Summary Notes

- Drivers:
  - ML with heuristic methods
  - Improved MIP and Convex optimization
  - Brings an opportunity for provable optimality / First principles modeling
- Optimization Material

Assume $x_1, x_2, \dots x_n$ and $y_1, y_2, \dots y_n$ are binary decision variables $\{0, 1\}$.

1. Exactly k of $x_1, x_2, \dots x_n$ are equal to 1.
   $x_1 + x_2 + x_3 + \dots + x_n = k$
2. At most k of $x_1, x_2, \dots x_n$ are equal to 1.
   $x_1 + x_2 + x_3 + \dots + x_n \leq k$
3. If $x_1 = 1$, then $y_1 = 1$.
   $x_1 \leq y_1$
4. If at least k of $x_1, x_2, \dots x_n$ equals 1, then $y_1 = 1$.
   $x_1 + x_2 + x_3 + \dots + x_n - (k-1) \leq n * y_1$

Suppose we have a continuous variable a_i,

- How to model $\|\boldsymbol{a}\|_1 \leq 5$?

$$\boldsymbol{e}^\top \boldsymbol{z} \leq 5,\ \boldsymbol{z} \geq \boldsymbol{a}, \boldsymbol{z} \geq -\boldsymbol{a}.$$

- How to model $\|\boldsymbol{a}\|_1 \geq 5$?

$$y_i \leq |a_i|,\ \sum_i y_i \geq 5.$$

That is,

$$\sum_i y_i \geq 5,\ a_i \geq y_i - Mz_i,\ a_i \leq -y_i + M(1 - z_i),\ z_i \in \{0, 1\}.$$

Suppose we have a continuous variable α_i such that if $z_i = 0$ then we must enforce $\alpha_i = 0$. What are two ways to model this?

First way: We add the constraint:
$-Mz_i \leq \alpha_i \leq Mz_i$, where M is a big number.

Second way: if the objective is convex and strong duality holds then we can impose a ridge regularizer, replace α_i with $\alpha_i z_i$, and take the dual.

- True/false: the problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \quad \|\boldsymbol{\beta}\|_1 + \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_1$$

can be written as a linear optimization problem.
  - True. We can write it as

$$\min_{\boldsymbol{\beta}, \boldsymbol{z} \in \mathbb{R}^p, \boldsymbol{r} \in \mathbb{R}^n} \sum_{i=1}^n r_i + \sum_{j=1}^p z_j \text{ s.t. } \boldsymbol{z} \geq \boldsymbol{\beta}, \boldsymbol{z} \geq -\boldsymbol{\beta}, \boldsymbol{r} \geq \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}, \boldsymbol{r} \geq \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}.$$

- True/false: when using the big-M method, the value of M does not affect solve time.
  - False. M affects the quality of the relaxations, which affects the number of nodes expanded.
- True/false: Warm starts are used in mixed integer optimization to improve the computational time required to find an optimal solution.
  - True. Warm starts allow us to obtain a high-quality incumbent quickly, which allows us to prune the tree earlier.

Let x_i be binaries. Model the requirement that if $\sum_i x_i \geq 6$ then $y \leq 7$.
Hint: let z be an indicator variable which denotes whether $\sum_i x_i \geq 6$.
Solution (first version):

$$y \leq 7 + M(1 - z), z = 1 \text{ if } \sum_i x_i \geq 6.$$

Solution (second version):

$$y \leq 7 + M(1 - z), z \geq \min\left(\sum_i x_i, 6\right) - 5.$$

Solution (final version): letting $w = \min\left(\sum_i x_i, 6\right)$: $y \leq 7 + M(1 - z)$ and

$$z \geq w - 5, w \leq \sum_i x_i, w \leq 6, w \geq \sum_i x_i - Mz, w \geq 6 - M(1 - z).$$

- Cutting Planes

Solve the problem: $\min_{\boldsymbol{z} \in \mathcal{Z}} \ f(\boldsymbol{z})$, where f is a convex function.

Algorithm Cutting-plane scheme

```
  Compute $z^1$ via warm-start procedure
  $t \leftarrow 1$
  **repeat**
      Compute $\boldsymbol{z}^{t+1}, \theta^{t+1}$ solution of
          $\min_{\boldsymbol{z} \in \mathcal{Z}, \theta} \ \theta \quad \text{s.t. } \forall s \in \{1, \dots, t\},\ \theta \geq f(\boldsymbol{z}^s) + \nabla f(\boldsymbol{z}^s)^\top (\boldsymbol{z} - \boldsymbol{z}^s)$
      Compute $f(\boldsymbol{z}^{t+1})$ and $\nabla f(\boldsymbol{z}^{t+1})$
      $t \leftarrow t + 1$
  **until** $f(\boldsymbol{z}^t) - \theta^t \leq \varepsilon$
  **return** $\boldsymbol{z}^t$
```

If $\mathcal{Z}$ is a binary set, will we converge? In what sense?

- We converge in a finite number of iterations since there is a finite number of binary points. In the worst case, we would visit all these points.

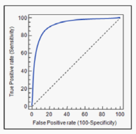
- Any convex function is greater than its Taylor approximation at a given x0.
- So we can min theta where theta is larger than the Taylor approx. at ever point up, 1…t
- We can maintain additional constraints
- We know we are done when the function at a new point is equal to the objective based on cutting planes; if it's not, add a new cutting plane and minimize again

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\bar{y} - y_i)^2}.$$

- y_i's: true values, $\hat{y}_i$'s: predicted values, $\bar{y}$: mean of training set.
- Measures how well the model fits the data compared to a simple baseline model that predicts $\bar{y}$.
- Can evaluate any **regression** model, not just linear regression.
- In-sample: $0 \leq R^2 \leq 1$.
- Out-of-sample: $-\infty < R^2 \leq 1$.

