisson Image Deblurring with TV Regularization
- Nesterov's Semidefinite Relaxation of Boolean Quadratic Program (QP)
- D-optimal Design
- Quantum State Tomography
- ......

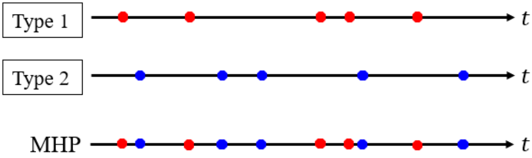
Let us briefly examine several of these problems ...

Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.

Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.



Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.
- MHPs are both self-exciting and mutually-exciting.

Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.

- MHPs are both self-exciting and mutually-exciting.
  - Occurrence of one type of events (say type 1) increases the chance of occurrence of both *this* type of events and *other* type of events (say type 2) in the future.

Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.
- MHPs are both self-exciting and mutually-exciting.
  - Occurrence of one type of events (say type 1) increases the chance of occurrence of both *this* type of events and *other* type of events (say type 2) in the future.
- Numerous applications:
  - Seismology: Modeling earthquake aftershocks
  - Finance: Modeling limit order books
  - Analysis of social network: Modeling influences among individuals

Learning of Multivariate Hawkes Process (MHP)

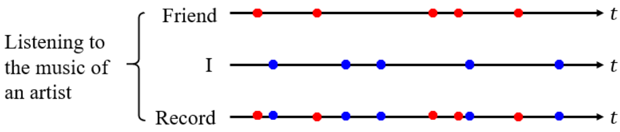
▷ An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \dots, m$.

▷ MHPs are both self-exciting and mutually-exciting.

- Occurrence of one type of events (say type 1) increases the chance of occurrence of both *this* type of events and *other* type of events (say type 2) in the future.

▷ Numerous applications:

- Seismology: Modeling earthquake aftershocks
- Finance: Modeling limit order books
- Analysis of social network: Modeling influences among individuals



Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.
- MHPs are both self-exciting and mutually-exciting.
  - Occurrence of one type of events (say type 1) increases the chance of occurrence of both *this* type of events and *other* type of events (say type 2) in the future.
- Numerous applications:
  - Seismology: Modeling earthquake aftershocks
  - Finance: Modeling limit order books
  - Analysis of social network: Modeling influences among individuals

Learning of Multivariate Hawkes Process (MHP)

- An m-dimensional MHP is a marked temporal point process that consist of m types of events, indexed by $1, \ldots, m$.
- MHPs are both self-exciting and mutually-exciting.
  - Occurrence of one type of events (say type 1) increases the chance of occurrence of both *this* type of events and *other* type of events (say type 2) in the future.
- Numerous applications:
  - Seismology: Modeling earthquake aftershocks
  - Finance: Modeling limit order books
  - Analysis of social network: Modeling influences among individuals

Learning MHPs helps reveal the network influence structure!

Maximum-Likelihood Estimation of MHPs

Maximum-Likelihood Estimation of MHPs

- We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:

Maximum-Likelihood Estimation of MHPs

- We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:
  - $t_i \in [0, t)$ denotes the occurrence time

Maximum-Likelihood Estimation of MHPs

- We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:
  - $t_i \in [0, t)$ denotes the occurrence time
  - $u_i \in [m]$ denotes the event type (or dimension index).

Maximum-Likelihood Estimation of MHPs

▷ We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:
- $t_i \in [0, t)$ denotes the occurrence time
- $u_i \in [m]$ denotes the event type (or dimension index).

▷ The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum\nolimits_{i:t_i<t} a_{u_i,k} \exp(t - t_i), \quad \forall\, t > 0$$

Maximum-Likelihood Estimation of MHPs

▷ We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:

- $t_i \in [0, t)$ denotes the occurrence time
- $u_i \in [m]$ denotes the event type (or dimension index).

▷ The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum_{i:t_i<t} a_{u_i,k} \exp(t - t_i), \quad \forall\, t > 0$$

- $\mu_k \geq 0$ is the the base intensity in dimension k

Maximum-Likelihood Estimation of MHPs

- We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:
  - $t_i \in [0, t)$ denotes the occurrence time
  - $u_i \in [m]$ denotes the event type (or dimension index).

- The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum\nolimits_{i:t_i<t} a_{u_i,k} \exp(t - t_i), \quad \forall\, t > 0$$

  - $\mu_k \geq 0$ is the the base intensity in dimension k
  - $a_{u_i,k} \geq 0$ is the mutual-excitation coefficient between dimensions u_i and k

Maximum-Likelihood Estimation of MHPs

- We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:
  - $t_i \in [0, t)$ denotes the occurrence time
  - $u_i \in [m]$ denotes the event type (or dimension index).

- The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum_{i:t_i<t} a_{u_i,k} \exp(t - t_i), \quad \forall\, t > 0$$

  - $\mu_k \geq 0$ is the the base intensity in dimension k
  - $a_{u_i,k} \geq 0$ is the mutual-excitation coefficient between dimensions u_i and k

- Assume that each type of event has occurred at least once over $[0, t)$.

Maximum-Likelihood Estimation of MHPs

▷ We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:

- $t_i \in [0, t)$ denotes the occurrence time
- $u_i \in [m]$ denotes the event type (or dimension index).

▷ The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum_{i:t_i<t} a_{u_i,k} \exp(t - t_i), \quad \forall\, t > 0$$

- $\mu_k \geq 0$ is the the base intensity in dimension k
- $a_{u_i,k} \geq 0$ is the mutual-excitation coefficient between dimensions u_i and k

▷ Assume that each type of event has occurred at least once over $[0, t)$.

▷ Maximum-likelihood estimation can be done in parallel for each dimension $k \in [m]$:

$$\begin{aligned} \max \quad & \sum_{i\in\mathcal{H}_k} \ln\Big(\mu_k + \sum_{l=1}^m a_{l,k}\, w_{i,l}\Big) - \Big(\mu_k t + \sum_{l=1}^m a_{l,k}\, v_l\Big) \\ \text{s.t.}\ & \mu_k \geq 0,\ a_{l,k} \geq 0, \forall\, l \in [m] \end{aligned} \tag{MHP}$$

Maximum-Likelihood Estimation of MHPs

▷ We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:

- $t_i \in [0, t)$ denotes the occurrence time
- $u_i \in [m]$ denotes the event type (or dimension index).

▷ The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum_{i: t_i < t} a_{u_i, k} \exp(t - t_i), \quad \forall\, t > 0$$

- $\mu_k \geq 0$ is the the base intensity in dimension k
- $a_{u_i, k} \geq 0$ is the mutual-excitation coefficient between dimensions u_i and k

▷ Assume that each type of event has occurred at least once over $[0, t)$.

▷ Maximum-likelihood estimation can be done in parallel for each dimension $k \in [m]$:

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{H}_k} \ln \left(\mu_k + \sum_{l=1}^m a_{l,k}\, w_{i,l}\right) - \left(\mu_k t + \sum_{l=1}^m a_{l,k}\, v_l\right) \\ \text{s.t.}\ & \mu_k \geq 0,\ a_{l,k} \geq 0, \forall\, l \in [m] \end{aligned} \qquad (\texttt{MHP})$$

where $\mathcal{H}_k := \{i \in [n] : t_i < t, u_i = k\}$,

Maximum-Likelihood Estimation of MHPs

- We have observed n events $\mathcal{E} := \{(t_i, u_i)\}_{i=1}^n$ over time interval $[0, t)$:
  - $t_i \in [0, t)$ denotes the occurrence time
  - $u_i \in [m]$ denotes the event type (or dimension index).

- The conditional intensity function in each dimension $k \in [m]$ is given by

$$\lambda_k(t) := \mu_k + \sum_{i:t_i<t} a_{u_i,k} \exp(t - t_i), \quad \forall\, t > 0$$

  - $\mu_k \geq 0$ is the the base intensity in dimension k
  - $a_{u_i,k} \geq 0$ is the mutual-excitation coefficient between dimensions u_i and k

- Assume that each type of event has occurred at least once over $[0, t)$.

- Maximum-likelihood estimation can be done in parallel for each dimension $k \in [m]$:

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{H}_k} \ln\left(\mu_k + \sum_{l=1}^m a_{l,k}\, w_{i,l}\right) - \left(\mu_k t + \sum_{l=1}^m a_{l,k}\, v_l\right) \\ \text{s.t.} \quad & \mu_k \geq 0,\ a_{l,k} \geq 0, \forall\, l \in [m] \end{aligned} \tag{MHP}$$

where $\mathcal{H}_k := \{i \in [n] : t_i < t, u_i = k\}$, and from $\mathcal{E}$, we can compute

$$w_{i,l} \geq 0 \quad \text{and} \quad v_l > 0, \quad \forall\, i \in \mathcal{H}_k, \quad \forall\, l \in [m]$$

Equivalent Formulation of (MHP)

Using standard techniques, we can reformulate (MHP) to the following problem:

$$\min_x \left\{F(x) := -\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \quad x \in \Delta_n \tag{PET}$$

Equivalent Formulation of (MHP)

Using standard techniques, we can reformulate (MHP) to the following problem:

$$\min_x \left\{ F(x) := -\sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \quad x \in \Delta_n \qquad \texttt{(PET)}$$

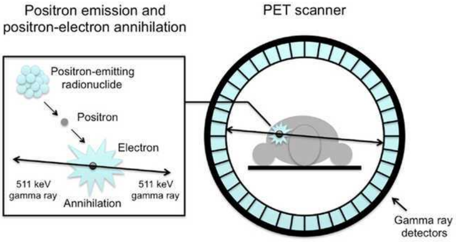
- ▷ Historically, this problem comes from Positron Emission Tomography (PET) in the field of medical imaging.

