Optimization methods for ℓ_0-problems

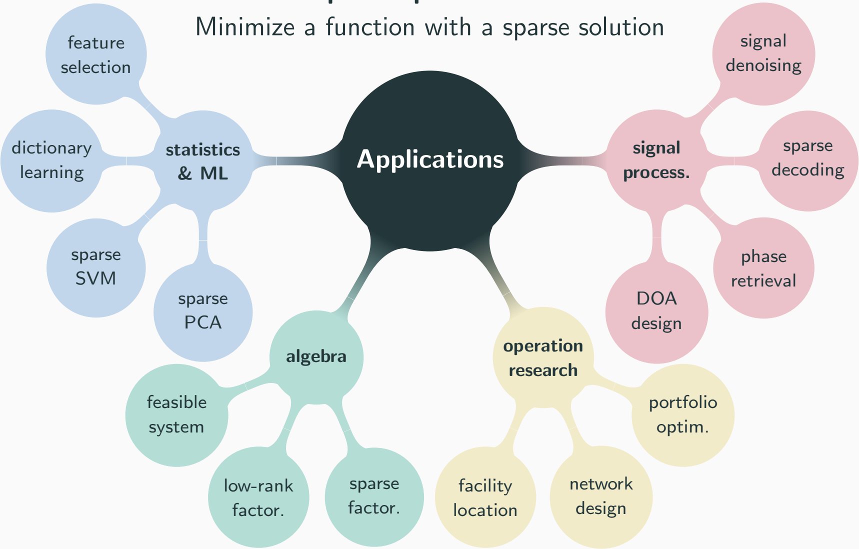
Théo Guyard

ML-MTP workshop – December 9th, 2024

Sparse optimization

Sparse optimization

Minimize a function with a sparse solution



A. Tillmann *et. al* (2024)

Compressed sensing

Compressed sensing

Sparse signal

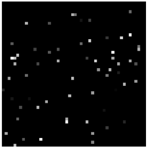
$\mathbf{x} \in \mathbf{R}^n$

Linear compression

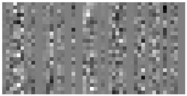
Recover $\mathbf{x}$ from $\mathbf{A}$ and $\mathbf{y}$

Observation

$\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon \in \mathbf{R}^m$



n pixels



m pixels, $m \ll n$

Goal

Find $\mathbf{x}$ such that $\mathbf{y} \simeq \mathbf{A}\mathbf{x}$

$m \ll n$: no unique solution

Goal

Find $\mathbf{x}$ sparse such that $\mathbf{y} \simeq \mathbf{A}\mathbf{x}$

Feature selection

Tabular ML dataset

	Feature 1	**Feature 2**	...	**Feature n**	**Target**
Sample 1	$a_{1,1}$	$a_{1,2}$	$\cdots$	$a_{1,n}$	y_1
Sample 2	$a_{2,1}$	$a_{2,2}$	$\cdots$	$a_{2,n}$	y_2
Sample 3	$a_{3,1}$	$\boxed{\mathbf{A} \in \mathbf{R}^{m\times n}}$	$\cdots$	$a_{3,n}$	$\boxed{\mathbf{y} \in \mathbf{R}^{m}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Sample m	$a_{m,1}$	$a_{m,2}$	$\cdots$	$a_{m,n}$	y_m

Features $\mathbf{A} \in \mathbf{R}^{m\times n}$ $\xleftrightarrow[\text{weights } \mathbf{x} \in \mathbf{R}^n]{}$ Target $\mathbf{y} = \phi(\mathbf{A}\mathbf{x})$

Model accuracy

Loss $\mathcal{L}_\phi(\mathbf{A}\mathbf{x}, \mathbf{y})$

Model explicability

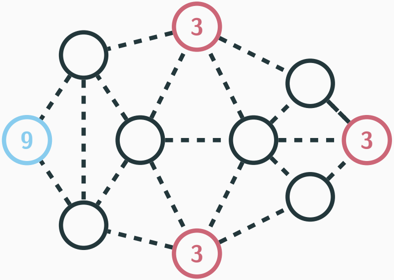
Use few features

Goal

Find $\mathbf{x}$ sparse such that $\mathcal{L}_\phi(\mathbf{A}\mathbf{x}, \mathbf{y})$ is small

Network design

Max. capacity per edge: 10
Edge construction cost: 5



Which edges to build to transport units from source to sink nodes ?