Area Under the Curve (AUC):



- Evaluate any **2-class classification** model.
- It tells how much the model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.
- In-sample: $0.5 \leq AUC \leq 1$.
- Out-of-sample: $0 \leq AUC \leq 1$.

Robust Regression (L2)

$$\min_{\beta} \max_{\mathbf{\Delta} \in \mathcal{U}} \|(\mathbf{y} - (\mathbf{X} + \mathbf{\Delta})\beta\|_q$$

Key Idea:

Robust <= equiv => Regularized

The success regularized regression is because it is a way of robustifying for uncertain input data. Ridge regression results from a robust regression problem with a Frobenius norm uncertainty set and Lasso results from a robust regression problem with an induced norm uncertainty set.

Regularized Regression: q,r in {1,2}

q=r=2: ridge

q=2,r=1: Lasso

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_q + \rho\|\beta\|_r$$

l1 (lasso) – protects against feature-wise perturbations

L2 (ridge) – protects against global perturbations

Theorem

1. *For* $\mathcal{U}_{F(q)} = \{\mathbf{\Delta} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta}\|_{q-F} \leq \rho\}$,

$$\min_{\beta} \max_{\mathbf{\Delta}\in\mathcal{U}_{F(q)}} \|\mathbf{y} - (\mathbf{X}+\mathbf{\Delta})\beta\|_q = \min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_q + \rho\|\beta\|_{q^*},$$

where $\frac{1}{q} + \frac{1}{q^*} = 1$.

2. *For* $\mathcal{U}_{(r,q)} = \{\mathbf{\Delta} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta}\|_{r,q} \leq \rho\}$,

$$\min_{\beta} \max_{\mathbf{\Delta}\in\mathcal{U}_{(r,q)}} \|\mathbf{y} - (\mathbf{X}+\mathbf{\Delta})\beta\|_q = \min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_q + \rho\|\beta\|_r$$

Proof Idea

Focusing on case when $\mathcal{U} = \mathcal{U}_{(r,q)}$ and loss function is ℓ_q.

- Using the norm properties, we have that

$$\|\mathbf{y} - (\mathbf{X}+\mathbf{\Delta})\beta\|_q \leq \|\mathbf{y} - \mathbf{X}\beta\|_q + \|\mathbf{\Delta}\beta\|_q.$$

- Since $\|\mathbf{\Delta}\beta\|_q \leq \|\mathbf{\Delta}\|_{r,q}\|\beta\|_r$ and for $\|\mathbf{\Delta}\|_{r,q} \leq \rho$

$$\|\mathbf{\Delta}\beta\|_q \leq \rho\|\beta\|_r.$$

- Thus,

$$\|\mathbf{y} - (\mathbf{X}+\mathbf{\Delta})\beta\|_q \leq \|\mathbf{y} - \mathbf{X}\beta\|_q + \rho\|\beta\|_r.$$

- We can select a $\mathbf{\Delta}^0 \in \mathcal{U}$ such that

$$\|\mathbf{y} - (\mathbf{X}+\mathbf{\Delta}^0)\beta\|_q = \|\mathbf{y} - \mathbf{X}\beta\|_q + \rho\|\beta\|_r.$$

(Check for yourself!)

- Leads to

$$\max_{\mathbf{\Delta}\in\mathcal{U}} \|\mathbf{y} - (\mathbf{X}+\mathbf{\Delta})\beta\|_q = \|\mathbf{y} - \mathbf{X}\beta\|_q + \rho\|\beta\|_r.$$

(4 points) Consider the robust regression

$$\min \max_{\|\delta \boldsymbol{x}_i\|_2 \leq \rho} \|\boldsymbol{y} - (\boldsymbol{X} + \Delta \boldsymbol{X})\beta\|_2,$$

where $\boldsymbol{x}_i$ is the ith row of matrix $\boldsymbol{X}$ and $\boldsymbol{\delta x}_i$ is the uncertainty in the ith row. The equivalent regularized problem is lasso.

- False. Lasso allows feature-wise uncertainty, so the formulation should be by column and not row.

(4 points) Lasso is a sparse regression method.

- False. Lasso is a robust regression method. It does induce some sparsity by construction but it is not a sparse method by definition.

(a) Lasso always produces sparse solutions. **False.** *Lasso produces partially sparse solutions, but it does not recover the true sparse solution.*

(b) Lasso always produce robust solutions. **True.** *Lasso is equivalent to a robust optimization problem.*

(e) The problem $\min_\beta \|\mathbf{y} - \mathbf{X}\beta\|_1 + \rho\|\beta\|_1$ can be written as a linear optimization problem. **True.** *We can linearize the 1-norm penalty terms in the objective function by adding in auxiliary variables and linear constraints.*

Total Deviations: (yields ridge with q and q* = 2)

Frobenius norm sets:

$$\mathcal{U}_{F(q)} = \{\mathbf{\Delta} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta}\|_{q-F} \leq \rho\},$$

where $\|\mathbf{\Delta}\|_{q-F} := \left(\sum_{ij} |\Delta_{ij}|^q\right)^{1/q}$.

$$\mathcal{U} = \mathcal{U}_{F(2)} = \{\mathbf{\Delta} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta}\|_{2-F} \leq \rho\} \longrightarrow \textstyle\sum_{ij} \Delta_{ij}^2 \leq \rho^2.$$

Feature-wise Perturbation: (yields lasso with q=2, r=1)

Induced norm sets:

$$\mathcal{U}_{(r,q)} = \{\mathbf{\Delta} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta}\|_{r,q} \leq \rho\},$$

where $\|\mathbf{\Delta}\|_{r,q} := \max_{\mathbf{x}} \frac{\|\mathbf{\Delta x}\|_q}{\|\mathbf{x}\|_r}$.