Equivalent Formulation of (MHP)

Using standard techniques, we can reformulate (MHP) to the following problem:

$$\min_x \left\{F(x) := -\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

▷ Historically, this problem comes from Positron Emission Tomography (PET) in the field of medical imaging.



Equivalent Formulation of (MHP)

Using standard techniques, we can reformulate (MHP) to the following problem:

$$\min_x \left\{ F(x) := -\sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \quad x \in \Delta_n \qquad \texttt{(PET)}$$

- ▷ Historically, this problem comes from Positron Emission Tomography (PET) in the field of medical imaging.
- ▷ For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}^n_+$, $a_j \neq 0$.

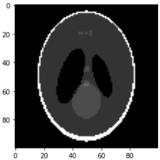
Equivalent Formulation of (MHP)

Using standard techniques, we can reformulate (MHP) to the following problem:

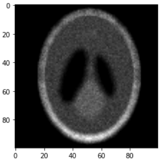
$$\min_x \left\{F(x) := -\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \quad x \in \Delta_n \qquad \texttt{(PET)}$$

- Historically, this problem comes from Positron Emission Tomography (PET) in the field of medical imaging.
- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}_+^n$, $a_j \neq 0$.
- $\Delta_n := \{x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.

Poisson Image Deblurring with TV Regularization

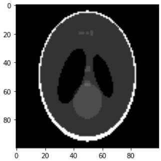


True image X

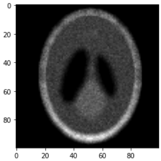


Noisy image Y

Poisson Image Deblurring with TV Regularization



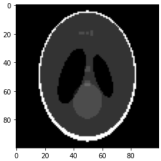
True image X



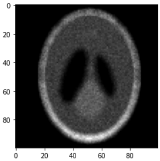
Noisy image Y

▷ Let an $m \times n$ matrix X denote the true representation of an image, such that $0 \leq X_{ij} \leq M$ denotes the pixel level at location (i, j).

Poisson Image Deblurring with TV Regularization



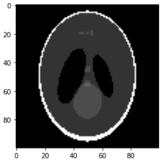
True image X



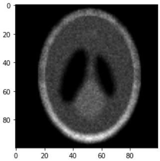
Noisy image Y

- Let an $m \times n$ matrix X denote the true representation of an image, such that $0 \leq X_{ij} \leq M$ denotes the pixel level at location (i, j).
- Let $\mathsf{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ denote the 2D discrete convolutional (linear) operator, which is assumed to be known.

Poisson Image Deblurring with TV Regularization



True image X



Noisy image Y

- Let an $m \times n$ matrix X denote the true representation of an image, such that $0 \leq X_{ij} \leq M$ denotes the pixel level at location (i, j).

- Let $\mathsf{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ denote the 2D discrete convolutional (linear) operator, which is assumed to be known.

- The observed image Y is obtained by first passing X through A, and then contaminated by additive independent (entry-wise) Poisson noise.

Poisson Image Deblurring with TV Regularization

Poisson Image Deblurring with TV Regularization

▷ For convenience, we

- represent the linear operator A in its matrix form $A \in \mathbb{R}^{N\times N}$ $(N := mn)$ and let the l-th row of A be $a_l^\top$ for $l \in [N]$,
- let $x = \mathsf{vec}(X) \in \mathbb{R}^N$ and $X = \mathsf{mat}(x) \in \mathbb{R}^{m\times n}$, and similar for y and Y.

Poisson Image Deblurring with TV Regularization

▷ For convenience, we

- represent the linear operator A in its matrix form $A \in \mathbb{R}^{N\times N}$ $(N := mn)$ and let the l-th row of A be $a_l^\top$ for $l \in [N]$,
- let $x = \mathsf{vec}(X) \in \mathbb{R}^N$ and $X = \mathsf{mat}(x) \in \mathbb{R}^{m\times n}$, and similar for y and Y.

▷ We seek to recover X from Y (equivalently x from y) using maximum-likelihood estimation on the TV-regularized problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^N} \quad & -\textstyle\sum_{l=1}^N y_l \ln(a_l^\top x) + (\sum_{l=1}^N a_l)^\top x + \lambda \mathrm{TV}(x) \\ \text{s.t.} \quad & 0 \le x \le Me \end{aligned} \qquad \texttt{(Deblur)}$$

Poisson Image Deblurring with TV Regularization

- For convenience, we
  - represent the linear operator A in its matrix form $A \in \mathbb{R}^{N\times N}$ ($N := mn$) and let the l-th row of A be $a_l^\top$ for $l \in [N]$,
  - let $x = \mathsf{vec}(X) \in \mathbb{R}^N$ and $X = \mathsf{mat}(x) \in \mathbb{R}^{m\times n}$, and similar for y and Y.
- We seek to recover X from Y (equivalently x from y) using maximum-likelihood estimation on the TV-regularized problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^N} \quad & -\textstyle\sum_{l=1}^N y_l \ln(a_l^\top x) + (\sum_{l=1}^N a_l)^\top x + \lambda \mathrm{TV}(x) \\ \text{s.t.} \quad & 0 \le x \le Me \end{aligned} \qquad \texttt{(Deblur)}$$

- (`Deblur`) has a (standard) total-variation (TV) regularization term to recover a smooth image with sharp edges. The TV term is given by

$$\mathrm{TV}(x) := \textstyle\sum_{i,j} |X_{i,j} - X_{i,j+1}| + \sum_{i,j} |X_{i,j} - X_{i+1,j}| \,.$$

Nesterov's Semidefinite Relaxation of Boolean QP

- The Boolean QP: $q^* := \max_{x\in\{\pm 1\}^n} x^\top Ax$ for some $A \succ 0$.

Nesterov's Semidefinite Relaxation of Boolean QP

- The Boolean QP: $q^* := \max_{x\in\{\pm 1\}^n} x^\top Ax$ for some $A \succ 0$.

- Nesterov [Nes98] showed that the semidefinite relaxation

$$s^* := \min_y \ \langle e, y\rangle \quad \text{s.t.} \quad \mathrm{Diag}(y) \succeq A \tag{SDP}$$

provides a $(2/\pi)$-approximation of the Boolean QP.

Nesterov's Semidefinite Relaxation of Boolean QP

- The Boolean QP: $q^* := \max_{x\in\{\pm1\}^n} x^\top Ax$ for some $A \succ 0$.

- Nesterov [Nes98] showed that the semidefinite relaxation
$$s^* := \min_y \ \langle e, y\rangle \quad \text{s.t.} \quad \mathrm{Diag}(y) \succeq A \tag{SDP}$$
provides a $(2/\pi)$-approximation of the Boolean QP.

- Nesterov [Nes11] later showed that (SDP) above can be equivalently written in the dual form:
$$\begin{aligned} \min_X \quad & F(X) := -2\ln\left(\textstyle\sum_{i=1}^n \langle X, r_i r_i^\top\rangle^{1/2}\right) \\ \text{s.t.} \quad & X \in \mathbb{S}^n_+,\ \langle I_n, X\rangle = 1 \end{aligned} \tag{RBQP}$$
where $A = R^\top R$ (Cholesky factorization) and $R := [r_1 \ \cdots \ r_n]$, and $\mathbb{S}^n_+$ denotes the cone of $n\times n$ real symmetric PSD matrices.

Nesterov's Semidefinite Relaxation of Boolean QP

- The Boolean QP: $q^* := \max_{x\in\{\pm 1\}^n} x^\top Ax$ for some $A \succ 0$.

- Nesterov [Nes98] showed that the semidefinite relaxation

$$s^* := \min_y \ \langle e, y\rangle \quad \text{s. t.} \quad \mathrm{Diag}(y) \succeq A \tag{SDP}$$

  provides a $(2/\pi)$-approximation of the Boolean QP.

- Nesterov [Nes11] later showed that (SDP) above can be equivalently written in the dual form:

$$\begin{aligned} \min_X \quad & F(X) := -2\ln\left(\textstyle\sum_{i=1}^n \langle X, r_i r_i^\top\rangle^{1/2}\right) \\ \text{s. t.} \quad & X \in \mathbb{S}_+^n,\ \langle I_n, X\rangle = 1 \end{aligned} \tag{RBQP}$$

  where $A = R^\top R$ (Cholesky factorization) and $R := [r_1 \ \cdots \ r_n]$, and $\mathbb{S}_+^n$ denotes the cone of $n \times n$ real symmetric PSD matrices.