Minimized, constrained, or regularized problem ?

Sparse optimization

Minimize a function with a sparse solution

Quantify cost

$f(\mathbf{x})$

Expression to minimize

Quantify sparsity

$\|\mathbf{x}\|_0$

Count non-zeros

Constrained problem

$$\begin{array}{ll} \min_{\mathbf{x}\in\mathbf{R}^n} & f(\mathbf{x}) \\ \text{subject to} & \|\mathbf{x}\|_0 \leq s \end{array}$$

Minimized problem

$$\begin{array}{ll} \min_{\mathbf{x}\in\mathbf{R}^n} & \|\mathbf{x}\|_0 \\ \text{subject to} & f(\mathbf{x}) \leq \epsilon \end{array}$$

Regularized problem

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0$$

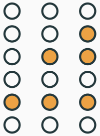
A bit of history

Problem

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0$$

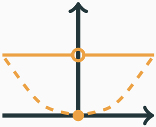
NP-hard to solve

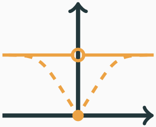
1995	**2000**	**2005**	**2010**	**2015**
Heuristics	Recovery cond.	Convex approx.	Concave approx.	Exact methods
MP, OMP, ...	RIP, NSP, ...	Lasso, Elastic-Net, ...	SCAD, MCP, ...	MIP, BnB, ...
S. Mallat (1993)	E. Candes (2004)	R. Tibshirani (2005)	C. Zhang (2010)	D. Bertsimas (2016)

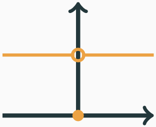


$\mathbf{x}^1$ $\mathbf{x}^2$ $\mathbf{x}^3$

OMP solves ℓ_0-problem under RIP







Complexity-tractability balance

[Slide, tell what's my point with this talk]

Mixed-Integer Programming

MIP – Pipeline

Application

ML, Stats, Signal, Operation Research, ...

↓

Problem

$\min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0$

↓

MIP formulation

Standardized expressions

↓

Problem solutions

Modelling framework

Python → `Cvxpy`

Julia → `JuMP`

C++ → `CMPL`

Matlab → `YALMIP`

...

Generic MIP solver

Gurobi, CPLEX

SCIP, Mosek

CbC, ...

MIP – Problem formulation

Problem

$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

linearize ℓ_0-norm

Lifted formulation

$$\begin{cases} \min & f(\mathbf{x}) + \lambda\mathbf{1}^{\mathrm{T}}\mathbf{z} + h(\mathbf{x}) \\ \text{s.t.} & x_i = 0 \implies z_i = 0, \ \forall i \\ & \mathbf{x} \in \mathbf{R}^n, \ \mathbf{z} \in \{0,1\}^n \end{cases}$$

avoid logical cstr.

MIP formulation

$$\begin{cases} \min & f(\mathbf{x}) + \lambda\mathbf{1}^{\mathrm{T}}\mathbf{z} + h_{\text{mip}}(\mathbf{x},\mathbf{z}) \\ \text{s.t.} & \mathbf{x} \in \mathbf{R}^n, \ \mathbf{z} \in \{0,1\}^n \end{cases}$$

Penalty term

Separable h, either application-based or introduced on purpose

Lifted ℓ_0-norm formulation

$\|\mathbf{x}\|_0 = \mathbf{1}^{\mathrm{T}}\mathbf{z}$ with $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{z} \in \{0,1\}^n$

if $x_i = 0 \iff z_i = 0$ for all $i \in [1, n]$

Logical constraint standardization

$h(\mathbf{x})$	$h_{\text{mip}}(\mathbf{x},\mathbf{z})$
$\text{Ind}(\|\mathbf{x}\|_\infty \leq M)$	$\text{Ind}(-M\mathbf{z} \leq \mathbf{x} \leq M\mathbf{z})$
$\beta\|\mathbf{x}\|_2^2$	$\sum_{i=1}^n \beta \frac{x_i^2}{z_i}$

MIP – Hands-on with cvxpy

```
$ pip install cvxpy
```

Sparse regression

Find $\mathbf{x}$ sparse such that $\mathbf{y} \simeq \mathbf{A}\mathbf{x}$

$\downarrow$

Optimization problem

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

- $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$
- $g(\mathbf{x}) = \mathrm{Ind}(\|\mathbf{x}\|_\infty \leq M)$