$$\mathcal{U} = \mathcal{U}_{(1,2)} = \{\mathbf{\Delta} \in \mathbb{R}^{n\times p} \mid \|\mathbf{\Delta}\|_{1,2} \leq \rho\} = \{\mathbf{\Delta} \mid \|\mathbf{\Delta x}\|_2 \leq \rho\|\mathbf{x}\|_1 \;\; \forall \mathbf{x}\}$$

$$\mathcal{U} = \{\mathbf{\Delta} \mid \text{every column } \mathbf{\Delta}_i \text{ has } \|\mathbf{\Delta}_i\|_2 \leq \rho\}$$

Stable Regression (L3)

$$\min_{\beta} \max_{z \in \mathrm{conv}(\mathcal{Z})} \sum_{i=1}^{n} z_i |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} \Gamma(\beta_i)$$

s.t.

$$\mathrm{conv}(\mathcal{Z}) = \left\{ z : \sum_{i=1}^{n} z_i = k \,, 0 \le z_i \le 1 \right\}$$

Key Idea:
We can improve on the traditional training / validation split by deliberately finding the hardest training subset (still randomizing the testing data). This is in comparison to a validation method with a single subset (not necessarily better than a cross-fold). It improves both the accuracy and the false alarm rate (for betas that shouldn't be in the support).

Inner Problem

$$\max_{z} \sum_{i=1}^{n} z_i |y_i - x_i^T \beta|$$

s.t.

$$\sum_{i=1}^{n} z_i = k$$

$$0 \le z_i \le 1$$

$\longrightarrow$

Dual

$$\min_{\theta, u_i} k\theta + \sum_{i=1}^{n} u_i$$

s.t.

$$\theta + u_i \ge |y_i - x_i^T \beta|$$
$$u_i \ge 0$$

$\longrightarrow$

Linearized

$$\min_{\theta, u_i} k\theta + \sum_{i=1}^{n} u_i$$

s.t.

$$\theta + u_i \ge y_i - x_i^T \beta$$
$$\theta + u_i \ge -y_i + x_i^T \beta$$
$$u_i \ge 0$$

$\longrightarrow$

Combined

$$\min_{\beta, \theta, u_i} k\theta + \sum_{i=1}^{n} u_i + \lambda \sum_{i=1}^{p} \Gamma(\beta_i)$$

s.t.

$$\theta + u_i \ge y_i - x_i^T \beta$$
$$\theta + u_i \ge -y_i + x_i^T \beta$$
$$u_i \ge 0$$

"Interpreting the Hardest Training Set"

- Stable regression and then use 0/1 as "easy" / "hard" labels
- Decision tree
- What makes a subset hard?

Sparse Linear Regression (L4)

$$y = X\beta + \epsilon, \quad X \in \mathbb{R}^{n\times p}$$

$$\min_{\beta} \ \frac{1}{2}\|y - X\beta\|_2^2 \ \text{ subject to } \ \|\beta\|_0 \le k$$

Key Idea:
We can get a sparse solution by modeling it explicitly (first principles).

Inner Problem

$$\begin{aligned} \min_{\beta} \quad & \tfrac{1}{2}\|y - X\beta\|_2^2 + \tfrac{1}{2\gamma}\|\beta\|_2^2 \\ \text{s.t.} \quad & \|\beta\|_0 \le k, \end{aligned} \longrightarrow$$

Re-define Based on Sparse

Rewrite $\beta_i \to \beta_i s_i$. Define $S = \mathit{diagonal}(s)$.
$S_k := \{s \in \{0,1\}^p \ : \ e's \le k\}$

New Problem

$$\min_{s\in S_k}\left[\min_{\beta\in\mathbb{R}^p} \ \frac{1}{2}\|y - XS\beta\|_2^2 + \frac{1}{2\gamma}\sum_{i=1}^{p} s_i\beta_i^2\right] \longrightarrow$$

Single Problem, Binary Convex Opt

$$\begin{aligned} \min \quad & c(s) = \frac{1}{2}y'\left(\mathbb{I}_n + \gamma \textstyle\sum_j s_j K_j\right)^{-1} y \\ \text{s.t.} \quad & s \in S_k. \end{aligned} \qquad K_j := X_j X_j'$$

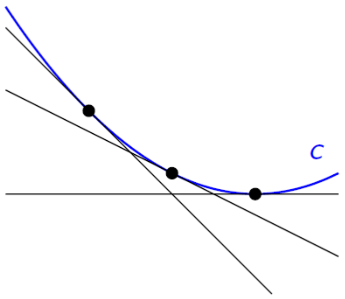
How to Minimize c(s)?
- Cutting planes

By convexity of c, for any $s, \bar{s} \in S_k$,

$$c(s) \ge c(\bar{s}) + \langle \nabla c(\bar{s}), s - \bar{s}\rangle$$

Therefore,

$$c(s) = \max_{\bar{s}\in S_k} c(\bar{s}) + \langle \nabla c(\bar{s}), s - \bar{s}\rangle$$



Convex (and Median) Regression (L5)

$$\min_{\boldsymbol{\theta},\{\boldsymbol{\xi}_i\}_{i\in[n]}} \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta_i)^2+\frac{1}{2\gamma}\sum_{i=1}^{n}\|\boldsymbol{\xi}_i\|^2$$
$$\text{s.t. } \theta_i+\boldsymbol{\xi}_i^T(\boldsymbol{x}_j-\boldsymbol{x}_i)\le\theta_j, \qquad \forall i,j\in[n],$$
$$\boldsymbol{\theta}\in\mathbb{R}^n,$$
$$\boldsymbol{\xi}_i\in\mathbb{R}^p, \qquad \forall i\in[n].$$

Key Idea:
Convex regression solves for a piece-wise convex function that best fits the data by solving for the points and slopes of each of the pieces. Implemented with constraint generation. LQS is a useful, robust-to-outliers technique that we can model using MIO and solve with good bounds.