- Nesterov [Nes11] proposed his "barrier subgradient method" for solving (RBQP) with convergence rate $O(\ln(t)/\sqrt{t})$, but I will present a new gradient method with convergence rate $O(1/t)$!

Two Other Applications

Two Other Applications

▷ *D*-optimal Design (and Minimum-Volume Enclosing Ellipsoid):

Play fundamental roles in computational geometry, statistics and machine learning.

Two Other Applications

- D-optimal Design (and Minimum-Volume Enclosing Ellipsoid):

  Play fundamental roles in computational geometry, statistics and machine learning.

- Quantum State Tomography:
  An Important problem in quantum computing and quantum information theory.

How to Tackle These Problems?

How to Tackle These Problems?

- Since these problems do not exhibit "reasonable" (namely Lipschitz-gradient) behavior, we need to discover *new problem structures* and develop *new methods*.

How to Tackle These Problems?

- Since these problems do not exhibit "reasonable" (namely Lipschitz-gradient) behavior, we need to discover *new problem structures* and develop *new methods*.

- We will introduce two new problem classes, and each class will include most of the applications mentioned previously.

How to Tackle These Problems?

- ▷ Since these problems do not exhibit "reasonable" (namely Lipschitz-gradient) behavior, we need to discover *new problem structures* and develop *new methods*.
- ▷ We will introduce two new problem classes, and each class will include most of the applications mentioned previously.
- ▷ For each problem class, we will develop a new gradient method for tackling the problem:

How to Tackle These Problems?

- Since these problems do not exhibit "reasonable" (namely Lipschitz-gradient) behavior, we need to discover *new problem structures* and develop *new methods*.

- We will introduce two new problem classes, and each class will include most of the applications mentioned previously.

- For each problem class, we will develop a new gradient method for tackling the problem:

  1. A generalized Frank-Wolfe method for convex composite optimization involving a log-homogeneous barrier.

How to Tackle These Problems?

- Since these problems do not exhibit "reasonable" (namely Lipschitz-gradient) behavior, we need to discover *new problem structures* and develop *new methods*.

- We will introduce two new problem classes, and each class will include most of the applications mentioned previously.

- For each problem class, we will develop a new gradient method for tackling the problem:

  1. A generalized Frank-Wolfe method for convex composite optimization involving a log-homogeneous barrier.

  2. An analog of the "Multiplicative Gradient" method for convex optimization involving a log-homogeneous and gradient log-convex function.

Problem of Interest

$$F^* := \min_{x \in \mathbb{R}^n} \; [F(x) := f(\mathsf{A}x) + h(x)] \tag{P-FW}$$

Problem of Interest

$$F^* := \min_{x \in \mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x) \right] \qquad \texttt{(P-FW)}$$

- $\mathsf{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator

Problem of Interest

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \qquad \texttt{(P-FW)}$$

- $\mathsf{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator
- $f : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is a θ-logarithmically-homogeneous self-concordant barrier (θ-LHSCB) for some regular cone $\mathcal{K} \subseteq \mathbb{R}^m$

Problem of Interest

$$F^* := \min_{x \in \mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x) \right] \qquad \texttt{(P-FW)}$$

- $\mathsf{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator
- $f : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is a θ-logarithmically-homogeneous self-concordant barrier (θ-LHSCB) for some regular cone $\mathcal{K} \subseteq \mathbb{R}^m$
- $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed and convex function, with compact domain $\mathcal{X} := \mathsf{dom}\, h$

Problem of Interest

$$F^* := \min_{x \in \mathbb{R}^n} \; [F(x) := f(\mathsf{A}x) + h(x)] \tag{P-FW}$$

- $\mathsf{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator
- $f : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is a θ-logarithmically-homogeneous self-concordant barrier (θ-LHSCB) for some regular cone $\mathcal{K} \subseteq \mathbb{R}^m$
- $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a closed and convex function, with compact domain $\mathcal{X} := \mathsf{dom}\, h$
- All the applications above (except RBQP) fall under (P-FW).

θ-**LHSCB** (logarithmically-homogeneous self-concordant barrier)

θ-**LHSCB** (logarithmically-homogeneous self-concordant barrier)

- Let $\mathcal{K} \subsetneq \mathbb{R}^m$ be a regular cone, i.e., $\mathcal{K}$ is closed, convex, pointed and has nonempty interior.

θ-**LHSCB** (logarithmically-homogeneous self-concordant barrier)

- Let $\mathcal{K} \subsetneq \mathbb{R}^m$ be a regular cone, i.e., $\mathcal{K}$ is closed, convex, pointed and has nonempty interior.
- Two prototypical examples:

θ-LHSCB (logarithmically-homogeneous self-concordant barrier)

- Let $\mathcal{K} \subsetneq \mathbb{R}^m$ be a regular cone, i.e., $\mathcal{K}$ is closed, convex, pointed and has nonempty interior.
- Two prototypical examples:
  - $f(U) = -\ln\det(U)$ for $U \in \mathcal{K} := \mathbb{S}^k_+$ and $\theta = k$,

θ-LHSCB (logarithmically-homogeneous self-concordant barrier)

- Let $\mathcal{K} \subsetneq \mathbb{R}^m$ be a regular cone, i.e., $\mathcal{K}$ is closed, convex, pointed and has nonempty interior.
- Two prototypical examples:
  - $f(U) = -\ln\det(U)$ for $U \in \mathcal{K} := \mathbb{S}^k_+$ and $\theta = k$,
  - $f(u) = -\sum_{j=1}^m w_j \ln(u_j)$ for $u \in \mathcal{K} := \mathbb{R}^m_+$ and $\theta = \sum_{j=1}^m w_j$ where $w_1, \ldots, w_n \geq 1$.

θ-**LHSCB** (logarithmically-homogeneous self-concordant barrier)

- Let $\mathcal{K} \subsetneqq \mathbb{R}^m$ be a regular cone, i.e., $\mathcal{K}$ is closed, convex, pointed and has nonempty interior.

- Two prototypical examples:
  - $f(U) = -\ln\det(U)$ for $U \in \mathcal{K} := \mathbb{S}_+^k$ and $\theta = k$,
  - $f(u) = -\sum_{j=1}^m w_j \ln(u_j)$ for $u \in \mathcal{K} := \mathbb{R}_+^m$ and $\theta = \sum_{j=1}^m w_j$ where $w_1, \ldots, w_n \geq 1$.

- f is a θ-LHSCB on $\mathcal{K}$ with *complexity parameter* $\theta \geq 1$ if f is three-times differentiable and strictly convex on $\mathsf{int}\,\mathcal{K}$, and satisfies

θ-LHSCB (logarithmically-homogeneous self-concordant barrier)

▷ Let $\mathcal{K} \subsetneq \mathbb{R}^m$ be a regular cone, i.e., $\mathcal{K}$ is closed, convex, pointed and has nonempty interior.

▷ Two prototypical examples:

- $f(U) = -\ln\det(U)$ for $U \in \mathcal{K} := \mathbb{S}_+^k$ and $\theta = k$,
- $f(u) = -\sum_{j=1}^m w_j \ln(u_j)$ for $u \in \mathcal{K} := \mathbb{R}_+^m$ and $\theta = \sum_{j=1}^m w_j$ where $w_1, \ldots, w_n \ge 1$.

▷ f is a θ-LHSCB on $\mathcal{K}$ with *complexity parameter* $\theta \ge 1$ if f is three-times differentiable and strictly convex on $\mathsf{int}\,\mathcal{K}$, and satisfies

1. $\left|D^3 f(u)[w,w,w]\right| \le 2\|w\|_u^3 \quad \forall\, u \in \mathsf{int}\,\mathcal{K}, \forall\, w \in \mathbb{R}^m$,
2. $f(u_k) \to \infty$ for any $\{u_k\}_{k\ge 1} \subseteq \mathsf{int}\,\mathcal{K}$ such that $u_k \to u \in \mathsf{bd}\,\mathcal{K}$,
3. $f(tu) = f(u) - \theta\ln(t) \quad \forall\, u \in \mathsf{int}\,\mathcal{K}, \forall\, t > 0$.

where $\|w\|_u := \langle \nabla^2 f(u) w, w\rangle^{1/2}$ denotes the local norm of w at $u \in \mathsf{int}\,\mathcal{K}$.

