A Unified Approach to Mixed-Integer Optimization: Nonlinear Formulations and Scalable Algorithms

Ryan Cory-Wright

November 2019

ORC, Massachusetts Institute of Technology
Joint work with Jean Pauphilet and Dimitris Bertsimas

Preprint available: ryancorywright.github.io

Motivation: A Tale of Two Problems

Best Subset Selection: Fit parsimonious model using at most k features

$$\min_{\boldsymbol{x}\in\mathbb{R}^p, \boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \text{ s.t.} \quad \sum_i z_i \leq k, -Mz_i \leq x_i \leq Mz_i, \forall i \in [p].$$

Motivation: A Tale of Two Problems

Best Subset Selection: Fit parsimonious model using at most k features

$$\min_{\boldsymbol{x}\in\mathbb{R}^p, \boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \text{ s.t.} \quad \sum_i z_i \leq k, -Mz_i \leq x_i \leq Mz_i, \forall i \in [p].$$

Facility Location: Build facilities z_i and ship amount $X_{i,j}$ to node j

$$\begin{aligned}\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \quad & \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \langle \boldsymbol{C}, \boldsymbol{X}\rangle \\ \text{s.t.} \quad & \sum_{j=1}^m X_{ij} \leq U_i,\ \forall i \in [n],\ \sum_{i=1}^n X_{ij} = d_j,\ \forall j \in [m], \\ & X_{ij} \leq U_i z_i,\ \forall i \in [n],\ \forall j \in [m].\end{aligned}$$

What do These Problems Have in Common?

- Two MIOs with big-M constraints between binary $\boldsymbol{z}$, continuous $\boldsymbol{x}$.
- MIO books (e.g. B.+Weismantel 2004) introduce problems this way *by default*.

What do These Problems Have in Common?

- Two MIOs with big-M constraints between binary $\boldsymbol{z}$, continuous $\boldsymbol{x}$.
- MIO books (e.g. B.+Weismantel 2004) introduce problems this way *by default*.
- But... big-M formulations actually **reformulations** of true problems!

What do These Problems Have in Common?

- Two MIOs with big-M constraints between binary $\boldsymbol{z}$, continuous $\boldsymbol{x}$.
- MIO books (e.g. B.+Weismantel 2004) introduce problems this way *by default*.
- But... big-M formulations actually **reformulations** of true problems!
- Here's another reformulation which scales as well/better.

A Tale of Two Problems: Second Order Cone Reformulation

Best Subset Selection:

$$\min_{\boldsymbol{x},\boldsymbol{\theta}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\theta} \text{ s.t.} \quad \sum_i z_i \leq k, x_i^2 \leq \theta_i z_i, \forall i \in [p].$$

A Tale of Two Problems: Second Order Cone Reformulation

Best Subset Selection:

$$\min_{\boldsymbol{x},\boldsymbol{\theta}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\theta} \text{ s.t.} \quad \sum_i z_i \leq k, x_i^2 \leq \theta_i z_i, \forall i \in [p].$$

Facility Location:

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X},\boldsymbol{\Theta}\in\mathbb{R}_+^{n\times m}} \quad \langle \boldsymbol{c},\boldsymbol{z}\rangle + \langle \boldsymbol{C},\boldsymbol{X}\rangle + \frac{1}{2\gamma}\langle \boldsymbol{E},\boldsymbol{\Theta}\rangle$$

$$\text{s.t.} \quad \sum_{j=1}^m X_{ij} \leq U_i,\ \forall i \in [n],\ \sum_{i=1}^n X_{ij} = d_j,\ \forall j \in [m],$$

$$X_{ij}^2 \leq z_i \Theta_{i,j},\ \forall i \in [n],\ \forall j \in [m].$$

A Tale of Two Problems: Second Order Cone Reformulation

Best Subset Selection:

$$\min_{\boldsymbol{x},\boldsymbol{\theta}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\theta} \text{ s.t.} \quad \sum_i z_i \le k, x_i^2 \le \theta_i z_i, \forall i \in [p].$$

Facility Location:

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X},\boldsymbol{\Theta}\in\mathbb{R}_+^{n\times m}} \quad \langle \boldsymbol{c},\boldsymbol{z}\rangle+\langle \boldsymbol{C},\boldsymbol{X}\rangle+\frac{1}{2\gamma}\langle \boldsymbol{E},\boldsymbol{\Theta}\rangle$$

$$\text{s.t.} \quad \sum_{j=1}^m X_{ij} \le U_i,\ \forall i\in[n],\ \sum_{i=1}^n X_{ij}=d_j,\ \forall j\in[m],$$

$$X_{ij}^2 \le z_i\Theta_{i,j},\ \forall i\in[n],\ \forall j\in[m].$$

SOCP formulations equivalent to and often more tractable than big-M.

A Tale of Two Problems: Second Order Cone Reformulation

Best Subset Selection:

$$\min_{\boldsymbol{x},\boldsymbol{\theta}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\frac{1}{2\gamma}\boldsymbol{e}^\top\boldsymbol{\theta} \text{ s.t.} \quad \sum_i z_i \le k, x_i^2 \le \theta_i z_i, \forall i \in [p].$$

Facility Location:

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X},\boldsymbol{\Theta}\in\mathbb{R}_+^{n\times m}} \quad \langle \boldsymbol{c},\boldsymbol{z}\rangle + \langle \boldsymbol{C},\boldsymbol{X}\rangle + \frac{1}{2\gamma}\langle \boldsymbol{E},\boldsymbol{\Theta}\rangle$$

$$\text{s.t.} \quad \sum_{j=1}^m X_{ij} \le U_i,\ \forall i\in[n],\ \sum_{i=1}^n X_{ij} = d_j,\ \forall j\in[m],$$

$$X_{ij}^2 \le z_i\Theta_{i,j},\ \forall i\in[n],\ \forall j\in[m].$$

SOCP formulations equivalent to and often more tractable than big-M.