- Given $(\boldsymbol{x}_i, y_i)$, $i\in[n]$, find a convex function $f:\mathbb{R}^p\to\mathbb{R}$ by solving

$$\min_{f\in\mathcal{C}} \frac{1}{2}\sum_{i=1}^{n}(y_i-f(\boldsymbol{x}_i))^2.$$

- Convexity of f implies that, for any pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$,

$$f(\boldsymbol{x}_i)+\xi_i^T(\boldsymbol{x}_j-\boldsymbol{x}_i)\le f(\boldsymbol{x}_j),$$

where $\boldsymbol{\xi}_i$ is a subgradient of f at $\boldsymbol{x}_i$.

- The variables θ_i represent the values of $f(\boldsymbol{x}_i)$.
- Given $(\theta_i,\boldsymbol{\xi}_i)$, $i\in[n]$, we recover the convex piecewise linear function

$$f(\boldsymbol{x}) := \max_{i\in[n]} \left\{\theta_i+\boldsymbol{\xi}_i^T(\boldsymbol{x}-\boldsymbol{x}_i)\right\}.$$

Cutting planes

$$\min_{\boldsymbol{\theta},\{\boldsymbol{\xi}_i\}_{i\in[n]}} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\theta}\|^2+\frac{1}{2\gamma}\sum_{i=1}^{n-1}\|\boldsymbol{\xi}_i\|^2$$
$$\text{s.t. } \theta_{i_1}+\boldsymbol{\xi}_{i_1}^T(\boldsymbol{x}_{i_2}-\boldsymbol{x}_{i_1})\le\theta_{i_2},$$
$$\theta_{i_2}+\boldsymbol{\xi}_{i_2}^T(\boldsymbol{x}_{i_3}-\boldsymbol{x}_{i_2})\le\theta_{i_3},$$
$$\vdots$$
$$\theta_{i_{n-1}}+\boldsymbol{\xi}_{i_{n-1}}^T(\boldsymbol{x}_{i_n}-\boldsymbol{x}_{i_{n-1}})\le\theta_{i_n},$$
$$\theta_{i_n}+\boldsymbol{\xi}_{i_n}^T(\boldsymbol{x}_{i_{n+1}}-\boldsymbol{x}_{i_n})\le\theta_{i_{n+1}}.$$

$$\longrightarrow$$

Given solution $\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\xi}}_1,\dots,\hat{\boldsymbol{\xi}}_n$ to the reduced master problem:

- Find $j(i)=\arg\max_{k\in[n]}\left\{\hat{\theta}_i-\hat{\theta}_k+\hat{\boldsymbol{\xi}}_i^T(\boldsymbol{x}_k-\boldsymbol{x}_i)\right\}$.
- Check if the corresponding largest value exceeds a given tolerance ϵ.
- If this is the case, we add the constraint

$$\theta_i+\boldsymbol{\xi}_i^T(\boldsymbol{x}_{j(i)}-\boldsymbol{x}_i)\le\theta_{j(i)}$$

  to the reduced master problem for each i, and re-solve.
  Let T_k be the set index pairs of the violated constraints we add in the kth iteration. At the $(k+1)$th iteration we solve

$$\min_{\boldsymbol{\theta},\{\boldsymbol{\xi}_i\}_{i\in[n]}} \frac{1}{2}\|\mathrm{y}-\boldsymbol{\theta}\|^2+\frac{1}{2\gamma}\sum_{i=1}^{n}\|\boldsymbol{\xi}_i\|^2$$
$$\text{s.t. } \theta_i+\boldsymbol{\xi}_i^T(\boldsymbol{x}_j-\boldsymbol{x}_i)\le\theta_j, \qquad (i,j)\in T_0\cup T_1\cup\cdots\cup T_k.$$

- If $\max_{k\in[n]}\left\{\hat{\theta}_i-\hat{\theta}_k+\hat{\boldsymbol{\xi}}_i^T(\boldsymbol{x}_k-\boldsymbol{x}_i)\right\}\le\epsilon$, $\forall i\in[n]$, then the current solution is optimal .

Least Quantile Regression

Theorem
The LQS problem is equivalent to

$$\min_{\boldsymbol{\beta}} |r_{(q)}| = \min_{\mathcal{I}\in\Omega_q}\left(\min_{\boldsymbol{\beta}} \|\mathrm{y}_{\mathcal{I}}-\mathrm{X}_{\mathcal{I}}\boldsymbol{\beta}\|_\infty\right),$$

where $\Omega_q := \{\mathcal{I}:\ \mathcal{I}\subseteq\{1,\dots,n\}, |\mathcal{I}|=q\}$ *and* $(\mathrm{y}_{\mathcal{I}},\mathrm{X}_{\mathcal{I}})$ *denotes the subsample* $(y_i,\mathrm{x}_i), i\in\mathcal{I}$.

$$\begin{array}{lll}
\displaystyle\min_{\boldsymbol{\beta},\mathrm{r},\gamma,\mathrm{z},\boldsymbol{\mu}} & \gamma & \\
\text{subject to} & \mathrm{r}=\mathrm{y}-\mathrm{X}\boldsymbol{\beta} & \\
& \gamma\ge|r_i|-\mu_i, & i=1\dots,n\\
& M(1-z_i)\ge\mu_i, & i=1,\dots,n\\
& \displaystyle\sum_{i=1}^{n} z_i=q & \\
& \mu_i\ge0, & i=1,\dots,n\\
& z_i\in\{0,1\}, & i=1,\dots,n,
\end{array}$$

Robust Logistic Regression (L6)

$$\max_{\beta,\beta_0}\ \min_{\Delta\mathbf{y}\in\mathcal{U}_y}\ \min_{\Delta\mathbf{X}\in\mathcal{U}_x} -\sum_{i=1}^n \log\left(1+e^{-y_i(1-2\Delta y_i)(\beta^T(\mathbf{x}_i+\Delta\mathbf{x}_i)+\beta_0)}\right)$$

$$\mathcal{U}_x = \{\Delta\mathbf{X}\in\mathbb{R}^{n\times p} \mid \|\Delta\mathbf{x}_i\|_q \le \rho, i=1,\dots,n\}$$

$$\mathcal{U}_y = \left\{\Delta\mathbf{y}\in\{0,1\}^n \;\middle|\; \sum_{i=1}^n \Delta y_i \le \Gamma\right\}$$