$\downarrow$

MIP formulation

$$\begin{cases} \min\ f(\mathbf{x}) + \lambda\mathbf{1}^{\mathrm{T}}\mathbf{z} \\ \text{s.t.}\ \ -M\mathbf{z} \leq \mathbf{x} \leq M\mathbf{z} \\ \qquad \mathbf{x} \in \mathbf{R}^n, \mathbf{z} \in \{0,1\}^n \end{cases}$$

```python
import cvxpy as cp
from sklearn.datasets import make_regression

# Generate sparse regression data
A, y = make_regression()

# Define variables
n = A.shape[1]
x = cp.Variable(n)
z = cp.Variable(n, boolean=True)

# Define objective and constraints
objective = cp.Minimize(
  cp.sum_squares(A @ x - y) + 10 * cp.sum(z)
)
constraints = [-M * z <= x, x <= M * z]

# Solve the problem using Gurobi
problem = cp.Problem(objective, constraints)
problem.solve(solver=cp.GUROBI)
```

MIP – Let's sum up

Problem

$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

Pipeline

1) Introduce binary variable

2) Establish MIP formulation

3) Use generic MIP solvers

Pros

✓ Rich MIP literature

✓ Black-box solvers

✓ Convenient for practitioners

Cons

✗ Mostly commercial solvers

✗ Unable to exploit structure

✗ Performance issues

Branch-and-Bound Algorithms

BnB – Pipeline

Application

ML, Stats, Signal, Operation Research, ...

$\downarrow$

Problem

$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

$\downarrow$

Problem solutions

Specialized BnB solver

`sbnb` $\rightarrow$ G. Samain *et al.* (2020)

`l0bnb` $\rightarrow$ H. Hazimeh *et al.* (2021)

`el0ps` $\rightarrow$ T. Guyard *et al.* (2024)

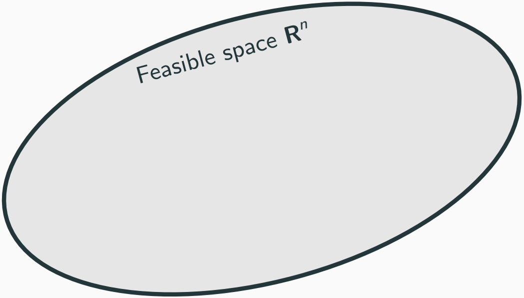
Why using `el0ps` ? [blabla]

BnB – Algorithmic principle

Explore regions in the feasible space and prune those that cannot contain any optimal solution.

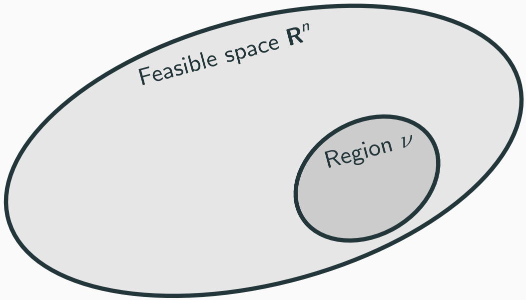
BnB – Algorithmic principle

Explore regions in the feasible space and prune those that cannot contain any optimal solution.



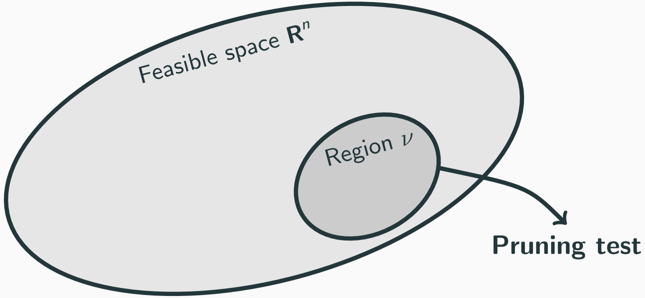
BnB – Algorithmic principle

Explore regions in the feasible space and prune those that cannot contain any optimal solution.



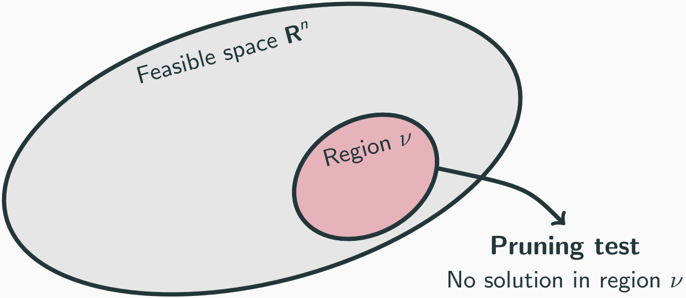
BnB – Algorithmic principle

Explore regions in the feasible space and prune those that cannot contain any optimal solution.