How do we unify SOCP, big-M? And what are the true formulations?

The True Formulations

Best Subset Selection

$$\min_{\boldsymbol{x}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \text{ s.t.} \quad \sum_i z_i \le k, x_i = 0 \text{ if } z_i = 0,\ \forall i \in [p].$$

Facility Location:

$$\begin{aligned}
\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \quad & \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \langle \boldsymbol{C}, \boldsymbol{X}\rangle \\
\text{s.t.} \quad & \sum_{j=1}^m X_{ij} \le U_i,\ \forall i \in [n],\ \sum_{i=1}^n X_{ij} = d_j,\ \forall j \in [m], \\
& X_{ij} = 0 \text{ if } z_i = 0,\ \forall i \in [n],\ \forall j \in [m].
\end{aligned}$$

The True Formulations

Best Subset Selection

$$\min_{\boldsymbol{x}\in\mathbb{R}^p,\boldsymbol{z}\in\{0,1\}^p} \quad \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \text{ s.t.} \quad \sum_i z_i \le k, x_i = 0 \text{ if } z_i = 0,\ \forall i \in [p].$$

Facility Location:

$$\begin{aligned}\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \quad & \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \langle \boldsymbol{C}, \boldsymbol{X}\rangle \\ \text{s.t.} \quad & \sum_{j=1}^m X_{ij} \le U_i,\ \forall i \in [n],\ \sum_{i=1}^n X_{ij} = d_j,\ \forall j \in [m], \\ & X_{ij} = 0 \text{ if } z_i = 0,\ \forall i \in [n],\ \forall j \in [m].\end{aligned}$$

True formulations are MIOs with logical structure: $x = 0$ *if* $z = 0$.

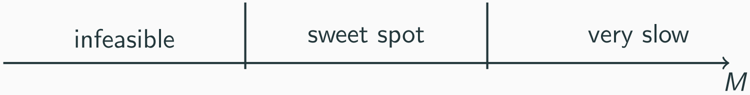
So what? What's wrong with big-M?

What's wrong with big-M?

- Big-M approach: "$x = 0$ if $z = 0$" as $-Mz \leq x \leq Mz$.
- Finding the right M is hard!

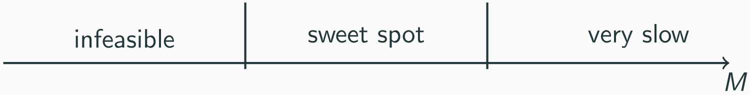
What's wrong with big-M?

- Big-M approach: "$x = 0$ if $z = 0$" as $-Mz \leq x \leq Mz$.
- Finding the right M is hard!



What’s wrong with big-M?

- Big-M approach: “$x = 0$ if $z = 0$” as $-Mz \leq x \leq Mz$.
- Finding the right M is hard!



Big-M constraints inhibit scalability; MISOCP constraints are expensive to manage and hard to branch over.
Alternatives are needed.

A Family of Problems With Logical Constraints

Logical on/off structure appears in many important problems!

Central problems in optimization/ML/statistics have logical relations between continuous variables $\boldsymbol{x}$, binary variables $\boldsymbol{z}$: $\boldsymbol{x} = \mathbf{0}$ *if* $\boldsymbol{z} = \mathbf{0}$.

A Family of Problems With Logical Constraints

Logical on/off structure appears in many important problems!

Central problems in optimization/ML/statistics have logical relations between continuous variables $\boldsymbol{x}$, binary variables $\boldsymbol{z}$: $\boldsymbol{x} = \mathbf{0}$ *if* $\boldsymbol{z} = \mathbf{0}$.

- **Best subset selection**: $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 : \ \|\boldsymbol{x}\|_0 \leq k$.

A Family of Problems With Logical Constraints

Logical on/off structure appears in many important problems!

Central problems in optimization/ML/statistics have logical relations between continuous variables $\boldsymbol{x}$, binary variables $\boldsymbol{z}$: $\boldsymbol{x} = \mathbf{0}$ *if* $\boldsymbol{z} = \mathbf{0}$.

- **Best subset selection**: $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 : \ \|\boldsymbol{x}\|_0 \leq k$.
- **Compressed sensing**: $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 : \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

A Family of Problems With Logical Constraints

Logical on/off structure appears in many important problems!

Central problems in optimization/ML/statistics have logical relations between continuous variables $\boldsymbol{x}$, binary variables $\boldsymbol{z}$: $\boldsymbol{x} = \mathbf{0}$ *if* $\boldsymbol{z} = \mathbf{0}$.

- **Best subset selection**: $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 : \ \|\boldsymbol{x}\|_0 \leq k$.
- **Compressed sensing**: $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 : \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.
- **Facility Location**:

$$\min_{\boldsymbol{z} \in \{0,1\}^n} \min_{\boldsymbol{X} \in \mathbb{R}_+^{n \times m}} \langle \boldsymbol{c}, \boldsymbol{z} \rangle + \langle \boldsymbol{C}, \boldsymbol{X} \rangle$$

$$\text{s.t.} \ \sum_{j=1}^{m} X_{ij} \leq U_i, \ \forall i \in [n], \ \sum_{i=1}^{n} X_{ij} = d_j, \ \forall j \in [m], \tag{1}$$

$$X_{ij} = 0 \ \text{ if } \ z_i = 0, \ \forall i \in [n], j \in [m].$$

A Family of Problems With Logical Constraints

Logical on/off structure appears in many important problems!

Central problems in optimization/ML/statistics have logical relations between continuous variables $\boldsymbol{x}$, binary variables $\boldsymbol{z}$: $\boldsymbol{x} = \boldsymbol{0}$ *if* $\boldsymbol{z} = \boldsymbol{0}$.