Key Idea:
We can robustify logistic regression to uncertainties in the data and find a robust counterpart to the optimization problem. Similarly, we can treat uncertainty in the labels.

$$\mathcal{U}_x = \{\Delta\mathbf{X}\in\mathbb{R}^{n\times p} \mid \|\Delta\mathbf{x}_i\|_q \le \rho, i=1,\dots,n\}$$

The robust counterpart to Problem (2) is

$$\max_{\beta,\beta_0} -\sum_{i=1}^n \log\left(1+e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)+\rho\|\beta\|_{q^*}}\right),$$

where L_{q^*} is the dual norm of L_q.

$$\mathcal{U}_y = \left\{\Delta\mathbf{y}\in\{0,1\}^n \;\middle|\; \sum_{i=1}^n \Delta y_i \le \Gamma\right\}$$

The robust counterpart to Problem (4) is

$$\begin{aligned}
\max_{\beta,\beta_0}\quad & -\sum_{i=1}^n \log\left(1+e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)}\right)+\Gamma\mu+\sum_{i=1}^n \nu_i \\
\text{s.t.}\quad & \mu+\nu_i \le \log\left(\frac{1+e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)}}{1+e^{y_i(\beta^T\mathbf{x}_i+\beta_0)}}\right) && i=1,\dots,n, \\
& \nu_i \le 0 && i=1,\dots,n, \\
& \mu \le 0.
\end{aligned}$$

Both x, y

The robust counterpart to Problem (6) is

$$\begin{aligned}
\max\quad & -\sum_{i=1}^n \log\left(1+e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)+\rho\|\beta\|_{q^*}}\right)+\Gamma\mu+\sum_{i=1}^n \nu_i \\
\text{s.t.}\quad & \mu+\nu_i \le \log\left(\frac{1+e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)+\rho\|\beta\|_{q^*}}}{1+e^{y_i(\beta^T\mathbf{x}_i+\beta_0)+\rho\|\beta\|_{q^*}}}\right) && i=1,\dots,n, \\
& \nu_i \le 0 && i=1,\dots,n, \\
& \mu \le 0.
\end{aligned}$$

where L_{q^*} is the dual norm of L_q.

Logistic Regression
Baseline to the more prevalent class
Set a threshold

$$P(Y=1)=\frac{1}{1+e^{-(\beta_0+\beta_1x_1+\ldots+\beta_px_p)}}$$

$$\text{Odds}=\frac{P(Y=1)}{P(Y=-1)} \qquad \text{Odds}=e^{\beta_0+\beta_1x_1+\ldots+\beta_px_p} \qquad \text{Logit}=\log\text{Odds}=\beta_0+\beta_1x_1+\ldots+\beta_px_p$$

$$\max_{\beta,\beta_0}\log\prod_{i=1}^n P(Y_i=y_i)=\max_{\beta,\beta_0}\sum_{i=1}^n -\log(1+e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)})$$

Sparse Classification

$$\min_{\boldsymbol{w}\in\mathbb{R}^p,b\in\mathbb{R}} \sum_{i=1}^n \ell\left(y_i,\boldsymbol{x}_i^\top\boldsymbol{w}+b\right) \quad \text{s.t.} \quad \|\boldsymbol{w}\|_0\le k$$

Ordinary least squares loss: $\ell\left(y_i,\boldsymbol{x}_i^\top\boldsymbol{w}+b\right)=\frac{1}{2}(y_i-\boldsymbol{w}^\top\boldsymbol{x}_i-b)^2$.

Logistic loss: $\ell\left(y_i,\boldsymbol{x}_i^\top\boldsymbol{w}+b\right)=\log\left(1+e^{-y_i(\boldsymbol{w}^\top\boldsymbol{x}_i+b)}\right)$.

Hinge loss: $\ell\left(y_i,\boldsymbol{x}_i^\top\boldsymbol{w}+b\right)=\max\left(0,1-y_i(\boldsymbol{w}^\top\boldsymbol{x}_i+b)\right)$.

$$\xrightarrow{\text{Binaries}}$$

$$\min_{\boldsymbol{s}\in\{0,1\}^p} c(\boldsymbol{s}) \text{ s.t. } \boldsymbol{e}^\top\boldsymbol{s}\le k,$$

where:

$$\begin{aligned}
c(\boldsymbol{s}) := \min_{\boldsymbol{w}\in\mathbb{R}^p,b\in\mathbb{R}}\quad & \sum_{i=1}^n \ell\left(y_i,\boldsymbol{x}_i^\top\boldsymbol{w}+b\right)+\frac{1}{2\gamma}\|\boldsymbol{w}\|_2^2 \\
\text{s.t.}\quad & w_i=0 \text{ if } s_i=0,\ \forall i\in[p].
\end{aligned}$$

Theorem 1

Suppose that for $y\in\{-1,1\}$, $\ell(y,\cdot)$ is convex. Then, strong duality holds and

$$c(\boldsymbol{s})=\max_{\alpha\in\mathbb{R}^n:\ \boldsymbol{e}^\top\alpha=0}\left(-\sum_{i=1}^n\hat{\ell}(y_i,\alpha_i)-\frac{\gamma}{2}\sum_{j=1}^p s_j\alpha^\top\boldsymbol{X}_j\boldsymbol{X}_j^\top\alpha\right),$$

where $\hat{\ell}(y,\alpha):=\max_{u\in\mathbb{R}} u\alpha-\ell(y,u)$ is the Fenchel conjugate

See recitation 4 slides; we can then solve the convex c(s) min with cutting planes, but we needed the dual to get there since the dual provided a value and gradient so it gave us a cut

Holistic Regression (L7)

See below.