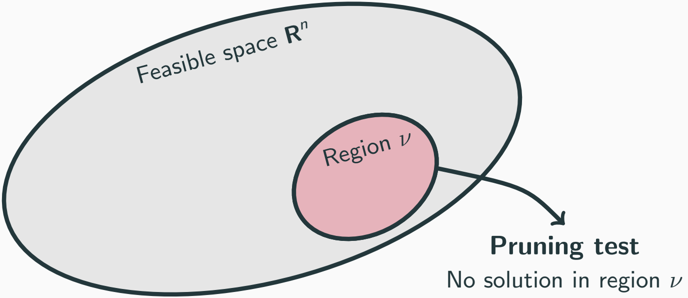
BnB – Algorithmic principle

Explore regions in the feasible space and prune those that cannot contain any optimal solution.



BnB – Algorithmic principle

Explore regions in the feasible space and prune those that cannot contain any optimal solution.

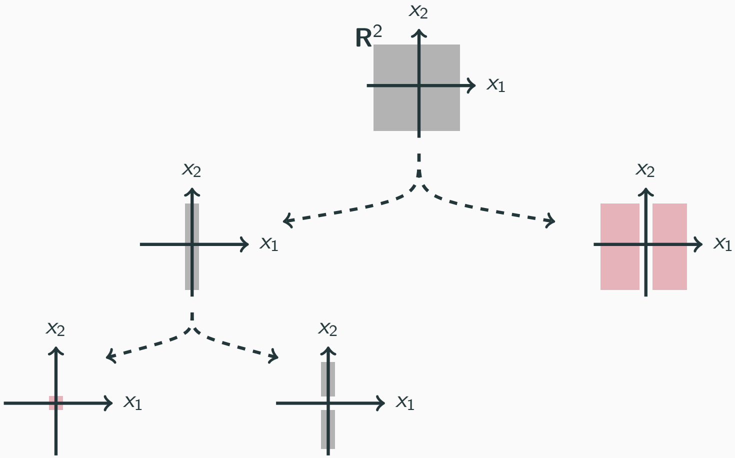


Branching step – Region design and exploration
Bounding step – Pruning test evaluation

BnB – Branching step

Problem

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$



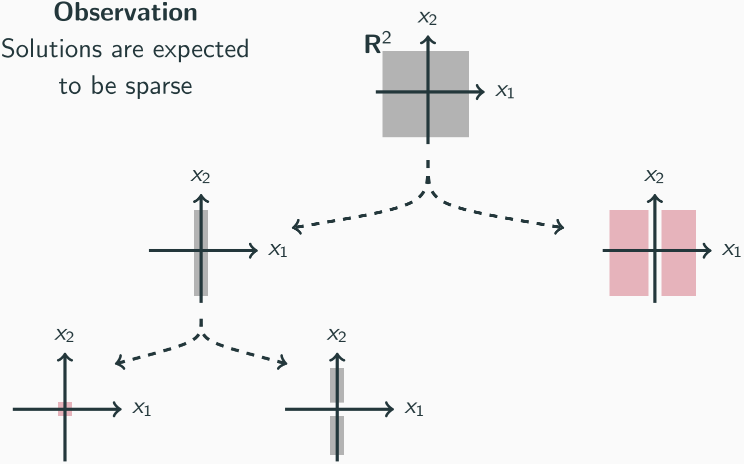
BnB – Branching step

Problem

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

Observation

Solutions are expected to be sparse



Problem

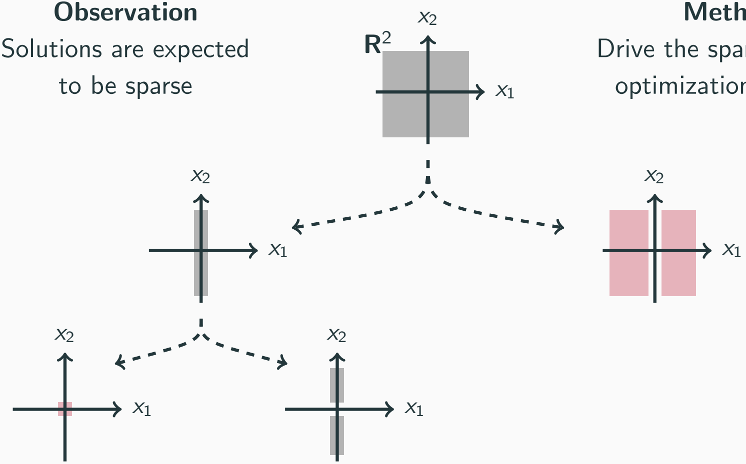
$$\min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