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \; [F(x) := f(\mathsf{A}x) + h(x)] \tag{P}$$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \tag{P}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \qquad \texttt{(P)}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$
- **Repeat** (until some convergence criterion is met)

$$v^k \in \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + h(x) \qquad \text{(“Linear” subproblem)}$$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \; [F(x) := f(\mathsf{A}x) + h(x)] \tag{P}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$
- **Repeat** (until some convergence criterion is met)

$$v^k \in \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + h(x) \qquad \text{(“Linear” subproblem)}$$

$$G_k := \langle \nabla f(\mathsf{A}x^k), \mathsf{A}(x^k - v^k)\rangle + h(x^k) - h(v^k) \qquad \text{(FW-Gap)}$$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \tag{P}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$
- **Repeat** (until some convergence criterion is met)

$$v^k \in \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + h(x) \qquad \text{(“Linear” subproblem)}$$

$$G_k := \langle \nabla f(\mathsf{A}x^k), \mathsf{A}(x^k - v^k)\rangle + h(x^k) - h(v^k) \qquad \text{(FW-Gap)}$$

$$D_k := \|\mathsf{A}(v^k - x^k)\|_{\mathsf{A}x^k} \qquad \text{(Local Distance)}$$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \; [F(x) := f(\mathsf{A}x) + h(x)] \tag{P}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$
- **Repeat** (until some convergence criterion is met)

$$v^k \in \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + h(x) \qquad \text{(“Linear” subproblem)}$$

$$G_k := \langle \nabla f(\mathsf{A}x^k), \mathsf{A}(x^k - v^k)\rangle + h(x^k) - h(v^k) \qquad \text{(FW-Gap)}$$

$$D_k := \|\mathsf{A}(v^k - x^k)\|_{\mathsf{A}x^k} \qquad \text{(Local Distance)}$$

$$\alpha_k := \min\left\{\frac{G_k}{D_k(G_k + D_k)}, 1\right\} \qquad \text{(Stepsize)}$$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \tag{P}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$
- **Repeat** (until some convergence criterion is met)

$$v^k \in \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + h(x) \qquad \text{(“Linear” subproblem)}$$

$$G_k := \langle \nabla f(\mathsf{A}x^k), \mathsf{A}(x^k - v^k)\rangle + h(x^k) - h(v^k) \qquad \text{(FW-Gap)}$$

$$D_k := \|\mathsf{A}(v^k - x^k)\|_{\mathsf{A}x^k} \qquad \text{(Local Distance)}$$

$$\alpha_k := \min\left\{\frac{G_k}{D_k(G_k + D_k)}, 1\right\} \qquad \text{(Stepsize)}$$

$$x^{k+1} := x^k + \alpha_k(v^k - x^k) \qquad \text{(Update)}$$

Our Method: (generalized) Frank-Wolfe (gFW-LHSCB)

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \qquad \texttt{(P)}$$

- **Initialize**: $x^0 \in \mathsf{dom}\, F$, $k := 0$
- **Repeat** (until some convergence criterion is met)

$$v^k \in \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + h(x) \qquad \text{(“Linear” subproblem)}$$

$$G_k := \langle \nabla f(\mathsf{A}x^k), \mathsf{A}(x^k - v^k)\rangle + h(x^k) - h(v^k) \qquad \text{(FW-Gap)}$$

$$D_k := \|\mathsf{A}(v^k - x^k)\|_{\mathsf{A}x^k} \qquad \text{(Local Distance)}$$

$$\alpha_k := \min\left\{\frac{G_k}{D_k(G_k + D_k)}, 1\right\} \qquad \text{(Stepsize)}$$

$$x^{k+1} := x^k + \alpha_k(v^k - x^k) \qquad \text{(Update)}$$

$$k := k + 1$$

Remarks on gFW-LHSCB

Remarks on gFW-LHSCB

- For most applications (including all of the applications mentioned previously), D_k in (Local Distance) can be computed in $O(n)$ time.

Remarks on gFW-LHSCB

- For most applications (including all of the applications mentioned previously), D_k in (Local Distance) can be computed in $O(n)$ time.

- The FW-gap G_k provides an effective stopping criterion:
$$G_k \geq [\delta_k := F(x^k) - F^*], \quad \text{for all } k \geq 0.$$

Remarks on gFW-LHSCB

- For most applications (including all of the applications mentioned previously), D_k in (Local Distance) can be computed in $O(n)$ time.

- The FW-gap G_k provides an effective stopping criterion:
$$G_k \geq [\delta_k := F(x^k) - F^*], \quad \text{for all } k \geq 0.$$

- For some applications (e.g., `PET` and `D-Optimal Design`), the step-size can also be efficiently computed via exact line-search.

Remarks on gFW-LHSCB

- For most applications (including all of the applications mentioned previously), D_k in (Local Distance) can be computed in $O(n)$ time.
- The FW-gap G_k provides an effective stopping criterion:
$$G_k \geq [\delta_k := F(x^k) - F^*], \quad \text{for all } k \geq 0.$$
- For some applications (e.g., `PET` and `D-Optimal Design`), the step-size can also be efficiently computed via exact line-search.
- Our algorithm does not use the special properties of the barrier or the logarithmic homogeneity of f. However, these properties are critical in deriving the computational guarantees.

Computational Guarantees

$$F^* := \min_{x \in \mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x) \right] \qquad \texttt{(P-FW)}$$

Computational Guarantees

$$F^* := \min_{x \in \mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x) \right] \qquad \texttt{(P-FW)}$$

▷ Define $R_h := \max_{x,y \in \mathsf{dom}\, h} \; |h(x) - h(y)|$ (the variation of h on its domain)

Computational Guarantees

$$F^* := \min_{x \in \mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x) \right] \qquad \texttt{(P-FW)}$$

- Define $R_h := \max_{x,y \in \mathsf{dom}\, h} |h(x) - h(y)|$ (the variation of h on its domain)
- Recall that δ_0 is the initial optimality gap

Computational Guarantees

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \qquad \texttt{(P-FW)}$$

- Define $R_h := \max_{x,y\in \mathsf{dom}\, h} |h(x) - h(y)|$ (the variation of h on its domain)
- Recall that δ_0 is the initial optimality gap

Theorem:

Computational Guarantees

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \qquad \texttt{(P-FW)}$$

- Define $R_h := \max_{x,y\in \mathsf{dom}\, h} \ |h(x) - h(y)|$ (the variation of h on its domain)
- Recall that δ_0 is the initial optimality gap

Theorem:

- (Iteration complexity for ε-optimality gap) Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$. Then:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil .$$

Computational Guarantees

$$F^* := \min_{x\in\mathbb{R}^n} \left[F(x) := f(\mathsf{A}x) + h(x)\right] \qquad \texttt{(P-FW)}$$

- Define $R_h := \max_{x,y\in \mathsf{dom}\, h} |h(x) - h(y)|$ (the variation of h on its domain)
- Recall that δ_0 is the initial optimality gap

Theorem:

- (Iteration complexity for ε-optimality gap) Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$. Then:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil .$$

- (Iteration complexity for ε-FW gap) Let $\mathrm{FWGAP}_\varepsilon$ be the number of iterations required by gFW-LHSCB to obtain $G_k \le \varepsilon$. Then:

$$\mathrm{FWGAP}_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{24(\theta + R_h)^2}{\varepsilon} \right\rceil .$$

Remarks on the Computational Guarantees

Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil$$

Remarks on the Computational Guarantees

Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil$$

Our computational guarantees only depend on three (natural) quantities:

Remarks on the Computational Guarantees

Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil$$

Our computational guarantees only depend on three (natural) quantities:

- the initial optimality gap δ_0,

Remarks on the Computational Guarantees

Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h) \ln(10.6\delta_0) \rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil$$

Our computational guarantees only depend on three (natural) quantities:

- the initial optimality gap δ_0,
- the complexity parameter θ of the barrier f,

Remarks on the Computational Guarantees

Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h) \ln(10.6\delta_0) \rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil$$

Our computational guarantees only depend on three (natural) quantities:

- the initial optimality gap δ_0,
- the complexity parameter θ of the barrier f,
- the variation of h on its domain $\mathsf{dom}\, h$ ($= 0$ if $h = \iota_{\mathcal{X}}$).

Remarks on the Computational Guarantees

Let K_ε be the number of iterations for gFW-LHSCB to obtain $\delta_k \le \varepsilon$:

$$K_\varepsilon \le \lceil 5.3(\delta_0 + \theta + R_h)\ln(10.6\delta_0)\rceil + \left\lceil \frac{12(\theta + R_h)^2}{\varepsilon} \right\rceil$$

Our computational guarantees only depend on three (natural) quantities:

- the initial optimality gap δ_0,
- the complexity parameter θ of the barrier f,
- the variation of h on its domain $\mathsf{dom}\, h$ ($= 0$ if $h = \iota_{\mathcal{X}}$).

For many applications, all of the three quantities can be easily estimated, and hence the computational guarantees are known before running the algorithm.

