Research Update: Rashomon Active Learning

August 13, 2024

Simon Nguyen

Data Generating Process

- 1000 observations $(\mathbf{x}_i, y_i)$ were generated
- X ~ Multivariate Normal($0_{4\times 1}$, Σ) such that

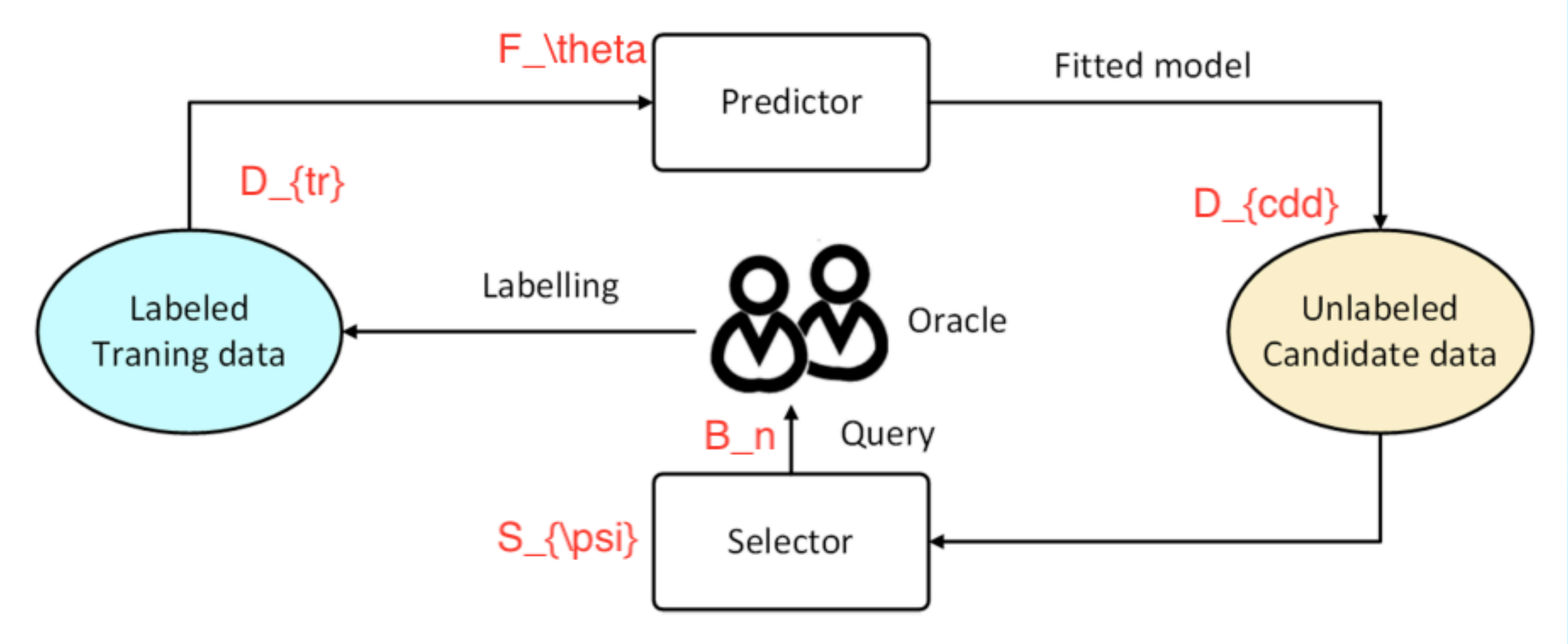
$$\Sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.5 \\ 0 & 0 & 1 & 0 \\ 0 & -0.5 & 0 & 0 \end{pmatrix}$$

- That is, X_2 and X_4 are mildly negatively correlated.

This will affect our LASSO coeffcients estimates

- True coeffcients $\beta_{4\times 1}$ are randomly generated ~ N(0,1).
- Classes generation: $\mathbb{P}(Y = 1) = \dfrac{1}{1 + e^{-\mathbf{x}\beta^T}}$
- Classes 0 and 1 had equal proportion (500 in each class).