Observation

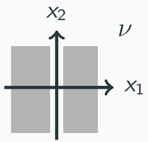
Solutions are expected to be sparse

Method

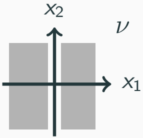
Drive the sparsity of the optimization variable



BnB – Bounding step



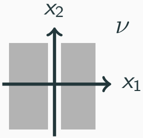
BnB – Bounding step



Problem

$$p^\star = \min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

BnB – Bounding step



Problem

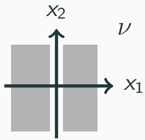
$$p^\star = \min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

restrict to ν

Restriction to region ν

$$p^\nu = \min_{\mathbf{x}\in\nu} \; f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

BnB – Bounding step



Problem

$$p^\star = \min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

$\downarrow$ restrict to ν

Restriction to region ν

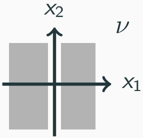
$$p^\nu = \min_{\mathbf{x} \in \nu} \; f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

$\downarrow$ compare

Pruning test

$$p^\nu > p^\star$$

BnB – Bounding step



Problem

$$p^\star = \min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

$\downarrow$ restrict to ν

Restriction to region ν

$$p^\nu = \min_{\mathbf{x}\in\nu} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

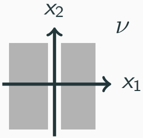
$\downarrow$ compare

Pruning test

$$p^\nu > p^\star$$

$\longrightarrow$ prune ν

BnB – Bounding step



Problem

$p^\star = \min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

restrict to ν

Restriction to region ν

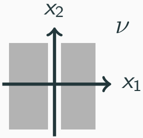
$p^\nu = \min_{\mathbf{x}\in\nu} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

compare

Pruning test

$p^\nu_{\text{lb}} > p^\star_{\text{ub}}$ $\longrightarrow$ prune ν

BnB – Bounding step



Problem

$$p^\star = \min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

$\downarrow$ $\boxed{\text{restrict to } \nu}$

Restriction to region ν

$$p^\nu = \min_{\mathbf{x} \in \nu} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

$\downarrow$ $\boxed{\text{compare}}$

Pruning test

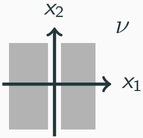
$$p^\nu_{\text{lb}} > p^\star_{\text{ub}}$$

$\longrightarrow$ prune ν

Easy task

$\boxed{\text{Compute an upper bound on } p^\star}$

BnB – Bounding step



Problem

$$p^\star = \min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

restrict to ν

Restriction to region ν

$$p^\nu = \min_{\mathbf{x} \in \nu} \; f(\mathbf{x}) + \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$$

compare

Pruning test

$$p^\nu_{\mathrm{lb}} > p^\star_{\mathrm{ub}}$$

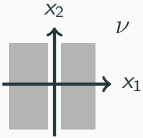
$\longrightarrow$ prune ν

Easy task

$\boxed{\text{Compute an upper bound on } p^\star}$

Construct and evaluate
a feasible vector in each
region explored to refine $p^\star_{\mathrm{ub}}$

BnB – Bounding step



Problem

$p^\star = \min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

restrict to ν

Restriction to region ν

$p^\nu = \min_{\mathbf{x}\in\nu} \; f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

compare

Pruning test

$p^\nu_{\text{lb}} > p^\star_{\text{ub}}$ $\longrightarrow$ prune ν

Easy task

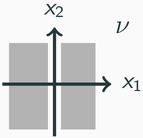
Compute an upper bound on $p^\star$

Construct and evaluate a feasible vector in each region explored to refine $p^\star_{\text{ub}}$