Numerical Experiments on Poisson Image Deblurring with TV Regularization

$$\min_{x\in\mathbb{R}^N} \quad \underbrace{-\sum_{l=1}^N y_l \ln(a_l^\top x)}_{=f(\mathsf{A}x)} + \underbrace{\langle \sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)}_{=h(x)} \qquad \texttt{(Deblur)}$$

$$\text{s.t.} \quad 0 \le x \le Me$$

Numerical Experiments on Poisson Image Deblurring with TV Regularization

$$\min_{x\in\mathbb{R}^N} \quad \underbrace{-\sum_{l=1}^N y_l \ln(a_l^\top x)}_{=f(\mathsf{A}x)} + \underbrace{\langle \sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)}_{=h(x)} \qquad \texttt{(Deblur)}$$
$$\text{s.t.} \quad 0 \le x \le Me$$

▷ Since $\mathrm{TV}(\cdot)$ is piece-wise linear convex, and the sub-problem

$$v^k \in \arg\min_{0\le x\le Me} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + \langle \sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)$$

can be formulated as a relatively simple LP and solved easily using a standard LP solver such as Gurobi.

Numerical Experiments on Poisson Image Deblurring with TV Regularization

$$\min_{x\in\mathbb{R}^N} \quad \underbrace{-\textstyle\sum_{l=1}^N y_l \ln(a_l^\top x)}_{=f(\mathsf{A}x)} + \underbrace{\langle \textstyle\sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)}_{=h(x)} \qquad \texttt{(Deblur)}$$
$$\text{s.t.} \quad 0 \le x \le Me$$

▷ Since $\mathrm{TV}(\cdot)$ is piece-wise linear convex, and the sub-problem

$$v^k \in \arg\min_{0\le x\le Me} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + \langle \textstyle\sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)$$

can be formulated as a relatively simple LP and solved easily using a standard LP solver such as Gurobi.

▷ Very few principled first-order methods have been proposed to solve (Deblur):

Numerical Experiments on Poisson Image Deblurring with TV Regularization

$$\min_{x\in\mathbb{R}^N} \quad \underbrace{-\sum_{l=1}^N y_l \ln(a_l^\top x)}_{=f(\mathsf{A}x)} + \underbrace{\langle \sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)}_{=h(x)} \qquad \text{(Deblur)}$$

$$\text{s.t.} \quad 0 \le x \le Me$$

- Since $\mathrm{TV}(\cdot)$ is piece-wise linear convex, and the sub-problem

$$v^k \in \arg\min_{0\le x\le Me} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + \langle \sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)$$

  can be formulated as a relatively simple LP and solved easily using a standard LP solver such as Gurobi.

- Very few principled first-order methods have been proposed to solve (Deblur):
  - The function $f : u \mapsto -\sum_{l=1}^N y_l \ln(u_l)$ is neither Lipschitz nor L-smooth,

Numerical Experiments on Poisson Image Deblurring with TV Regularization

$$\min_{x\in\mathbb{R}^N} \quad \underbrace{-\textstyle\sum_{l=1}^N y_l \ln(a_l^\top x)}_{=f(\mathsf{A}x)} + \underbrace{\langle \textstyle\sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)}_{=h(x)} \qquad \texttt{(Deblur)}$$
$$\text{s.t.} \quad 0 \le x \le Me$$

▷ Since $\mathrm{TV}(\cdot)$ is piece-wise linear convex, and the sub-problem

$$v^k \in \arg\min_{0\le x\le Me} \langle \nabla f(\mathsf{A}x^k), \mathsf{A}x\rangle + \langle \textstyle\sum_{l=1}^N a_l, x\rangle + \lambda \mathrm{TV}(x)$$

can be formulated as a relatively simple LP and solved easily using a standard LP solver such as Gurobi.

▷ Very few principled first-order methods have been proposed to solve (**Deblur**):

- The function $f : u \mapsto -\sum_{l=1}^N y_l \ln(u_l)$ is neither Lipschitz nor L-smooth,
- The (Bregman) proximal sub-problem involving $\mathrm{TV}(\cdot)$ and the "box" constraint may not be efficiently solved [HJN15].

Implementation Details/Issues

Implementation Details/Issues

- We evaluate the numerical performance of our FW method gFW-LHSCB (with adaptive stepsize) which we call `FW-Adapt`.

Implementation Details/Issues

- We evaluate the numerical performance of our FW method gFW-LHSCB (with adaptive stepsize) which we call `FW-Adapt`.
- It turns out that an exact line-search step-size for gFW-LHSCB can be computed for this particular problem, which we call `FW-Exact`.

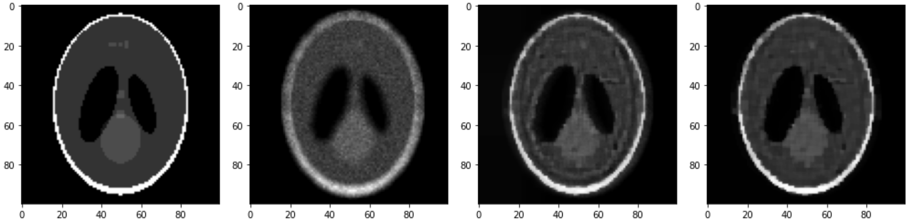
Implementation Details/Issues

- We evaluate the numerical performance of our FW method gFW-LHSCB (with adaptive stepsize) which we call **`FW-Adapt`**.
- It turns out that an exact line-search step-size for gFW-LHSCB can be computed for this particular problem, which we call **`FW-Exact`**.
- We tested **`FW-Adapt`** and **`FW-Exact`** on the Shepp-Logan phantom image of size 100×100 (hence $N = 10,000$).

Implementation Details/Issues

- We evaluate the numerical performance of our FW method gFW-LHSCB (with adaptive stepsize) which we call `FW-Adapt`.
- It turns out that an exact line-search step-size for gFW-LHSCB can be computed for this particular problem, which we call `FW-Exact`.
- We tested `FW-Adapt` and `FW-Exact` on the Shepp-Logan phantom image of size 100×100 (hence $N = 10,000$).
- We chose the starting point $x^0 = \mathsf{vec}(Y)$ (the vectorized noisy image), and we set $\lambda = 0.01$.

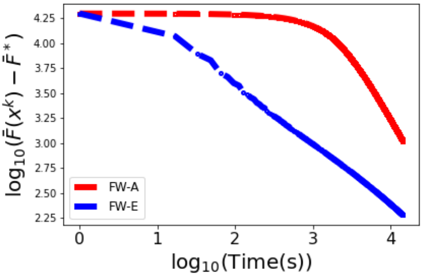
Results: Recovered Images



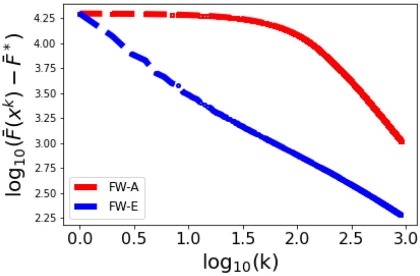
(a) True image X (b) Noisy image Y (c) `FW-Adapt` (d) `FW-Exact`

Figure 1: True, noisy and recovered Shepp-Logan phantom images.

Results: Optimality Gaps versus Time and Iterations



(a) Optimality gap versus time (in seconds)



(b) Optimality gap versus iterations

Figure 2: Comparison of optimality gaps of `FW-Adapt` (`FW-A`) and `FW-Exact` (`FW-E`) for image recovery of the Shepp-Logan phantom image.

Motivating Example: Positron Emission Tomography

$$\max_x \left\{ F(x) := \sum_{j=1}^{m} p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

Motivating Example: Positron Emission Tomography

$$\max_x \left\{F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}_+^n$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.

Motivating Example: Positron Emission Tomography

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s. t.} \quad x \in \Delta_n \qquad \texttt{(PET)}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}^n_+$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.
- $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.

Motivating Example: Positron Emission Tomography

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}^n_+$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.
- $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.
- Multiplicative gradient method: $x^0 \in \mathsf{ri}\,\Delta_n$

$$x^{t+1} = x^t \circ \nabla F(x^t) \quad \equiv \quad x_i^{t+1} := x_i^t \nabla_i F(x^t), \quad \forall\, i \in [n].$$

Motivating Example: Positron Emission Tomography

$$\max_x \left\{F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}^n_+$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.
- $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.
- Multiplicative gradient method: $x^0 \in \mathsf{ri}\,\Delta_n$

$$x^{t+1} = x^t \circ \nabla F(x^t) \quad \equiv \quad x_i^{t+1} := x_i^t \nabla_i F(x^t), \quad \forall\, i \in [n]. \tag{MG}$$

- A reviewer brought this method to my attention during the revision of my Frank-Wolfe paper.