- **Best subset selection**: $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 : \ \|\boldsymbol{x}\|_0 \leq k$.
- **Compressed sensing**: $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 : \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.
- **Facility Location**:

$$\min_{\boldsymbol{z} \in \{0,1\}^n} \min_{\boldsymbol{X} \in \mathbb{R}_+^{n \times m}} \langle \boldsymbol{c}, \boldsymbol{z} \rangle + \langle \boldsymbol{C}, \boldsymbol{X} \rangle$$

$$\text{s.t.} \ \sum_{j=1}^m X_{ij} \leq U_i, \ \forall i \in [n], \ \sum_{i=1}^n X_{ij} = d_j, \ \forall j \in [m], \tag{1}$$

$$X_{ij} = 0 \ \text{ if } \ z_i = 0, \ \forall i \in [n], j \in [m].$$

- Many others: **Sparse Portfolio Selection, Network Design, Unit Commitment, Scheduling, Binary Quadratic, Sparse PCA, ...**

Modelling Contributions

- We propose a **non-linear reformulation** of the logical constraints, substituting xz for x.

Modelling Contributions

- We propose a **non-linear reformulation** of the logical constraints, substituting xz for x.
- We show that adding a $\frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ **ridge regularizer** to the objective is a viable and often more scalable alternative to big-M.

Modelling Contributions

- We propose a **non-linear reformulation** of the logical constraints, substituting xz for x.
- We show that adding a $\frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ **ridge regularizer** to the objective is a viable and often more scalable alternative to big-M.
- We unify ridge and big-M penalties under the lens of regularization.

Algorithmic Contributions

- By using strong duality, we derive a saddle-point reformulation, which is **exactly** solvable via an outer-approximation procedure.

Algorithmic Contributions

- By using strong duality, we derive a saddle-point reformulation, which is **exactly** solvable via an outer-approximation procedure.
- We obtain provably **near-optimal solutions in polynomial time** by solving a Boolean relaxation efficiently.

Algorithmic Contributions

- By using strong duality, we derive a saddle-point reformulation, which is **exactly** solvable via an outer-approximation procedure.
- We obtain provably **near-optimal solutions in polynomial time** by solving a Boolean relaxation efficiently.
- Our approach **is scalable**: it solves sparse regression problems with $100,000$s of covariates, sparse portfolio selection problems with 1000s of securities, network design problems with 100s of nodes.

The Unified Framework

A Mixed-Integer Nonlinear Program With Logical Constraints

$$\min_{\boldsymbol{z}\in\mathcal{Z},\ \boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex function}} + \underbrace{\Omega(\boldsymbol{x})}_{\text{regularizer}} \text{ s.t. } \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}},\ \forall i,$$

where:

A Mixed-Integer Nonlinear Program With Logical Constraints

$$\min_{\boldsymbol{z}\in\mathcal{Z},\ \boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex function}} + \underbrace{\Omega(\boldsymbol{x})}_{\text{regularizer}} \quad \text{s.t.} \quad \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}},\ \forall i,$$

where:

- $\mathcal{Z} \subseteq \{0,1\}^n$ constrains $\boldsymbol{z}$, e.g., cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$.

A Mixed-Integer Nonlinear Program With Logical Constraints

$$\min_{\boldsymbol{z}\in\mathcal{Z},\ \boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex function}} + \underbrace{\Omega(\boldsymbol{x})}_{\text{regularizer}} \ \text{s.t.}\ \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}},\ \forall i,$$

where:

- $\mathcal{Z} \subseteq \{0,1\}^n$ constrains $\boldsymbol{z}$, e.g., cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$.
- We model convex constraints $\boldsymbol{x} \in \mathcal{X}$ via $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$.
- The regularizer $\Omega(\cdot)$ convexifies the logical constraints.

A Mixed-Integer Nonlinear Program With Logical Constraints

$$\min_{\boldsymbol{z}\in\mathcal{Z},\ \boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex function}} + \underbrace{\Omega(\boldsymbol{x})}_{\text{regularizer}} \text{ s.t. } \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}},\ \forall i,$$

where:

- $\mathcal{Z} \subseteq \{0, 1\}^n$ constrains $\boldsymbol{z}$, e.g., cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \le k$.
- We model convex constraints $\boldsymbol{x} \in \mathcal{X}$ via $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$.
- The regularizer $\Omega(\cdot)$ convexifies the logical constraints. It is one of:
  1. A big-M penalty: $\Omega(\boldsymbol{x}) = 0$ if $\|\boldsymbol{x}\|_\infty \le M$ and $+\infty$ otherwise.
  2. A ridge penalty: $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$.

A Mixed-Integer Nonlinear Program With Logical Constraints

$$\min_{\boldsymbol{z}\in\mathcal{Z},\ \boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex function}} + \underbrace{\Omega(\boldsymbol{x})}_{\text{regularizer}} \text{ s.t. } \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}},\ \forall i,$$

where:

- $\mathcal{Z} \subseteq \{0,1\}^n$ constrains $\boldsymbol{z}$, e.g., cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$.
- We model convex constraints $\boldsymbol{x} \in \mathcal{X}$ via $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$.
- The regularizer $\Omega(\cdot)$ convexifies the logical constraints. It is one of:
  1. A big-M penalty: $\Omega(\boldsymbol{x}) = 0$ if $\|\boldsymbol{x}\|_\infty \leq M$ and $+\infty$ otherwise.
  2. A ridge penalty: $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$.

All six problems on the second slide fit into this framework!