Main challenge

Compute a lower bound on p^ν

BnB – Bounding step



Problem

$p^\star = \min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

restrict to ν

Restriction to region ν

$p^\nu = \min_{\mathbf{x}\in\nu} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

compare

Pruning test

$p^\nu_{\mathrm{lb}} > p^\star_{\mathrm{ub}}$ $\longrightarrow$ prune ν

Easy task

Compute an upper bound on $p^\star$

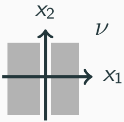
Construct and evaluate
a feasible vector in each
region explored to refine $p^\star_{\mathrm{ub}}$

Main challenge

Compute a lower bound on p^ν

Construct and
solve a relaxation

BnB – Building relaxations



Region $\nu \equiv (\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_\bullet)$ with $\begin{cases} x_i = 0 & \text{if } i \in \mathcal{S}_0 \\ x_i \neq 0 & \text{if } i \in \mathcal{S}_1 \\ x_i \in \mathbf{R} & \text{if } i \in \mathcal{S}_\bullet \end{cases}$

Restriction to region ν

$$p^\nu \;=\; \min_{\mathbf{x} \in \nu} f(\mathbf{x}) + g(\mathbf{x})$$

with $g(\mathbf{x}) = \lambda \|\mathbf{x}\|_0 + h(\mathbf{x})$

$\downarrow$ $\boxed{\text{reformulation}}$

Restriction to region ν

$$p^\nu \;=\; \min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + g^\nu(\mathbf{x})$$

with g^ν proper and closed

$\downarrow$ $\boxed{g^\nu_{\text{lb}} \leq g,\ g^\nu_{\text{lb}} \text{ convex}}$

Relaxation for region ν

$$p^\nu_{\text{lb}} \;=\; \min_{\mathbf{x} \in \mathbf{R}^n} f(\mathbf{x}) + g^\nu_{\text{lb}}(\mathbf{x})$$

set g^ν_{lb} as the convex envelope of g^ν

BnB – Solving relaxations

Relaxation for region ν

$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + g_{\text{lb}}^{\nu}(\mathbf{x})$

with g_{lb}^{ν} proper, closed, convex, and non-differentiable at $\mathbf{x} = \mathbf{0}$

$\updownarrow$

Convex problem

$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \tilde{g}(\mathbf{x})$

lasso-like problem

First-order methods

Proximal gradient

Coordinate descent

Splitting methods

...

Acceleration strategies

Working set

Screening tests

Homotopy

...

iteration

$\mathbf{x}^0 \longrightarrow \boxed{\mathbf{x}^k \to \mathbf{x}^{k+1}} \longrightarrow \mathbf{x}^\star$

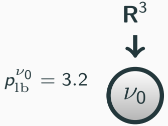
Guarantee of numerical efficiency

BnB – Tree search

Best upper bound

$p^\star_{\mathrm{ub}} = +\infty$

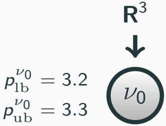
BnB – Tree search



Best upper bound

$$p^\star_{\text{ub}} = +\infty$$

BnB – Tree search



Best upper bound

$$p_{\mathrm{ub}}^{\star} = +\infty$$

BnB – Tree search

$\mathbf{R}^3$

$$\downarrow$$

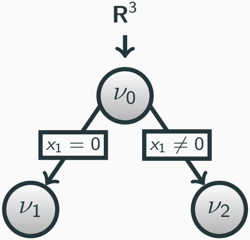
$p_{\mathrm{lb}}^{\nu_0} = 3.2$
$p_{\mathrm{ub}}^{\nu_0} = 3.3$ ν_0