Active Learning Set Up



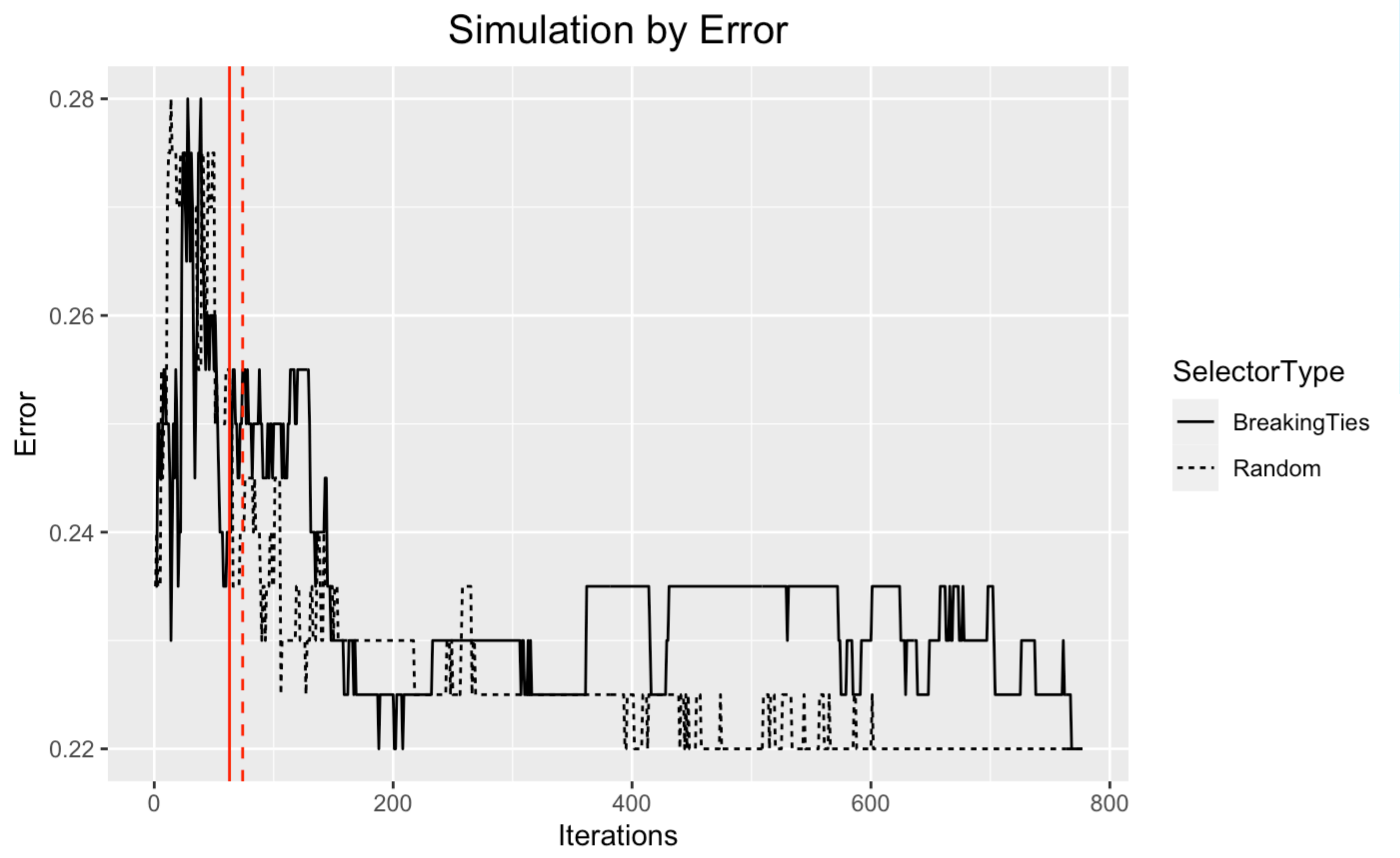
- 80% Training Set, 20% Test Set
- Random Start:
  - Observations are randomly selected for initial oracle labelling (before training the first predictor) until there are at least 10 observations from each class.
- Each selector will select 1 observation for oracle labelling. Unlabeled candidate data are matched based on Mahalanobis Distance.
- The predictor LASSO is then updated/re-trained with the new training set.
- Two types of selectors:
  - Random: Randomly selects observations
  - Breaking Ties: Selects the observations with the most uncertainty:

$$\epsilon_i := \max_{c \in \mathscr{C}} \mathbb{P}(\hat{y}^n_{tr,i} = c \,|\, \mathbf{x}^{(n)}_{tr,i}) - \max_{c \in \mathscr{C} \backslash c^+} \mathbb{P}(\hat{y}^n_{tr,i} = c \,|\, \mathbf{x}^{(n)}_{tr,i})$$