Motivating Example: Positron Emission Tomography

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}^n_+$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.
- $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.
- Multiplicative gradient method: $x^0 \in \mathsf{ri}\, \Delta_n$

$$x^{t+1} = x^t \circ \nabla F(x^t) \quad \equiv \quad x_i^{t+1} := x_i^t \nabla_i F(x^t), \quad \forall\, i \in [n]. \tag{MG}$$

- A reviewer brought this method to my attention during the revision of my Frank-Wolfe paper.
- I studied it for a while, and realized that (MG) does not fall under any "well-known" optimization frameworks:

Motivating Example: Positron Emission Tomography

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}^n_+$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.
- $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.
- Multiplicative gradient method: $x^0 \in \mathsf{ri}\,\Delta_n$

$$x^{t+1} = x^t \circ \nabla F(x^t) \quad \equiv \quad x_i^{t+1} := x_i^t \nabla_i F(x^t), \quad \forall\, i \in [n]. \tag{MG}$$

- A reviewer brought this method to my attention during the revision of my Frank-Wolfe paper.
- I studied it for a while, and realized that (MG) does not fall under any "well-known" optimization frameworks:
  - Not Newton-type method

Motivating Example: Positron Emission Tomography

$$\max_x \left\{F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

- For all $j \in [m]$, let $p_j > 0$, $a_j \in \mathbb{R}_+^n$, $a_j \neq 0$ and $\sum_{j=1}^m p_j = 1$.
- $\Delta_n := \{x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1\}$ is the unit simplex in $\mathbb{R}^n$.
- Multiplicative gradient method: $x^0 \in \mathsf{ri}\,\Delta_n$

$$x^{t+1} = x^t \circ \nabla F(x^t) \quad \equiv \quad x_i^{t+1} := x_i^t \nabla_i F(x^t), \quad \forall\, i \in [n]. \tag{MG}$$

- A reviewer brought this method to my attention during the revision of my Frank-Wolfe paper.
- I studied it for a while, and realized that (MG) does not fall under any "well-known" optimization frameworks:
  - Not Newton-type method
  - Not entropic mirror descent

The Mystery of MG

$$\max_x \left\{F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \quad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

The Mystery of MG

$$\max_x \left\{F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

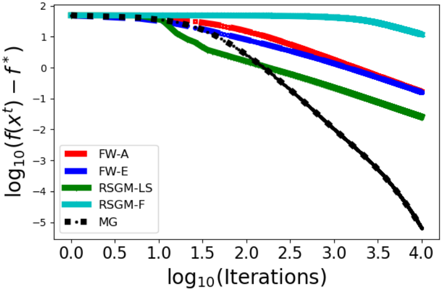
- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.

The Mystery of MG

$$\max_x \left\{F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- Surprisingly good numerical performance: $x^0 = (1/n)e$



FW-A & FW-E [Dvu20; ZF22]: Generalized FW methods for LHB (with adaptive stepsize and exact line search)

RSGM-F & RSGM-LS [BBT17; LFN18]: Relatively smooth gradient method (with fixed stepsize and backtracking line search)

The Mystery of MG

$$\max_x \left\{ F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- Surprisingly good numerical performance
- This made me curious and dig into this method ...

The Mystery of MG

$$\max_x \left\{ F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- Surprisingly good numerical performance
- This made me curious and dig into this method ...

1970s (MG) was proposed by information theorists [Ari72]

The Mystery of MG

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- Surprisingly good numerical performance
- This made me curious and dig into this method ...

1970s	(MG) was proposed by information theorists [Ari72]
1980s	Iterates have a unique limit point that is optimal to (PET) [Csi84]

The Mystery of MG

$$\max_x \left\{ F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- Surprisingly good numerical performance
- This made me curious and dig into this method ...

1970s	(MG) was proposed by information theorists [Ari72]
1980s	Iterates have a unique limit point that is optimal to (PET) [Csi84]
1990s – 2021	(MG) seems to be forgotten — but what's the convergence rate?

The Mystery of MG

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \quad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- ▷ The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- ▷ Surprisingly good numerical performance
- ▷ This made me curious and dig into this method ...

1970s	(MG) was proposed by information theorists [Ari72]
1980s	Iterates have a unique limit point that is optimal to (PET) [Csi84]
1990s – 2021	(MG) seems to be forgotten — but what's the convergence rate?
2021	I showed that (MG) has convergence rate $O(\ln(n)/t)$ [Zha22]

The Mystery of MG

$$\max_x \left\{ F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

- The MG method is deceptively simple, since it doesn't involve choosing step-sizes and solving sub-problems.
- Surprisingly good numerical performance
- This made me curious and dig into this method ...

1970s	(MG) was proposed by information theorists [Ari72]
1980s	Iterates have a unique limit point that is optimal to (PET) [Csi84]
1990s – 2021	(MG) seems to be forgotten — but what's the convergence rate?
2021	I showed that (MG) has convergence rate $O(\ln(n)/t)$ [Zha22]

- More interestingly, there's no constant hidden in $O(\cdot)$:

$$F^* - F(x^t) \leq \ln(n)/t, \quad \forall t \geq 1$$

Some Deeper Questions

$$\max_x \left\{F(x) := \sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \quad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

$$\boxed{F^* - F(x^t) \le \ln(n)/t, \quad \forall t \ge 1} \tag{Rate}$$

▷ Why does (MG) work for PET?

Some Deeper Questions

$$\max_x \left\{F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

$$\boxed{F^* - F(x^t) \le \ln(n)/t, \quad \forall t \ge 1} \tag{Rate}$$

- ▷ Why does (MG) work for PET?
- ▷ What are the essential structures of the problem the drive the success of (MG)? Is there a general problem class that (MG) works well?

Some Deeper Questions

$$\max_x \left\{F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s. t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

$$\boxed{F^* - F(x^t) \le \ln(n)/t, \quad \forall t \ge 1} \tag{Rate}$$

- Why does (MG) work for PET?
- What are the essential structures of the problem the drive the success of (MG)? Is there a general problem class that (MG) works well?
- Can we develop a general method in the same spirit of (MG) that works for this general problem class?

Some Deeper Questions

$$\max_x \left\{F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x)\right\} \qquad \text{s.t.} \qquad x \in \Delta_n \tag{PET}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \tag{MG}$$

$$\boxed{F^* - F(x^t) \le \ln(n)/t, \quad \forall t \ge 1} \tag{Rate}$$

- Why does (MG) work for PET?
- What are the essential structures of the problem the drive the success of (MG)? Is there a general problem class that (MG) works well?
- Can we develop a general method in the same spirit of (MG) that works for this general problem class?
- Finally, what is the interaction between the convergence rate of (MG) and the problem structure?

Some Deeper Questions

$$\max_x \left\{ F(x) := \textstyle\sum_{j=1}^m p_j \ln(a_j^\top x) \right\} \qquad \text{s.t.} \qquad x \in \Delta_n \qquad \texttt{(PET)}$$

$$\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \qquad x^{t+1} = x^t \circ \nabla F(x^t)} \qquad \texttt{(MG)}$$

$$\boxed{F^* - F(x^t) \le \ln(n)/t, \quad \forall t \ge 1} \qquad \texttt{(Rate)}$$

- Why does (MG) work for PET?
- What are the essential structures of the problem the drive the success of (MG)? Is there a general problem class that (MG) works well?
- Can we develop a general method in the same spirit of (MG) that works for this general problem class?
- Finally, what is the interaction between the convergence rate of (MG) and the problem structure?

These questions kept me working for half a year, and I eventually came up with some satisfactory answers to these questions ...

My Answers

- I identified a broad problem class and develop an analog of the MG (AMG) method, that converges at rate $O(1/t)$.

My Answers

- I identified a broad problem class and develop an analog of the MG (AMG) method, that converges at rate $O(1/t)$.

- Roughly speaking, this problem class minimizes a *log-homogeneous* and *gradient log-convex* function over a "slice" of symmetric cone.

My Answers

- ▷ I identified a broad problem class and develop an analog of the MG (AMG) method, that converges at rate $O(1/t)$.
- ▷ Roughly speaking, this problem class minimizes a *log-homogeneous* and *gradient log-convex* function over a "slice" of symmetric cone.
  - Typical symmetric cones include nonnegative orthant, second-order cone, positive semidefinite cone and their (finite) Cartesian product.

My Answers

- I identified a broad problem class and develop an analog of the MG (AMG) method, that converges at rate $O(1/t)$.
- Roughly speaking, this problem class minimizes a *log-homogeneous* and *gradient log-convex* function over a "slice" of symmetric cone.
  - Typical symmetric cones include nonnegative orthant, second-order cone, positive semidefinite cone and their (finite) Cartesian product.
- The development and analysis of the AMG method are based on the framework of Euclidean Jordan algebra.

My Answers

- I identified a broad problem class and develop an analog of the MG (AMG) method, that converges at rate $O(1/t)$.
- Roughly speaking, this problem class minimizes a *log-homogeneous* and *gradient log-convex* function over a "slice" of symmetric cone.
  - Typical symmetric cones include nonnegative orthant, second-order cone, positive semidefinite cone and their (finite) Cartesian product.
- The development and analysis of the AMG method are based on the framework of Euclidean Jordan algebra.
- I will only show the specific form of AMG method on the following applications:
  - Nesterov's Semidefinite relaxation of Boolean QP
  - D-optimal design
  - Quantum state tomography

My Answers

- I identified a broad problem class and develop an analog of the MG (AMG) method, that converges at rate $O(1/t)$.
- Roughly speaking, this problem class minimizes a *log-homogeneous* and *gradient log-convex* function over a "slice" of symmetric cone.
  - Typical symmetric cones include nonnegative orthant, second-order cone, positive semidefinite cone and their (finite) Cartesian product.
- The development and analysis of the AMG method are based on the framework of Euclidean Jordan algebra.
- I will only show the specific form of AMG method on the following applications:
  - Nesterov's Semidefinite relaxation of Boolean QP
  - D-optimal design
  - Quantum state tomography
- In all of these applications, the objective functions involve "$\ln(\cdot)$", and hence do not have Lipschitz-gradient on the feasible sets.