Best upper bound

$$\cancel{p_{\mathrm{ub}}^\star = +\infty}$$

$$p_{\mathrm{ub}}^\star = 5.5$$

BnB – Tree search

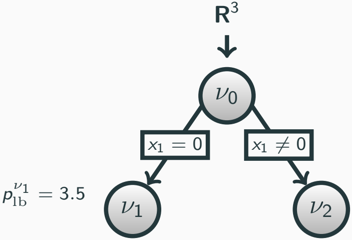


Best upper bound

$\cancel{p^\star_{\mathrm{ub}} = +\infty}$

$p^\star_{\mathrm{ub}} = 5.5$

BnB – Tree search

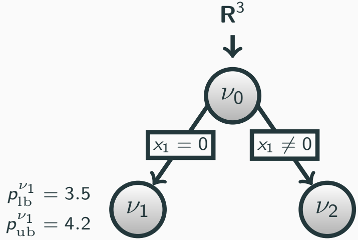


Best upper bound

$\cancel{p_{\text{ub}}^\star = +\infty}$

$p_{\text{ub}}^\star = 5.5$

BnB – Tree search

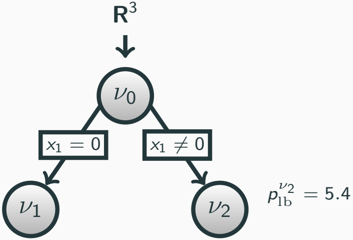


Best upper bound

$\cancel{p_{\mathrm{ub}}^{\star} = +\infty}$

$\cancel{p_{\mathrm{ub}}^{\star} = 5.5}$

$p_{\mathrm{ub}}^{\star} = 4.2$

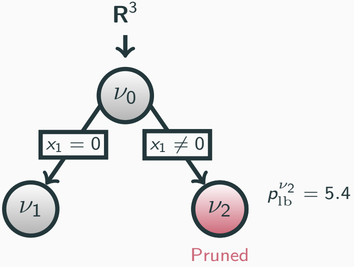


Best upper bound

$$\cancel{p_{\text{ub}}^{\star} = +\infty}$$

$$\cancel{p_{\text{ub}}^{\star} = 5.5}$$

$$p_{\text{ub}}^{\star} = 4.2$$



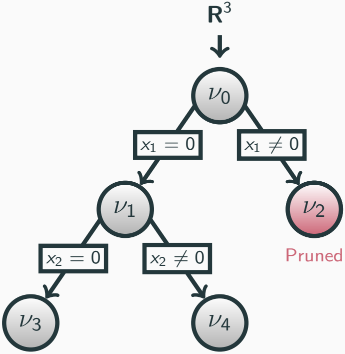
Best upper bound

$\cancel{p_{\mathrm{ub}}^{\star} = +\infty}$

$\cancel{p_{\mathrm{ub}}^{\star} = 5.5}$

$p_{\mathrm{ub}}^{\star} = 4.2$

BnB – Tree search



Best upper bound

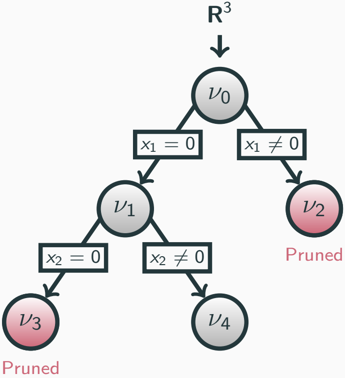
$\cancel{p^\star_{\mathrm{ub}} = +\infty}$

$\cancel{p^\star_{\mathrm{ub}} = 5.5}$

$p^\star_{\mathrm{ub}} = 4.2$

...

BnB – Tree search



Best upper bound

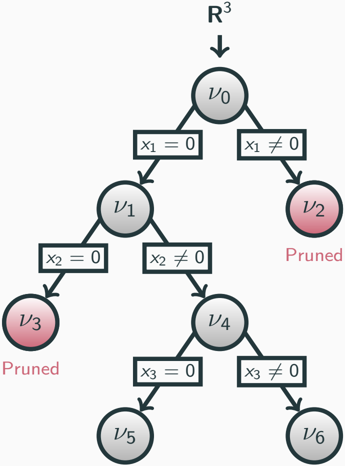
$\cancel{p^\star_{\mathrm{ub}} = +\infty}$

$\cancel{p^\star_{\mathrm{ub}} = 5.5}$

$p^\star_{\mathrm{ub}} = 4.2$

...

BnB – Tree search



Best upper bound

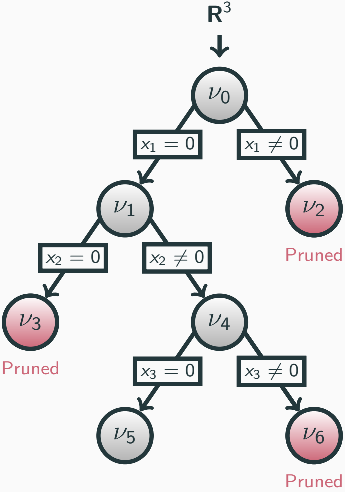
$\cancel{p^\star_{\mathrm{ub}} = +\infty}$

$\cancel{p^\star_{\mathrm{ub}} = 5.5}$

$p^\star_{\mathrm{ub}} = 4.2$

...