A Mixed-Integer Nonlinear Program With Logical Constraints

$$\min_{\boldsymbol{z}\in\mathcal{Z},\ \boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex function}} + \underbrace{\Omega(\boldsymbol{x})}_{\text{regularizer}} \ \text{s.t.}\ \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}},\ \forall i,$$

where:

- $\mathcal{Z} \subseteq \{0,1\}^n$ constrains $\boldsymbol{z}$, e.g., cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$.
- We model convex constraints $\boldsymbol{x} \in \mathcal{X}$ via $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$.
- The regularizer $\Omega(\cdot)$ convexifies the logical constraints. It is one of:
  1. A big-M penalty: $\Omega(\boldsymbol{x}) = 0$ if $\|\boldsymbol{x}\|_\infty \leq M$ and $+\infty$ otherwise.
  2. A ridge penalty: $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$.

All six problems on the second slide fit into this framework!

- Allows us to solve all six problems using the same piece of code.

Fitting Facility Location Within Our Framework

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \boldsymbol{c}^\top \boldsymbol{z} + \langle \boldsymbol{C}, \boldsymbol{X} \rangle$$

$$\text{s.t.} \quad \sum_{i=1}^n X_{ij} = d_j,\ \forall j \in [m],\ \sum_{j=1}^m X_{ij} \le U_i,\ \forall i \in [n],$$

$$X_{ij} = 0 \ \text{ if } \ z_i = 0,\ \forall i \in [n], j \in [m].$$

Fitting Facility Location Within Our Framework

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle + \langle \boldsymbol{C}, \boldsymbol{X}\rangle + \begin{cases} 0, & \text{if } \boldsymbol{X}^\top \boldsymbol{e} = \boldsymbol{d} \\ \infty, & \text{o/w} \end{cases} + \begin{cases} 0, & \text{if } \boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{u} \\ \infty, & \text{o/w} \end{cases}$$

$$\text{s.t.} \quad X_{ij} = 0 \ \text{ if } \ z_i = 0, \ \forall i \in [n], j \in [m].$$

Fitting Facility Location Within Our Framework

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle \underbrace{+\langle \boldsymbol{C}, \boldsymbol{X}\rangle + \begin{cases} 0, & \text{if } \boldsymbol{X}^\top \boldsymbol{e} = \boldsymbol{d} \\ \infty, & \text{o/w} \end{cases}}_{=:g(\boldsymbol{X}) \text{ convex function}} \underbrace{+ \begin{cases} 0, & \text{if } \boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{u} \\ \infty, & \text{o/w} \end{cases}}_{=:\Omega(\boldsymbol{X}) \text{ regularizer}}$$

$$\text{s.t.} \quad X_{ij} = 0 \text{ if } z_i = 0, \ \forall i \in [n], j \in [m].$$

Fitting Facility Location Within Our Framework

$$\min_{\boldsymbol{z}\in\{0,1\}^n} \min_{\boldsymbol{X}\in\mathbb{R}_+^{n\times m}} \langle \boldsymbol{c}, \boldsymbol{z}\rangle \underbrace{+\langle \boldsymbol{C}, \boldsymbol{X}\rangle + \begin{cases} 0, & \text{if } \boldsymbol{X}^\top \boldsymbol{e} = \boldsymbol{d} \\ \infty, & \text{o/w} \end{cases} + \begin{cases} 0, & \text{if } \boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{u} \\ \infty, & \text{o/w} \end{cases}}_{=: g(\boldsymbol{X}) \text{ convex function}}$$

$$\text{s.t.} \quad X_{ij} = 0 \;\; \text{if} \;\; z_i = 0, \; \forall i \in [n], j \in [m].$$

Simplifying the Problem

We rewrite the problem as

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \; f(\boldsymbol{z}),$$

where

$$f(\boldsymbol{z}) = \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex}} + \underbrace{\Omega(\boldsymbol{x})}_{\textit{regularizer}} \quad \text{s.t. } \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}}, \; \forall i,$$

Simplifying the Problem

We rewrite the problem as

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \; f(\boldsymbol{z}),$$

where

$$\begin{aligned} f(\boldsymbol{z}) &= \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex}} + \underbrace{\Omega(\boldsymbol{x})}_{\textit{regularizer}} \quad \text{s.t. } \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}}, \; \forall i, \\ &= \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{z} \circ \boldsymbol{x}) + \Omega(\boldsymbol{z} \circ \boldsymbol{x}) \qquad \text{non-linear} \end{aligned}$$

Simplifying the Problem

We rewrite the problem as

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \; f(\boldsymbol{z}),$$

where

$$\begin{aligned}
f(\boldsymbol{z}) &= \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex}} + \underbrace{\Omega(\boldsymbol{x})}_{\textit{regularizer}} \quad \text{s.t. } \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}}, \; \forall i, \\
&= \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{z} \circ \boldsymbol{x}) + \Omega(\boldsymbol{z} \circ \boldsymbol{x}) && \text{non-linear} \\
&= \max_{\boldsymbol{\alpha}} \; \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{h(\boldsymbol{\alpha})}_{\text{concave}} - \sum_i z_i \Omega^\star(\alpha_i) && \text{strong duality}
\end{aligned}$$

Simplifying the Problem

We rewrite the problem as

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \; f(\boldsymbol{z}),$$

where

$$\begin{aligned} f(\boldsymbol{z}) &= \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{g(\boldsymbol{x})}_{\text{convex}} + \underbrace{\Omega(\boldsymbol{x})}_{\textit{regularizer}} \quad \text{s.t.} \; \underbrace{x_i = 0 \text{ if } z_i = 0}_{\text{logical constraint}}, \; \forall i, \\ &= \min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{z} \circ \boldsymbol{x}) + \Omega(\boldsymbol{z} \circ \boldsymbol{x}) && \text{non-linear} \\ &= \max_{\boldsymbol{\alpha}} \; \boldsymbol{c}^\top \boldsymbol{z} + \underbrace{h(\boldsymbol{\alpha})}_{\text{concave}} - \sum_i z_i \Omega^\star(\alpha_i) && \text{strong duality} \end{aligned}$$

which proves $f(z)$ is convex!