D-optimal Design

$$\min_x F(x) := -\ln\det\left(\textstyle\sum_{i=1}^n x_i a_i a_i^\top\right) \quad \text{s.t.} \quad x \in \Delta_n \tag{D-OPT}$$

- Problem data: n points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^m$ that are symmetric about the origin and linearly span $\mathbb{R}^m$.

D-optimal Design

$$\min_x F(x) := -\ln\det\left(\sum_{i=1}^n x_i a_i a_i^\top\right) \quad \text{s.t.} \quad x \in \Delta_n \qquad \texttt{(D-OPT)}$$

- Problem data: n points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^m$ that are symmetric about the origin and linearly span $\mathbb{R}^m$.
- Arises as the dual of the minimum-volume enclosing ellipsoid (MVEE) problem.

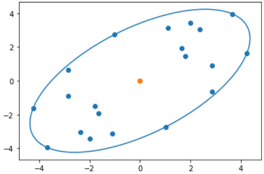
D-optimal Design

$$\min_x F(x) := -\ln\det\left(\sum_{i=1}^n x_i a_i a_i^\top\right) \quad \text{s.t.} \quad x \in \Delta_n \qquad \texttt{(D-OPT)}$$

▷ Problem data: n points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^m$ that are symmetric about the origin and linearly span $\mathbb{R}^m$.

▷ Arises as the dual of the minimum-volume enclosing ellipsoid (MVEE) problem.

Given $\{a_i\}_{i=1}^n$, we wish to find a minimum-volume ellipsoid that encloses $\{a_i\}_{i=1}^n$.



D-optimal Design

$$\min_x \; F(x) := -\ln\det\left(\textstyle\sum_{i=1}^n x_i a_i a_i^\top\right) \quad \text{s.t.} \quad x \in \Delta_n \tag{D-OPT}$$

- Problem data: n points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^m$ that are symmetric about the origin and linearly span $\mathbb{R}^m$.
- Arises as the dual of the minimum-volume enclosing ellipsoid (MVEE) problem.
- (D-OPT) and (MVEE) plays fundamental roles in computational geometry, statistics and machine learning.

D-optimal Design

$$\min_x \; F(x) := -\ln\det\left(\textstyle\sum_{i=1}^n x_i a_i a_i^\top\right) \quad \text{s.t.} \quad x \in \Delta_n \qquad \texttt{(D-OPT)}$$

- Problem data: n points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^m$ that are symmetric about the origin and linearly span $\mathbb{R}^m$.
- Arises as the dual of the minimum-volume enclosing ellipsoid (MVEE) problem.
- (D-OPT) and (MVEE) plays fundamental roles in computational geometry, statistics and machine learning.
- AMG method: $\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \quad x^{t+1} = x^t \circ \nabla F(x^t)}$

D-optimal Design

$$\min_x F(x) := -\ln\det\left(\sum_{i=1}^n x_i a_i a_i^\top\right) \quad \text{s.t.} \quad x \in \Delta_n \tag{D-OPT}$$

- Problem data: n points $\{a_i\}_{i=1}^n$ in $\mathbb{R}^m$ that are symmetric about the origin and linearly span $\mathbb{R}^m$.
- Arises as the dual of the minimum-volume enclosing ellipsoid (MVEE) problem.
- (D-OPT) and (MVEE) plays fundamental roles in computational geometry, statistics and machine learning.
- AMG method: $\boxed{x^0 \in \mathsf{ri}\,\Delta_n, \quad x^{t+1} = x^t \circ \nabla F(x^t)}$
- Computational guarantee:

$$F^* - F(\bar{x}^t) \le \ln(n)/t, \quad \forall t \ge 1 \qquad \left[\bar{x}^t := (1/t)\sum_{i=0}^{t-1} x^i\right]$$

Quantum State Tomography (QST)

$$\begin{aligned} \max_X \; & F(X) := m^{-1} \textstyle\sum_{j=1}^{q} n_j \ln(\langle X, a_j a_j^H \rangle) \\ \text{s.t.} \quad & X \in \mathbb{H}_+^n, \;\; \operatorname{tr}(X) = \langle I_n, X \rangle = 1 \end{aligned} \tag{QST}$$

Quantum State Tomography (QST)

$$
\begin{aligned}
&\max_X \; F(X) := m^{-1} \textstyle\sum_{j=1}^{q} n_j \ln(\langle X, a_j a_j^H \rangle) \\
&\text{s.t.} \quad X \in \mathbb{H}_+^n, \;\; \mathrm{tr}(X) = \langle I_n, X \rangle = 1
\end{aligned}
\tag{QST}
$$

- In quantum physics, this problem aims to reconstruct the state of a quantum system using the measured output of particles [Hra04].

Quantum State Tomography (QST)

$$\begin{aligned} \max_X \ & F(X) := m^{-1} \textstyle\sum_{j=1}^q n_j \ln(\langle X, a_j a_j^H \rangle) \\ \text{s.t.} \ & X \in \mathbb{H}_+^n, \ \ \mathrm{tr}(X) = \langle I_n, X \rangle = 1 \end{aligned} \tag{QST}$$

- In quantum physics, this problem aims to reconstruct the state of a quantum system using the measured output of particles [Hra04].
- $a_1, \ldots, a_q \in \mathbb{C}^n$, $\sum_{j=1}^q a_j a_j^H = I_n$ and $\sum_{j=1}^q n_j = m$.

Quantum State Tomography (QST)

$$\begin{aligned} \max_X \; & F(X) := m^{-1} \textstyle\sum_{j=1}^q n_j \ln(\langle X, a_j a_j^H \rangle) \\ \text{s.t.} \; & X \in \mathbb{H}_+^n, \;\; \mathrm{tr}(X) = \langle I_n, X \rangle = 1 \end{aligned} \tag{QST}$$

▷ In quantum physics, this problem aims to reconstruct the state of a quantum system using the measured output of particles [Hra04].

▷ $a_1, \ldots, a_q \in \mathbb{C}^n$, $\sum_{j=1}^q a_j a_j^H = I_n$ and $\sum_{j=1}^q n_j = m$.

▷ $\mathbb{H}_+^n$ denotes the cone of $n \times n$ complex Hermitian PSD matrices.

Quantum State Tomography (QST)

$$\begin{aligned} \max_X \; & F(X) := m^{-1} \textstyle\sum_{j=1}^q n_j \ln(\langle X, a_j a_j^H \rangle) \\ \text{s.t.} \; & X \in \mathbb{H}_+^n, \;\; \mathrm{tr}(X) = \langle I_n, X \rangle = 1 \end{aligned} \tag{QST}$$

- In quantum physics, this problem aims to reconstruct the state of a quantum system using the measured output of particles [Hra04].
- $a_1, \ldots, a_q \in \mathbb{C}^n$, $\sum_{j=1}^q a_j a_j^H = I_n$ and $\sum_{j=1}^q n_j = m$.
- $\mathbb{H}_+^n$ denotes the cone of $n \times n$ complex Hermitian PSD matrices.
- AMG method: $X^0 \succ 0$, $\mathrm{tr}(X^0) = 1$,

$$\boxed{\begin{aligned} \hat{X}^{t+1} &= \exp\{\ln(X^t) + \ln(\nabla F(X^t))\} \\ X^{t+1} &= \hat{X}^{t+1} / \mathrm{tr}(\hat{X}^{t+1}) \end{aligned}}$$

(For any $X = \sum_{i=1}^n \lambda_i u_i u_i^H \succ 0$, $\ln(X) := \ln(\lambda_i) u_i u_i^H$.)

Quantum State Tomography (QST)

$$\begin{aligned} \max_X \ & F(X) := m^{-1} \textstyle\sum_{j=1}^q n_j \ln(\langle X, a_j a_j^H \rangle) \\ \text{s.t.} \ & X \in \mathbb{H}_+^n, \ \ \operatorname{tr}(X) = \langle I_n, X \rangle = 1 \end{aligned} \tag{QST}$$

- In quantum physics, this problem aims to reconstruct the state of a quantum system using the measured output of particles [Hra04].
- $a_1, \ldots, a_q \in \mathbb{C}^n$, $\sum_{j=1}^q a_j a_j^H = I_n$ and $\sum_{j=1}^q n_j = m$.
- $\mathbb{H}_+^n$ denotes the cone of $n \times n$ complex Hermitian PSD matrices.
- AMG method: $X^0 \succ 0$, $\operatorname{tr}(X^0) = 1$,

$$\boxed{\begin{aligned} \hat{X}^{t+1} &= \exp\{\ln(X^t) + \ln(\nabla F(X^t))\} \\ X^{t+1} &= \hat{X}^{t+1} / \operatorname{tr}(\hat{X}^{t+1}) \end{aligned}}$$

(For any $X = \sum_{i=1}^n \lambda_i u_i u_i^H \succ 0$, $\ln(X) := \ln(\lambda_i) u_i u_i^H$.)