BnB – Tree search



Best upper bound

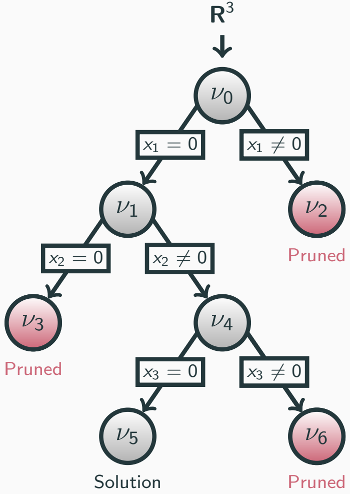
$\cancel{p^\star_{\mathrm{ub}} = +\infty}$

$\cancel{p^\star_{\mathrm{ub}} = 5.5}$

$p^\star_{\mathrm{ub}} = 4.2$

...

BnB – Tree search



Best upper bound

$\cancel{p^\star_{\mathrm{ub}} = +\infty}$

$\cancel{p^\star_{\mathrm{ub}} = 5.5}$

$p^\star_{\mathrm{ub}} = 4.2$

...

BnB – Simultaneous pruning

[Slide]

BnB – Hands-on with `el0ps`

Sparse regression

$$\mathbf{y} \;=\; \mathbf{A}\mathbf{x}^{\dagger} + \epsilon$$

Recover $\mathbf{x}^{\dagger}$ from $(\mathbf{y}, \mathbf{A})$

$\mathbf{x}^{\dagger}$ sparse density ρ

$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$

MAP estimator

$$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$$

- $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$
- $g(\mathbf{x}) = \lambda\|\mathbf{x}\|_0 + \beta\|\mathbf{x}\|_2^2$
- (λ, β) depends on (ρ, σ)

```
$ pip install el0ps
```

```python
from sklearn.datasets import make_regression
from el0ps.datafits import Leastsquares
from el0ps.penalties import L2norm
from el0ps.solvers import BnbSolver

# Generate sparse regression data
A, y = make_regression()

# Instantiate the loss and penalty
f = Leastsquares(y)
h = L2norm(beta=0.1)

# Solve the problem with el0ps’ BnB solver
solver = BnbSolver()
result = solver.solve(f, h, A, lmbd=0.01)
```

BnB – Let's sum up

Problem

$\min_{\mathbf{x}\in\mathbf{R}^n} f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + h(\mathbf{x})$

Pipeline

1) Use specialized BnB

2) Solve the problem

Pros

- ✓ Numerical efficiency
- ✓ Open-source softwares
- ✓ Convenient for practitioners

Cons

- ✗ Assumptions on f/h
- ✗ f/h proper, closed, convex
- ✗ h separable, coercive

Conclusion

People working with ℓ_0-problems

Non-exhaustive list

MIT
D. Bertsimas, R. Mazmuder, ...
MIP tools for ℓ_0-problems

Google Deep Mind
H. Hazimeh, A. Dedieu, ...
MIP-based heuristics

Berkley
A. Atamtürk, A. Gomès, ...
Convex-based acceleration

Lund University
M. Carlsson, C. Olsson...
Quadratic envelope

Frankfurt / Wurzburg Universities
C. Kanzow, A. Tillmann, ...
Optimality conditions

London Business School
J. Pauphilet, R. Cory-Wright, ...
Healthcare applications

Ponts ParisTech
M. De Lara, P. Chancelier, A. Parmentier, ...
Non-convex analysis for ℓ_0-norm, ML appli.

Centrale Nantes / ENSTA Bretagne
S. Bourguignon, J. Ninin, ...
Branch-and-Bound for ℓ_0-problems

Inria / CentraleSupélec
C. Herzet, C. Elvira, A. Arslan, ...
Generalization, acceleration

IRIT / I3S
E. Soubies, L. Blanc-Féraud, ...
Strong relax. of ℓ_0-norm

Take-home messages

- Although NP-hard, ℓ_0-problems are relevant for many applications
- There exists MIP methods to tackle them
  - Generic MIP solvers
  - Specialized BnB algorithms (`el0ps`)
  - Structure-exploitation is key
- It's an active research area
  - Theoretical/methodological developments still needed
  - Help us diffusing to practitioners :)

Question time !

🤓