Key Idea:
We can incorporate all of the first principles into a single MIP for regression that considers them all.

Regression Elements to Consider
- Sparsity
- Selective sparsity (group; limited pairwise multi-collinearity)
- Detecting NL transforms
- Robustness
- Statistical significance
- Low global multicollinearity

Process
- Split data
- Normalize training set: columns with zero mean and unit l2 norm
- Gamma for robustification
- Max pairwise correlation rho
- Set HC as pairs of highly correlated variables
- GS_m for mth set of group-sparse variables
- Set of NL transforms, T_m
- Set sparsity, k
- Selective sparsity (group; limited pairwise multi-collinearity)
- Detecting NL transforms
- Robustness
- Statistical significance
- Low global multicollinearity

$$\min_{\beta,z} \frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|^2 + \Gamma\|\beta\|_1$$

subject to:

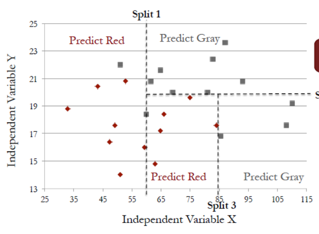
$$\begin{array}{ll}
z_i, b_i \in \{0,1\} & i \in [p] \\
-Mz_i \le \beta_i \le Mz_i, & \text{Big-M constraint } (i \in [p]) \\
\sum_{i=1}^{p} z_i \le k & \text{Sparsity} \\
z_i = z_l & \text{Group Sparsity } (\forall i,j \in GS_m\ \forall m) \\
z_i + z_j \le 1 & \text{Pairwise Collinearity } (\forall i,j \in HC) \\
\sum_{i \in \mathcal{T}_j} z_i \le 1 \quad j \in [p], & \text{Nonlinear transformations}
\end{array}$$

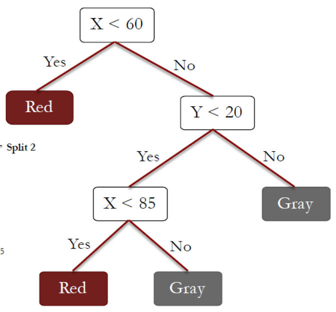
$$\begin{array}{ll}
\frac{\beta_j}{\tilde{\sigma}\sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{jj}}} + Mb_j \ge t_\alpha z_j & \text{Significance } (j = 1, \cdots, p) \\
-\frac{\beta_j}{\tilde{\sigma}\sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{jj}}} + M(1-b_j) \ge t_\alpha z_j & \text{Significance } (j = 1, \cdots, p) \\
\sum_{i \in Supp(\mathbf{a})} z_i \le |Supp(\mathbf{a})| - 1 & \forall \text{ multicollinear relations } \mathbf{a} \\
 & \text{identified by MC-detection Algorithm.}
\end{array}$$

CART, Rand. Forests, Boosted Trees (L8)

Key Idea:
We can classify or regress with trees, which are extremely interpretable. The accuracy can be improved in ensemble, either with RF or gradient boosting, but we lose the interpretability.

CART



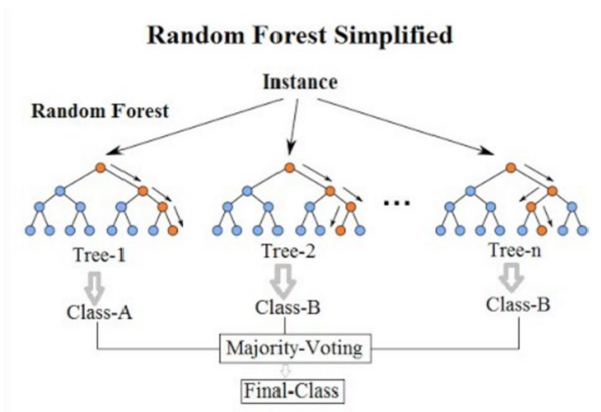


Let us now look at the loss function within a certain node. Say we have $1, \cdots, K$ classes, and the probability of being class i in this node is p_i. Then the three losses are:

- Misclassification Loss $$1 - \max_{i\in\{1,\cdots,K\}} p_i$$
- Gini Impurity $$\sum_{i=1}^{K} p_i(1-p_i)$$
- Entropy $$\sum_{i=1}^{K} -p_i \log(p_i)$$

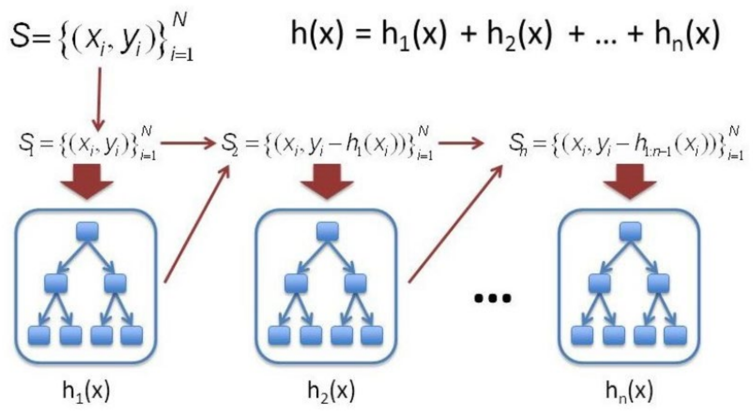
1. Start with the empty tree that always predict the majority class.
2. Pick a leaf node. Construct a candidate split.
3. For every variable in the state, x_i, find the optimal split point.
4. Loop over all variables to find the optimal variable split.
5. Check if the optimal variable split reduces the loss loss$(\cdot)$ by at least γ. If not, pick a new leaf node and repeat.
6. Stop when there are no more γ-level improvements possible.

