Sparse Integer Regression

Hao Liu Keivan Sadeghzadeh Cynthia Rudin

May 2, 2017

Problem

1. Given n data points $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$
2. hope to find a linear relationship between $\mathbf{x}_i$ and y_i, i.e. $y_i \approx \boldsymbol{\beta}^T \mathbf{x}_i$, $\boldsymbol{\beta} \in \mathbb{R}^p$
3. Integer Regression: find $\boldsymbol{\beta} \in \mathbb{Z}^p$, that minimize the L2 loss $\sum_{i=1}^n (y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2$, or $\|Y - X\boldsymbol{\beta}\|^2$, where $Y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$

Ellipsoidal Hypersurface

1. Two forms of an ellipsoid
   - 1. $(X-U)^T Q(X-U) = 1$, $Q = RDR^T$, R is the rotation, D is the diameter, U is the center of X, Q is the transformation matrix
   - 2. $X^T AX + B^T X + C = 0$
2. After solving it, we get $Q = RDR^T = \frac{A}{(B^T A^{-1} B)/4 - C}$

Randomization

- Fact: An ellipsoid is a transformed ball, and the transformation matrix $Q = RDR^T$
- Generating m random points in the unit ball P(0,1)
- Using $E = PDR + U$, we can transform these points to the ellipsoid $E(U, Q)$

Algorithm setup

- Objective: minimize error function over integer coefficient sets inside the high dimension ellipsoidal hypersurface to reach optimal integer solution
- $F = \|Y - X\boldsymbol{\beta}\|^2 = (Y - X\boldsymbol{\beta})^T(Y - X\boldsymbol{\beta})$
  $\rightarrow \boldsymbol{\beta}^T X^T X \boldsymbol{\beta} - 2\boldsymbol{\beta}^T X^T Y + Y^T Y - F = 0$
- We can get an ellipsoid with β on its border and β_{LS} at its center.
- This ellipsoid is like a counter line for the error function. Once we get a β, we can get this ellipsoid, all the β on the border of it have the same error F, and all the possible better solutions lie inside the ellipsoid.
- So we only need to sample inside the ellipsoid and eliminate the solution space outside the ellipsoid.
- Let $A = X^T X$, $B = -2X^T Y$, $C = Y^T Y - F$, we can calculate the $Q = \frac{A}{(B^T A^{-1} B)/4 - C}$, where Q is the transformation matrix of the ellipsoid. By using Q, we can sample in a unit ball, and transform them to the ellipsoid.

Algorithm Description

- We want to minimize the error function $F = \|Y - X\boldsymbol{\beta}\|^2$. And we observe that in the solution space(β space), all the β with the same F lies on a ellipsoid centered at β_{LS}. A bigger F correspond to a ellipsoid with larger size. Once we get a β, we can get this ellipsoid with all the possible better solutions lie inside it. So we can sample inside it, and once we get a better solution, which means a β with smaller F, we can draw another ellipsoid with the same center(β_{LS}), rotation and shape but a smaller size. Then we only need to sample inside this smaller ellipsoid. Continue this procedure, we can eventually get a series of ellipsoid with the same center(β_{LS}), rotation and shape but different sizes and a good integer solution to this problem.

Algorithm Procedure

Algorithm 1: Sparse Integer Regression

Data: $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$

initialization:

1. calculate the least square error solution, $\boldsymbol{\beta}_{LS} = (X^T X)^{-1} X^T Y$;

2. calculate $\tilde{\beta} = round(\boldsymbol{\beta}_{LS})$ as the initial integer solution ;

while *true* **do**

  3. set current best Least Square error $F_{ellip} = F(\tilde{\beta}) = \sum_{i=1}^{n} (Y_i - \tilde{\beta} X_i)^2$;

  4. use $\tilde{\beta}$ to determine the current transformation matrix Q ;

  5. sample M solutions(points) in the unit ball, and transform them to the ellipsoid centered at β_{LS}, with Q as the transformation matrix ;

  6. round the solution ;

  7. calculate each solutions' least square error, and find the solution with the min loss ;

  if *min loss* $<$ F_{ellip}*(current best loss)* **then**

    set $\tilde{\beta}$ equal to that solution ;

  else

    sample again/ enumerate all the remaining solution/ break the loop ;

Future work

- Try more data
- Sparsity
  - Add regularizer into the objective function
  - Back Elimination
- Get a bound on how much space we eliminate if the objective reduce by some percent