So What?

The Outer-Approximation Method

Our saddle-point representation:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \ f(\boldsymbol{z}) = \ \min_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{\alpha}} \ \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

lends itself to a tractable outer-approximation method.

The Outer-Approximation Method

Our saddle-point representation:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z}) = \min_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{\alpha}} \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

lends itself to a tractable outer-approximation method.

- Fix $\boldsymbol{z}_0$, solve an easy convex program to obtain $\boldsymbol{\alpha}^\star(\boldsymbol{z}_0)$ and $f(\boldsymbol{z}_0)$.

The Outer-Approximation Method

Our saddle-point representation:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z}) = \min_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{\alpha}} \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

lends itself to a tractable outer-approximation method.

- Fix $\boldsymbol{z}_0$, solve an easy convex program to obtain $\boldsymbol{\alpha}^\star(\boldsymbol{z}_0)$ and $f(\boldsymbol{z}_0)$.
- Obtain subgradient for all n indices **even those where** $z_{i,0} = 0$.

The Outer-Approximation Method

Our saddle-point representation:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \ f(\boldsymbol{z}) = \ \min_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{\alpha}} \ \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

lends itself to a tractable outer-approximation method.

- Fix $\boldsymbol{z}_0$, solve an easy convex program to obtain $\boldsymbol{\alpha}^\star(\boldsymbol{z}_0)$ and $f(\boldsymbol{z}_0)$.
- Obtain subgradient for all n indices **even those where** $z_{i,0} = 0$.
- $f(\boldsymbol{z}) \geq f(\boldsymbol{z}_0) + \nabla f(\boldsymbol{z}_0)^\top (\boldsymbol{z} - \boldsymbol{z}_0)$ is a valid outer-approximation cut.

The Outer-Approximation Method

Our saddle-point representation:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \ f(\boldsymbol{z}) = \ \min_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{\alpha}} \ \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

lends itself to a tractable outer-approximation method.

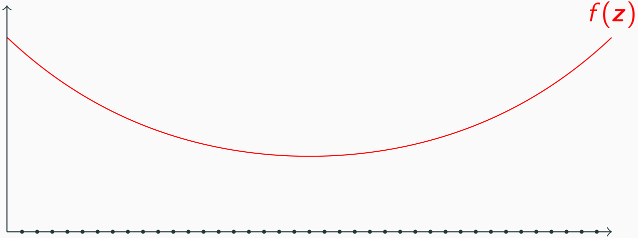
- Fix $\boldsymbol{z}_0$, solve an easy convex program to obtain $\boldsymbol{\alpha}^\star(\boldsymbol{z}_0)$ and $f(\boldsymbol{z}_0)$.
- Obtain subgradient for all n indices **even those where** $z_{i,0} = 0$.
- $f(\boldsymbol{z}) \geq f(\boldsymbol{z}_0) + \nabla f(\boldsymbol{z}_0)^\top (\boldsymbol{z} - \boldsymbol{z}_0)$ is a valid outer-approximation cut.
- Iteratively adding cuts, minimizing piecewise linear underestimator in Julia/CPLEX minimizes $f(\boldsymbol{z})$. Using Branch-and-Cut with lazy constraints solves entire problem using one branch-and-bound tree.
- As will see in numerical results, solves very large-scale problems.

The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.

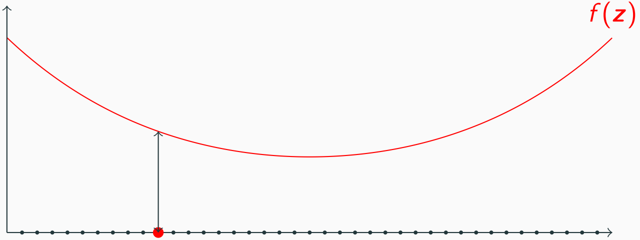


The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.

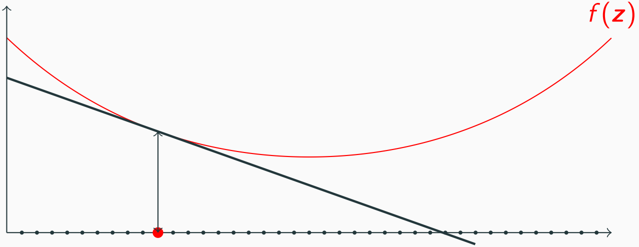


The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.

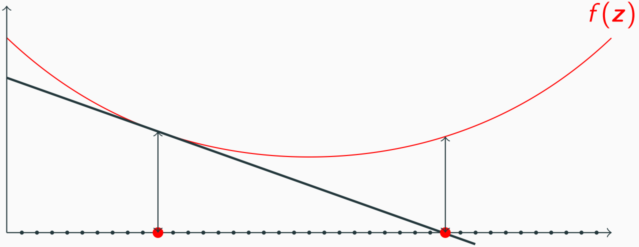


The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.

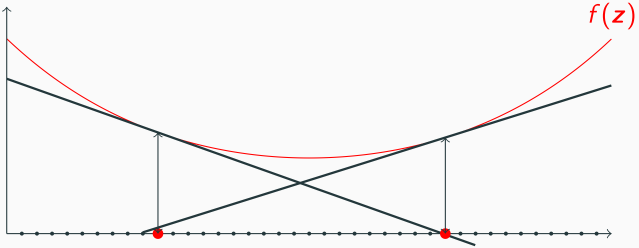


The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.

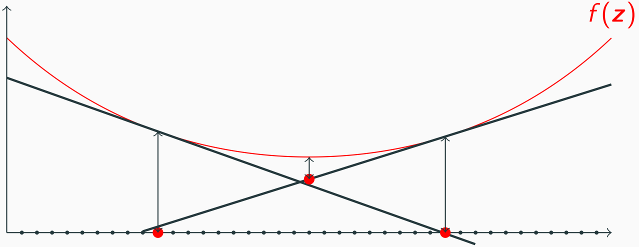


The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.