Random Forests



1. Each tree can split on only a random subset of the variables
2. Each tree is built from a "bagged"/"bootstrapped" sample of the data
   - Select observations randomly with replacement
   - Example – original data: 1 2 3 4 5
   - New "data":
     1. 2 3 1 2 5
     2. 3 1 4 5 1
     3. 4 4 2 1 5

Gradient Boosted Trees



$$S=\{(x_i, y_i)\}_{i=1}^{N} \qquad h(x) = h_1(x) + h_2(x) + \ldots + h_n(x)$$

$$S_1=\{(x_i, y_i)\}_{i=1}^{N} \longrightarrow S_2=\{(x_i, y_i - h_1(x_i))\}_{i=1}^{N} \longrightarrow S_n=\{(x_i, y_i - h_{1:n-1}(x_i))\}_{i=1}^{N}$$

Optimal Classification Trees (L9-10)

Key Idea:
CART is a greedy method, and strong splits can be hidden behind weak splits. We might never find the best solution. OCT models with MIP the selection of the optimal tree. We use a search technique to solve.

CART Adjustments

- **Step 1: Growing**
  - Choose split that locally maximizes an impurity measure
  - Repeat recursively until no more splits possible
  - Grows a deep tree that probably overfits training data
- **Step 2: Pruning**
  - Work back up the tree and remove splits that do not reduce misclassification error by at least α
- Prevents a weak split hiding a strong split

- Trees grown to optimize impurity measure won't necessarily optimize misclassification error

Local Search for OCT: But the MIO has n*2^D variables – hard. The key variables are a and b because they give the splits (hyperplane * x < #). So the real decision space is a,b. Enter the local search with random restarts to optimize over this reduced search space.

OCT MIO

min Error + Complexity

$$\min \quad \frac{1}{\hat{L}} \sum_{t\in\mathcal{L}} L_t + \alpha \cdot \sum_{t\in\mathcal{B}} d_t$$

$$\text{s.t.} \quad L_t \geq N_t - N_{kt} - n(1 - c_{kt}) \qquad L_t \leq N_t - N_{kt} + n c_{kt}, \qquad k = 1,\dots,K,\ \forall t \in \mathcal{L}$$

Linearizes the mis-class term

$$N_{kt} = \sum_{i=1}^{n} \frac{1}{2}(1 + Y_{ik}) z_{it} \qquad k = 1,\dots,K,\ \forall t \in \mathcal{L}$$

Counts points of label k in leaf t

$$N_t = \sum_{i=1}^{n} z_{it} \qquad L_t \geq 0 \qquad \forall t \in \mathcal{L}$$

$$\mathbf{a}_m^T(\mathbf{x}_i + \epsilon) \leq b_m + (1 + \epsilon_{\max})(1 - z_{it}), \qquad i = 1,\dots,n,\ \forall t \in \mathcal{L},\ \forall m \in \mathcal{P}_t^L,$$

Left-branch ancestors

$$\mathbf{a}_m^T\mathbf{x}_i \geq b_m - (1 - z_{it}), \qquad i = 1,\dots,n,\ \forall t \in \mathcal{L},\ \forall m \in \mathcal{P}_t^R,$$

Right branch ancestors

$$\sum_{t\in\mathcal{L}} z_{it} = 1, \qquad i = 1,\dots,n,$$

Each point to one leaf.

The left and right ancestors enforce that for that point to go to that leaf, then aTx < b on the left splits to get there and aTx >= b on the right splits.

$$\sum_{j=1}^{p} a_{jt} = d_t, \qquad 0 \leq b_t \leq d_t, \qquad \forall t \in \mathcal{B}$$

New binary dt if we split at t; if no split, then bt = 0, and all splits go right – as if no split

$$d_t \leq d_{p(t)}, \qquad \forall t \in \mathcal{B}$$

Right branch ancestors

$$\sum_{i=1}^{n} z_{it} \geq N_{\min} \cdot l_t, \qquad \forall t \in \mathcal{L}$$

Enforces Nmin at leaf

$$l_t \geq z_{it}, \qquad \forall t \in \mathcal{L},\ i = 1,\dots,n.$$

If leaf has at least one point

Binary z_it if point I to leaf t

Binary c_kt if class k to leaf t

- Our previous constraints on the splits were:

$$\sum_{j=1}^{p} a_{jt} = 1, \quad \forall t \in \mathcal{B}$$

$$a_{jt} \in \{0,1\}, \quad j = 1,\dots,p,\ \forall t \in \mathcal{B}$$

These two are needed for OCT;
a_jt is for if feature j is split at node t

- We can simply relax integrality on the a_{jt} and replace sum with norm:

$$\|\mathbf{a}_t\|_1 \leq 1, \quad \forall t \in \mathcal{B}$$

- Introduce binary variable s_{jt} to track if jth feature is used in split t:

$$-s_{jt} \leq a_{jt} \leq s_{jt}$$

- New objective:

$$\min \sum_{t\in\mathcal{L}} L_t + \alpha \cdot \sum_{t\in\mathcal{B}} \sum_{j=1}^{p} s_{jt}$$

This is for OCT-H;
a_jt is the same, and the you can include all the different features in the split if you want.

Neural Nets / OCT-Hs (L11)

For classification:
OCT-H < equiv modeling > NNs

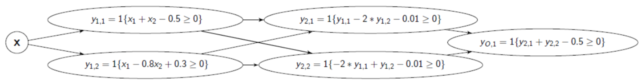
Key Idea:
OCT-Hs are of similar modeling power to deep learning neural nets for classification problems. While not computationally available (to get as deep in a tree to be equivalent) it is an avenue for additional research.