- Computational guarantee:

$$F^* - F(\bar{X}^t) \le \ln(n)/t, \quad \forall t \ge 1 \qquad \left[\bar{X}^t := (1/t) \textstyle\sum_{i=0}^{t-1} X^i\right]$$

Nesterov's Semi-definite Relaxation of Boolean QP

$$\begin{aligned} \max_X \quad & F(X) := 2\ln\left(\textstyle\sum_{i=1}^n \langle X, r_i r_i^\top\rangle^{1/2}\right) \\ \text{s.t.} \quad & X \in \mathbb{S}^n_+,\ \langle I_n, X\rangle = 1 \end{aligned} \qquad \text{(RBQP)}$$

Nesterov's Semi-definite Relaxation of Boolean QP

$$\begin{aligned} \max_X \quad & F(X) := 2\ln\left(\sum_{i=1}^n \langle X, r_i r_i^\top\rangle^{1/2}\right) \\ \text{s.t.} \quad & X \in \mathbb{S}_+^n,\ \langle I_n, X\rangle = 1 \end{aligned} \tag{RBQP}$$

▷ AMG method: $X^0 \succ 0$, $\mathrm{tr}(X^0) = 1$,

$$\boxed{\begin{aligned} \hat{X}^{t+1} &= \exp\{\ln(X^t) + \ln(\nabla F(X^t))\} \\ X^{t+1} &= \hat{X}^{t+1} / \mathrm{tr}(\hat{X}^{t+1}) \end{aligned}}$$

Nesterov's Semi-definite Relaxation of Boolean QP

$$\begin{aligned} \max_X \quad & F(X) := 2\ln\left(\textstyle\sum_{i=1}^n \langle X, r_i r_i^\top\rangle^{1/2}\right) \\ \text{s.t.} \quad & X \in \mathbb{S}_+^n,\ \langle I_n, X\rangle = 1 \end{aligned} \qquad \texttt{(RBQP)}$$

- ▷ AMG method: $X^0 \succ 0$, $\mathrm{tr}(X^0) = 1$,

$$\boxed{\begin{aligned} \hat{X}^{t+1} &= \exp\{\ln(X^t) + \ln(\nabla F(X^t))\} \\ X^{t+1} &= \hat{X}^{t+1} / \,\mathrm{tr}(\hat{X}^{t+1}) \end{aligned}}$$

- ▷ Computational guarantee:

$$F^* - F(\bar{X}^t) \le \ln(n)/t, \quad \forall t \ge 1 \qquad \left[\bar{X}^t := (1/t)\textstyle\sum_{i=0}^{t-1} X^i\right]$$

Comparison of Computational Guarantees

RSGM [BBT17; LFN18]: Relatively smooth gradient method
FW [ZF21]: Generalized FW method for LHB
GMG: Generalized Multiplicative gradient method
BSG [Nes11]: Barrier subgradient method

Table 1: Comparison of arithmetic-operations complexities (with $x^0 = (1/n)e$ or $X^0 = (1/n)I_n$)

	RSGM	FW	GMG	BSG	Regime
PET	$O\left(\frac{mn^2}{\varepsilon}\ln\left(\frac{\ln(n)}{\varepsilon}\right)\right)$	$O\left(\frac{m^2 n}{\varepsilon}\right)$	$O\left(\frac{mn\ln(n)}{\varepsilon}\right)$	$O\left(\frac{mn^2}{\varepsilon^2}\ln^2\left(\frac{n}{\varepsilon}\right)\right)$	$n = O(\exp(m))$
D-OPT	$O\left(\frac{mn^2}{\varepsilon}\ln\left(\frac{\ln(n/m)}{\varepsilon}\right)\right)$	$O\left(\frac{m^2 n}{\varepsilon}\right)$	$O\left(\frac{m^2 n\ln(n)}{\varepsilon}\right)$	$O\left(\frac{m^2 n^2}{\varepsilon^2}\ln^2\left(\frac{n}{\varepsilon}\right)\right)$	
QST	x?	$O\left(\frac{m^2 n^2}{\varepsilon}\right)$	$O\left(\frac{mn^2\ln(n)}{\varepsilon}\right)$	$O\left(\frac{mn^3}{\varepsilon^2}\ln^2\left(\frac{n}{\varepsilon}\right)\right)$	$n = O(\exp(m))$
RBQP	x?	x?	$O\left(\frac{n^3\ln(n)}{\varepsilon}\right)$	$O\left(\frac{n^4}{\varepsilon^2}\ln^2\left(\frac{n}{\varepsilon}\right)\right)$	

A Fun Comment From Steve

After presenting this work at U. Waterloo, Steve Vavasis commented:

A Fun Comment From Steve

After presenting this work at U. Waterloo, Steve Vavasis commented:

> "I have been working on optimization for many years, and I have developed a mental map to categorize each talk that I have attended. *But this talk simply doesn't fit into any of the existing categories!*"

Some Words About This Line of Research

This line of research has great potential, and many problems remain open:

Some Words About This Line of Research

This line of research has great potential, and many problems remain open:

- Can we identify new problem classes, based on new applications arising in machine learning and data science?

Some Words About This Line of Research

This line of research has great potential, and many problems remain open:

- Can we identify new problem classes, based on new applications arising in machine learning and data science?
- For the identified problem classes, are there faster first-order methods that can solve them?

Some Words About This Line of Research

This line of research has great potential, and many problems remain open:

- ▷ Can we identify new problem classes, based on new applications arising in machine learning and data science?
- ▷ For the identified problem classes, are there faster first-order methods that can solve them?
- ▷ Lower bound on computational guarantees?

Some Words About Future Research

Some Words About Future Research

- Besides my current research directions, I am also eager to explore the interface of optimization with other exciting topics:

Some Words About Future Research

▷ Besides my current research directions, I am also eager to explore the interface of optimization with other exciting topics:

- high-dimensional statistics
- online learning
- reinforcement learning
- decision-making under uncertainty ...

Some Words About Future Research

▷ Besides my current research directions, I am also eager to explore the interface of optimization with other exciting topics:

- high-dimensional statistics
- online learning
- reinforcement learning
- decision-making under uncertainty ...

▷ I also look forward to collaborating with many talented colleagues to discover new opportunities!

References

[Ari72] S. Arimoto. "An algorithm for computing the capacity of arbitrary discrete memoryless channels". In: *IEEE Trans. Inf. Theory* 18.1 (1972), pp. 14–20.

[BBT17] Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. "A Descent Lemma Beyond Lipschitz Gradient Continuity: First-Order Methods Revisited and Applications". In: *Math. Oper. Res.* 42.2 (2017), pp. 330–348.

[Csi84] I. Csiszar. "Information geometry and alternating minimization procedures". In: *Stat. Decis.* 1 (1984), pp. 205–237.

[Dvu20] Pavel Dvurechensky et al. "Self-Concordant Analysis of Frank-Wolfe Algorithms". In: *Proc. ICML.* 2020, pp. 2814–2824.

[HJN15] Z. Harchaoui, A. Juditsky, and A. Nemirovski. "Conditional gradient algorithms for norm-regularized smooth convex optimization". In: *Math. Program.* 152 (2015), 75–112.

[Hra04] Zdeněk Hradil et al. "Maximum-Likelihood Methodsin Quantum Mechanics". In: *Quantum State Estimation.* Springer Berlin Heidelberg, 2004, pp. 59–112.

[LFN18] Haihao. Lu, Robert M. Freund, and Yurii. Nesterov. "Relatively Smooth Convex Optimization by First-Order Methods, and Applications". In: *SIAM J. Optim.* 28.1 (2018), pp. 333–354.

[Nes11] Y. Nesterov. "Barrier subgradient method". In: *Math. Program.* (2011), 31—56.

[Nes98] Yu. Nesterov. "Semidefinite relaxation and nonconvex quadratic optimization". In: *Optim. Methods Softw.* 9.1-3 (1998), pp. 141–160.

[ZF21] Renbo Zhao and Robert M. Freund. *Global and Local Linear Convergence of Away-step Frank-Wolfe for Logarithmically-Homogeneous Barriers over Polytopes.* 2021.

References

[ZF22] Renbo Zhao and Robert M. Freund. *Analysis of the Frank-Wolfe Method for Convex Composite Optimization involving a Logarithmically-Homogeneous Barrier*. arXiv:2010.08999. 2022.

[Zha22] Renbo Zhao. "Non-Asymptotic Convergence Analysis of the Multiplicative Gradient Algorithm for PET-Type Problems". In: *Oper. Res. Lett., to appear* (2022).