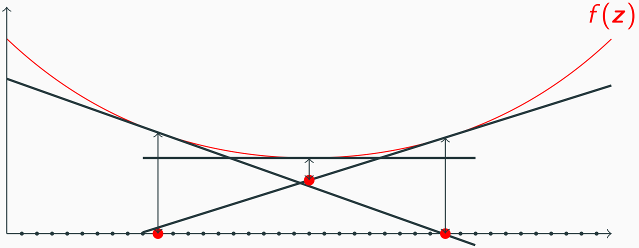


The Outer Approximation Process

We solve the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}} f(\boldsymbol{z})$$

by iteratively minimizing a piecewise linear underestimator of f.



A Boolean Relaxation

$$\min_{\boldsymbol{z} \in \mathrm{Conv}(\mathcal{Z})} \max_{\boldsymbol{\alpha}} \; \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

A Boolean Relaxation

$$\min_{\boldsymbol{z} \in \mathrm{Conv}(\mathcal{Z})} \max_{\boldsymbol{\alpha}} \ \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

- Solve by sub-gradient descent method, or transform to SOCP.
- Randomly rounding relaxation $\boldsymbol{z}^\star$ according to $z_i \sim \mathrm{Bernoulli}(z_i^\star)$ gives a Boolean vector $\boldsymbol{z}$. How good is it?

A Boolean Relaxation

$$\min_{\boldsymbol{z}\in\mathrm{Conv}(\mathcal{Z})} \max_{\boldsymbol{\alpha}} \ \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_i z_i \Omega^\star(\alpha_i)$$

- Solve by sub-gradient descent method, or transform to SOCP.
- Randomly rounding relaxation $\boldsymbol{z}^\star$ according to $z_i \sim \mathrm{Bernoulli}(z_i^\star)$ gives a Boolean vector $\boldsymbol{z}$. How good is it?
- Let $\boldsymbol{z}$ be a random rounding of $\boldsymbol{z}^\star$. Then,

$$0 \leq f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \leq \epsilon$$

with probability at least

$$1 - |\mathcal{R}| \exp\left(\frac{-\epsilon^2}{\kappa}\right)$$

  - $|\mathcal{R}|$ is number of strictly fractional entries in $\boldsymbol{z}^\star$.
  - κ is a function of $|\mathcal{R}|$, problem data.

Unifying Big-M and Ridge Via Regularization

Suppose that we take the dual of the saddle-point formulation:

- Under big-M regularization, we obtain:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \min_{\boldsymbol{x}} \ \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) \text{ s.t.} - Mz_i \leq x_i \leq Mz_i, \ \forall i,$$

- Under ridge regularization, we obtain:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \min_{\boldsymbol{x}} \ \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \frac{1}{2\gamma} \boldsymbol{e}^\top \boldsymbol{\theta} \text{ s.t. } x_i^2 \leq \theta_i z_i, \ \forall i,$$

- Recover convex relaxation by relaxing integrality on $\boldsymbol{z}$.
- Applying outer approximation is typically much faster than solving directly via CPLEX/Gurobi.

How does the approach perform on real data?

Sparse Empirical Risk Minimization Scalability

- For regression $f(\boldsymbol{z})$ is closed form, scales to $100,000$s of features.
- For classification, $f(\boldsymbol{z})$ is cheap, scales to $10,000$s of features.
- Outer-approximation algorithm is more accurate than ElasticNet, MCP, SCAD, and runtimes are comparable to Lasso.
- Code available: github.com/jeanpauphilet.

Sparse Portfolio Selection Scalability

Solves sparse portfolio selection problems with $1,000$s of securities.

Reference	Solution method	Size (no. securities)
Frangioni and Gentile ($'09$)	Perspective cut+SDP	400
Bonami and Lejeune ($'09$)	Nonlinear Branch-and-Bound	200
Gao and Li ($'13$)	SOCP relaxation Branch-and-Bound	300
Cui et al. ($'13$)	SOCP relaxation Branch-and-Bound	300
Zheng et. al. ($'14$)	SDP Branch-and-Bound	400
Frangioni et. al. ($'16$)	Aprox. Proj. Perspective Cut	400
Bertsimas and C-W ($'18$)	OA with γ-regularization	$3,200$

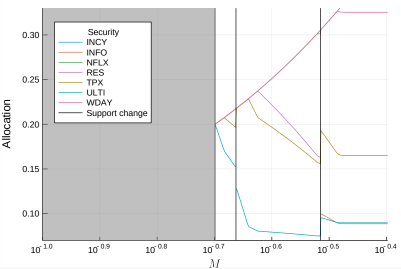
Network Design Scalability

- $f(\boldsymbol{z})$ obtained by solving a quadratic program.
- Approach solves problems with 100s of nodes.
- Objective value 5% better than CPLEX for small problems, 40% better for large problems.

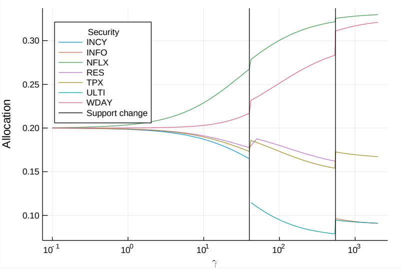
Big-M vs. Ridge Regularization: Which one should I use?