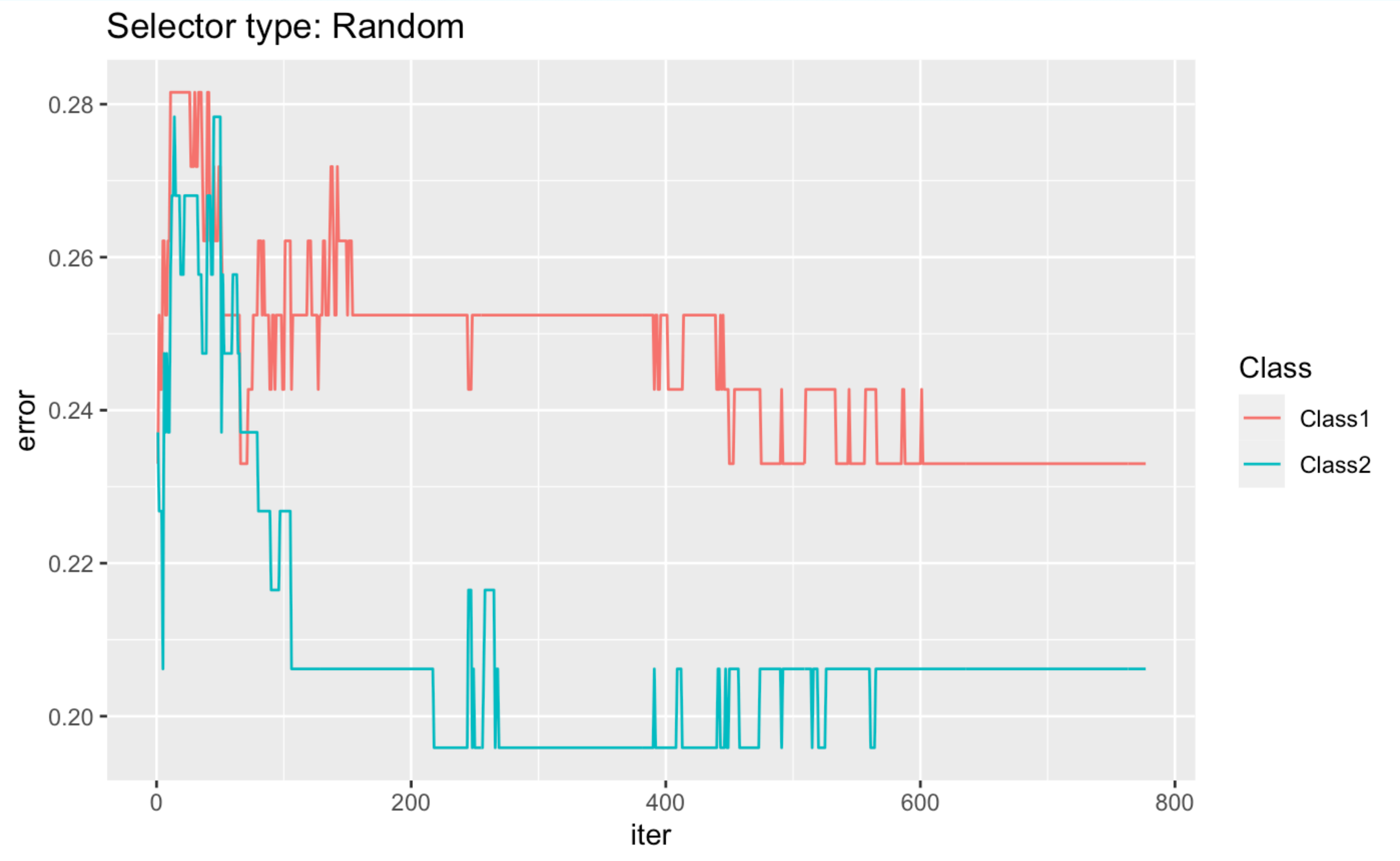
- Stopping Criteria: 10 consecutive iterations have to have an error less than 0.25 and a variance less than 0.2.