FF w/ Perceptron (:= indicator{x>=0})

- T1 is ind. Of L – only depends on N1
- Output function does not impact T1 construction
- Depth is same as num nodes in 1st hidden layer
- Because perceptron is indicator, the output of the 1st layer is a vector of 0s and 1s with 2^N1 options; so we need 2^N1 depth in the tree
- We then match the output of the tree to the output of the NN to make the equivalent tree

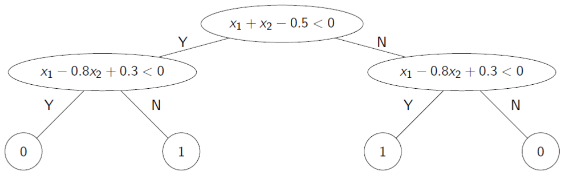
Theorem 1

An OCT-H with maximum depth N_1 can classify the data in a training set at least as well as a given classification FNN with the perceptron activation function and N_1 nodes in the first hidden layer.



$y_{1,1} = 1\{x_1 + x_2 - 0.5 \geq 0\}$
$y_{2,1} = 1\{y_{1,1} - 2 * y_{1,2} - 0.01 \geq 0\}$
x
$y_{O,1} = 1\{y_{2,1} + y_{2,2} - 0.5 \geq 0\}$
$y_{1,2} = 1\{x_1 - 0.8x_2 + 0.3 \geq 0\}$
$y_{2,2} = 1\{-2 * y_{1,1} + y_{1,2} - 0.01 \geq 0\}$

(a)



(b)

- We are given a feedforward neural network $\mathcal{N}_1$ with the following characteristics:
  - The perceptron activation function, defined as $\phi(x) = 1\{x \geq 0\}$
  - Output function $\phi_O(\mathbf{x}) : [0,1]^q \to [0,1]^q$
  - L hidden layers and one output layer, indexed $\ell = 1, \ldots, L, O$
  - N_ℓ nodes in each layer, indexed $i = 1, \ldots, N_\ell$
  - Node $n_{\ell,i}$ defined by $\mathbf{W}_{\ell,i}, b_{\ell,i}$
- Given the inequality from the first node in the first hidden layer of $\mathcal{N}_1$ defined by the weight vector $\mathbf{W}_{1,1}$ and bias scalar $b_{1,1}$, we define the first split of $\mathcal{T}_1$ as

$$\mathbf{W}_{1,1}^T \mathbf{x} + b_{1,1} < 0 \tag{4}$$

- This results in the simple split seen in Figure 5

FF w/ ReLU (:= max(x,0))

- Lots of subtrees concatenated to each other
- 1st subtree just like above (to depth N1)
- But max function means more possibilities so additional trees, one per layer, appended to each branch, repeat – each to depth of that layer
- Then output layer established

An OCT-H with maximum depth $q - 1 + \sum_{\ell=1}^{L} N_\ell$ can classify data in a training set at least as well as a given classification FNN with the rectified linear unit activation function, L hidden layers, N_ℓ nodes in layer $\ell = 1, \ldots, L$, and q nodes in the output layer.

- Each branch of the tree explicitly calculates which output node outputs the highest value (using a lexicographic decision rule in case of ties)
- We can then assign the class associated with that node as the output for the appropriate leaf nodes of $\mathcal{T}_{2,O}(\mathbf{y}_L)$
- Since at each branch we know $\mathbf{y}_L$ as an explicit linear function of $\mathbf{x}$, we also have that $\mathcal{T}_{2,O}(\mathbf{y}_L)$ is a decision tree where all inequalities are explicitly written linear functions of $\mathbf{x}$
- Once we append it to $\mathcal{T}_2$, by construction we have a decision tree that makes the same predictions as $\mathcal{N}_2$
- Since an OCT-H does at least as well as $\mathcal{T}_2$ in classifying the training data, it must do at least as well as $\mathcal{N}_2$ too, completing the proof of the theorem

Prescriptive Trees (L12)

$$z^*(x) \in \arg\min_{z \in \mathcal{Z}} \mathbb{E}\left[c(z;Y) \middle| X = x\right]$$

Key Idea:
Instead of making a point prediction, and then using that point prediction in an optimization, we can take all of the data that would have fed the ML model and use it directly in the optimization.

Instead of: X – ML --> y – Fed to Opt --> Decision, Z
We have: X – Presc Tree --> Decision, Z

$$\hat{z}_N(x) \in \arg\min_{z \in \mathcal{Z}} \sum_{i=1}^{N} w_N^i(x) c(z; y^i)$$

$$\hat{z}_N^{k\text{NN}}(x) \in \arg\min_{z \in \mathcal{Z}} \sum_{x^i \text{ is } k\text{NN of } x} c(z; y^i)$$

Implied binning rule $R(x) = (j \text{ s.t. } x \in R_j)$

$$\hat{z}_N^{\text{CART}}(x) \in \arg\min_{z \in \mathcal{Z}} \sum_{\mathcal{R}(x^i) = \mathcal{R}(x)} c(z; y^i)$$

- Train T trees on bootstrapped samples and randomly selected feature subsets
- Get T binning rules $R^t(x) = (j \text{ s.t. } x \in R_j^t)$

$$\hat{z}_N^{\text{RF}}(x) \in \arg\min_{z \in \mathcal{Z}} \sum_{t=1}^{T} \frac{1}{|\{j : R^t(x^j) = R^t(x)\}|} \sum_{R^t(x^i) = R^t(x)} c(z; y^i)$$

Coefficient of prescriptiveness (P):

$$P = \frac{\min_{z \in \mathcal{Z}} \sum_{i=1}^{N} c(z; y^i) - \sum_{i=1}^{N} c(\hat{z}_N(x^i); y^i)}{\min_{z \in \mathcal{Z}} \sum_{i=1}^{N} c(z; y^i) - \sum_{i=1}^{N} \min_{z \in \mathcal{Z}} c(z; y^i)}.$$

- Objective value of our prescriptive method.
- Objective value of the Sample Average Approximation.
- Objective value of the optimal solution with perfect foresight.
- Measures how well our prescriptive model performs compared to a baseline Sample Average Approximation model which ignores the side information x^i.
- In-sample: $0 \leq P \leq 1$.
- Out-of-sample: $-\infty < P \leq 1$.