The Two Regularizers Perform Fundamentally The Same Role

Example: Selecting five securities from the Russell 1000



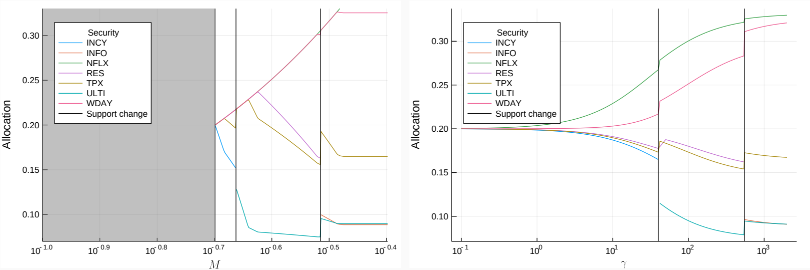
(a) Big-M regularization



(b) Ridge-regularization.

The Two Regularizers Perform Fundamentally The Same Role

Example: Selecting five securities from the Russell 1000



(a) Big-M regularization

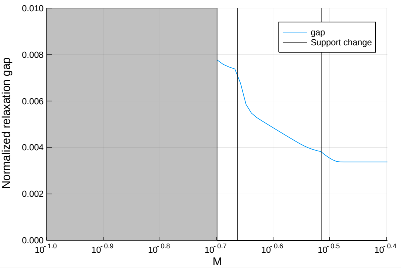
(b) Ridge-regularization.

There are differences:

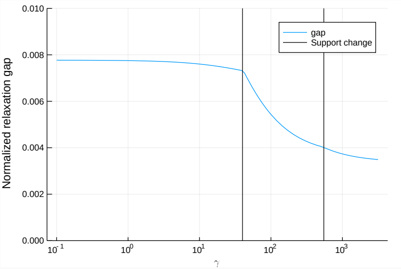
- Setting $M < M_0$ renders the problem infeasible; feasible for $\gamma > 0$.
- Setting $M > M_1$ recovers unregularized problem; not so for finite γ.

The Bound Gaps Are Comparable For Portfolio Selection

Example: selecting five stocks from the Russell 1000



(a) Big-M regularization



(b) γ-regularization.

Depending on application, one may give smaller gaps than other.

Ridge vs Big-M Regularization: What Works Best When?

It depends on the problem (you need to try both). But ...

Ridge vs Big-M Regularization: What Works Best When?

It depends on the problem (you need to try both). But ...

- If the problem is highly degenerate, ridge probably works better, since it breaks dual degeneracy, while big-M does not.
- If the objective is linear, big-M probably work betters.
  - For binary quadratic optimization, big-M works better.
- If the objective is already quadratic, ridge probably works better.
  - For sparse regression, sparse portfolio selection, ridge works better.

Main Messages and Highlights

- Don't feel married to big-M! We provide a **non-linear alternative** which often scales as well or better: substituting xz for x and adding a ridge regularizer.

Main Messages and Highlights

- Don't feel married to big-M! We provide a **non-linear alternative** which often scales as well or better: substituting xz for x and adding a ridge regularizer.
- By using strong duality, we derive a saddle-point reformulation, which is **exactly** solvable via an outer-approximation procedure.

Main Messages and Highlights

- Don't feel married to big-M! We provide a **non-linear alternative** which often scales as well or better: substituting xz for x and adding a ridge regularizer.
- By using strong duality, we derive a saddle-point reformulation, which is **exactly** solvable via an outer-approximation procedure.
- Our approach: outer-approximation+warm-start+random rounding **is scalable**.

Thanks for listening!

Questions?

Preprint available at: ryancorywright.github.io

Selected References

- Bertsimas, D., Cory-Wright, R.: A scalable algorithm for sparse portfolio selection. arXiv:1811.00138 (2018), revision submitted Sept 2019.
- Bertsimas, D., Cory-Wright, R., Pauphilet, J: A Unified Approach to Mixed-Integer Optimization: Nonlinear Reformulations and Scalable Algorithms. arXiv:1907.02109 (2019).
- Bertsimas, D., Lamperski, J., Pauphilet, J: Certifiably optimal sparse inverse covariance estimation. Math. Prog. (2019).
- Bertsimas, D., Pauphilet, J., Van Parys, B.: Sparse regression: Scalable algorithms and empirical performance. arXiv:1902.06547 (2019)
- Bertsimas, D., Van Parys, B.: Sparse high dimensional regression: Exact scalable algorithms and phase transitions (2019). Ann. Statist., to appear (2019).
- Dong, H., Chen, K., Linderoth, J: Regularization vs. Relaxation: A conic optimization perspective of statistical variable selection. Opt. Online (2015).
- Frangioni, A., Gentile, M. Perspective cuts for a class of convex 0–1 mixed integer programs. Math. Prog. **106**:225–236 (2006).
- Pilanci, M., Wainwright, M.J., El Ghaoui, L.: Sparse learning via boolean relaxations. Math. Prog. **151**(1), 63–87 (2015).
- Zheng, X., Sun, X., Li, D.: Improving the Performance of MIQP Solvers for Quadratic Programs with Cardinality and Minimum Threshold Constraints: A Semidefinite Program Approach. INFORMS J. Comput. **26**(4):690–703 (2014).

Supplementary Material

Back up slide: Sparse PCA Formulation

$$\min_{\boldsymbol{z} \in \{0,1\}^n : \boldsymbol{e}^\top \boldsymbol{z}} f(\boldsymbol{z}),$$

where

$$\begin{aligned} f(\boldsymbol{z}) = \min_{\boldsymbol{X} \in S_+^n} \quad & \langle -\boldsymbol{\Sigma}, \boldsymbol{X} \rangle \\ \text{s.t.} \quad & \mathrm{tr}(\boldsymbol{X}) = 1, \\ & X_{i,j} = 0 \text{ if } z_i = 0, \ \forall i, j \in [n], \\ & X_{i,j} = 0 \text{ if } z_j = 0, \ \forall i, j \in [n]. \end{aligned}$$

Back up slide: Sparse PCA Formulation

$$\min_{\boldsymbol{z}\in\{0,1\}^n:\boldsymbol{e}^\top \boldsymbol{z}} f(\boldsymbol{z}),$$

where

$$\begin{aligned} f(\boldsymbol{z}) = \min_{\boldsymbol{X}\in S_+^n} \quad & \langle -\boldsymbol{\Sigma}, \boldsymbol{X}\rangle \\ \text{s.t.} \quad & \mathrm{tr}(\boldsymbol{X}) = 1, \\ & X_{i,j} = 0 \text{ if } z_i = 0, \ \forall i,j \in [n], \\ & X_{i,j} = 0 \text{ if } z_j = 0, \ \forall i,j \in [n]. \end{aligned}$$

Caution: better to use big-M regularization here. With big-M regularization some optimal $\boldsymbol{X}$ is always rank-1, but with ridge regularization optimal solutions are not rank-1.

Back-up Slide: Relationship With Perspective Cuts

- The dual of our saddle point formulation with ridge regularization is a perspective reformulation. So, perspective cuts are similar.
- Key difference with perspective cuts: we decompose into master and sub problems, allows us to take advantage of subproblem structure.
  - For SPS we transform cuts into Pareto optimal cuts without solving an aux. problem.
- Our approach can also be implemented using one lazy callback; perspective cuts require a more complicated implementation (see Frangioni+Gentile '06, for a discussion).
- Full details on differences in section 3.5 of paper.

Back-up Slide: When does big-M and/or ridge work better?

It depends on the problem

- Problems with convex quadratic objectives generally benefit more from ridge regularization. E.g., ridge works better for sparse regression, sparse portfolio selection.
- No clear advantage for problems with linear objectives, ridge slightly better (e.g. FLP, ND).
- Big-M clearly better for problems with linear objectives and small M's, e.g., binary quadratic optimization.

Back-up: Does Imposing a Regularizer Change the Problem?

Sort of, but not really:

- In many cases there is natural regularization
  - A quadratic term with a positive semidefinite hessian matrix gives natural ridge regularization.
  - Boundedness gives natural big-M regularization.
- You can also obtain the optimal $\boldsymbol{z}$ for a lightly regularized problem, fix $\boldsymbol{z}$ and resolve the unregularized problem.
  - Section 3.4 shows that this strategy is certifiably near-optimal.
- Regularization is intimately related to robustness anyway, and therefore usually beneficial.
  - E.g. in portfolio selection, ridge and big-M regularization both push towards the $\frac{1}{n}$ strategy.

Back-up Slide: Can we use other penalties?

Yes! As discussed in B./Lamperski/P. (2019) Appendix A.1, can use any regularizer $\Omega(\boldsymbol{x})$ which satisfies:

- Decomposability: $\Omega(\boldsymbol{x}) := \sum_i \Omega(x_i)$.
- Regularizes towards 0: $\min_{\boldsymbol{x}} \Omega(\boldsymbol{x}) = \Omega(\mathbf{0})$.

For instance, can use $\| \cdot \|_p^p$ for any $p > 1$. Issue is tractability; ℓ_p^p norms leads to (less tractable) power-cone representable subproblems.

May be beneficial when there is natural ℓ_p regularization (e.g. in machine scheduling problems[1]).

[1] See Akturk, M.S., Atamturk, A., Gurel, S.: *A strong conic quadratic reformulation for machine-job assignment with controllable processing times*. ORL (2009).

Connection to Perspective Formulations

- The bi-dual formulation with ridge regularization is usually called a *perspective* formulation[2]
- Called a "perspective" formulation because we are minimizing

$$\frac{\gamma}{2}\sum_i z_i f(\frac{x_i}{z_i}) = \frac{\gamma}{2}\sum_i \frac{x_i^2}{z_i}, \text{ where } f(x) = x^2, \ \frac{x}{0} = \begin{cases} 0 \text{ if } x = 0 \\ +\infty \text{ o/w.} \end{cases}$$

  rather than $\frac{\gamma}{2}\sum_i x_i^2$. Formulations equivalent when $z_i \in \{0, 1\}$, but perspective strictly tighter on $z_i \in (0, 1)$.
- The perspective formulation actually gives the convex hull of the epigraph if the rest of the problem is "nice".

[2] A great survey is Günlük, O., Linderoth, J.: Perspective reformulations of mixed integer nonlinear programs with indicator variables. MP (2010)

Selected References

- Bertsimas, D., Cory-Wright, R.: A scalable algorithm for sparse portfolio selection. arXiv:1811.00138 (2018), revision submitted Sept 2019.
- Bertsimas, D., Cory-Wright, R., Pauphilet, J: A Unified Approach to Mixed-Integer Optimization: Nonlinear Reformulations and Scalable Algorithms. arXiv:1907.02109 (2019).
- Bertsimas, D., Lamperski, J., Pauphilet, J: Certifiably optimal sparse inverse covariance estimation. Math. Prog. (2019).
- Bertsimas, D., Pauphilet, J., Van Parys, B.: Sparse regression: Scalable algorithms and empirical performance. arXiv:1902.06547 (2019)
- Bertsimas, D., Van Parys, B.: Sparse high dimensional regression: Exact scalable algorithms and phase transitions (2019). Ann. Statist., to appear (2019).
- Dong, H., Chen, K., Linderoth, J: Regularization vs. Relaxation: A conic optimization perspective of statistical variable selection. Opt. Online (2015).
- Frangioni, A., Gentile, M. Perspective cuts for a class of convex 0–1 mixed integer programs. Math. Prog. **106**:225–236 (2006).
- Pilanci, M., Wainwright, M.J., El Ghaoui, L.: Sparse learning via boolean relaxations. Math. Prog. **151**(1), 63–87 (2015).
- Zheng, X., Sun, X., Li, D.: Improving the Performance of MIQP Solvers for Quadratic Programs with Cardinality and Minimum Threshold Constraints: A Semidefinite Program Approach. INFORMS J. Comput. **26**(4):690–703 (2014).