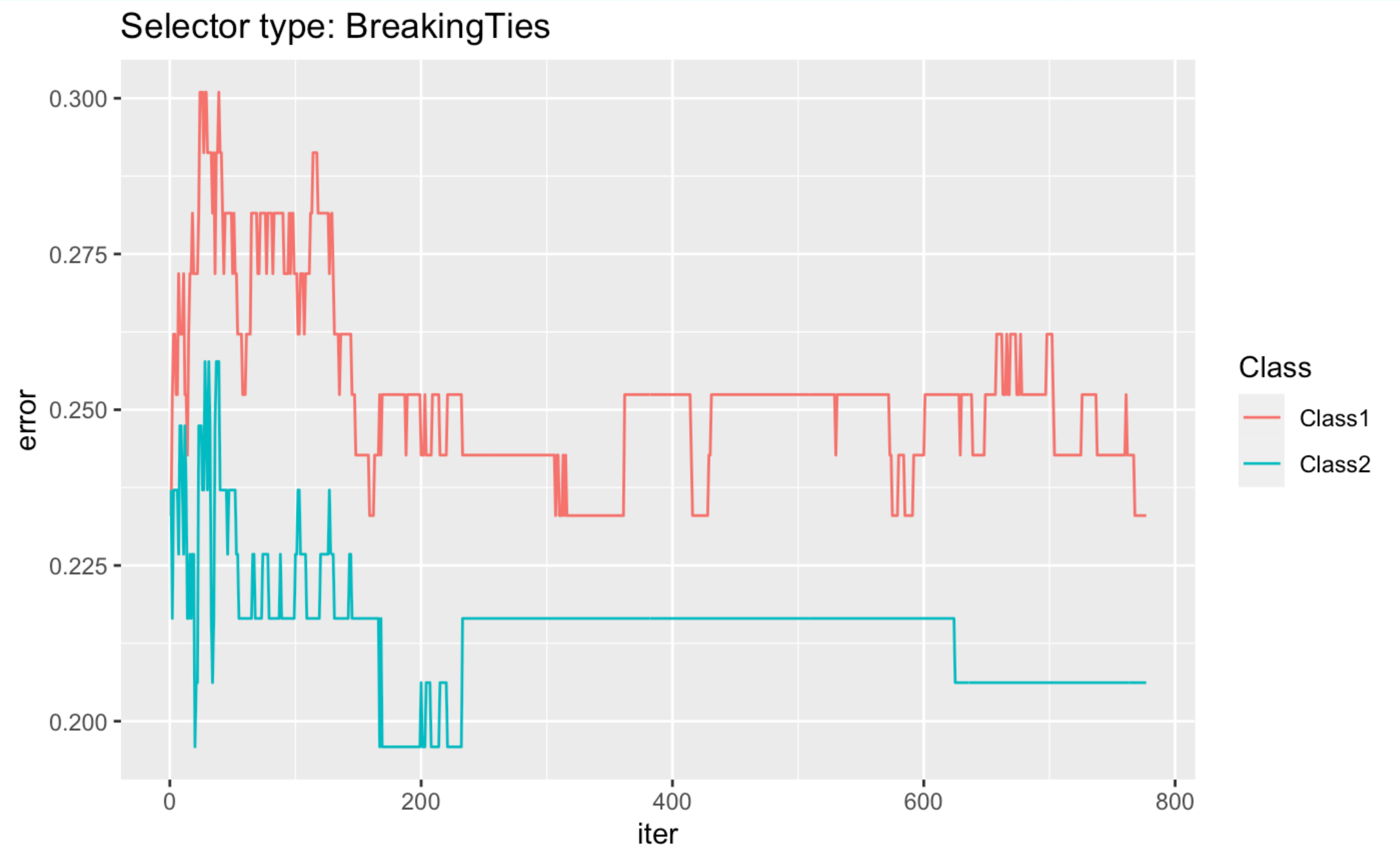
Active Learning Results

- Breaking Ties Selection stops (ever slightly) before Random Selection
  - I can probably change this to induce a larger difference
  - Sometimes Random Selection does better (known in the literature as Cold Start)
- In general, LASSO achieves an accuracy of ~70 - 80%



LASSO Class Error by Selector





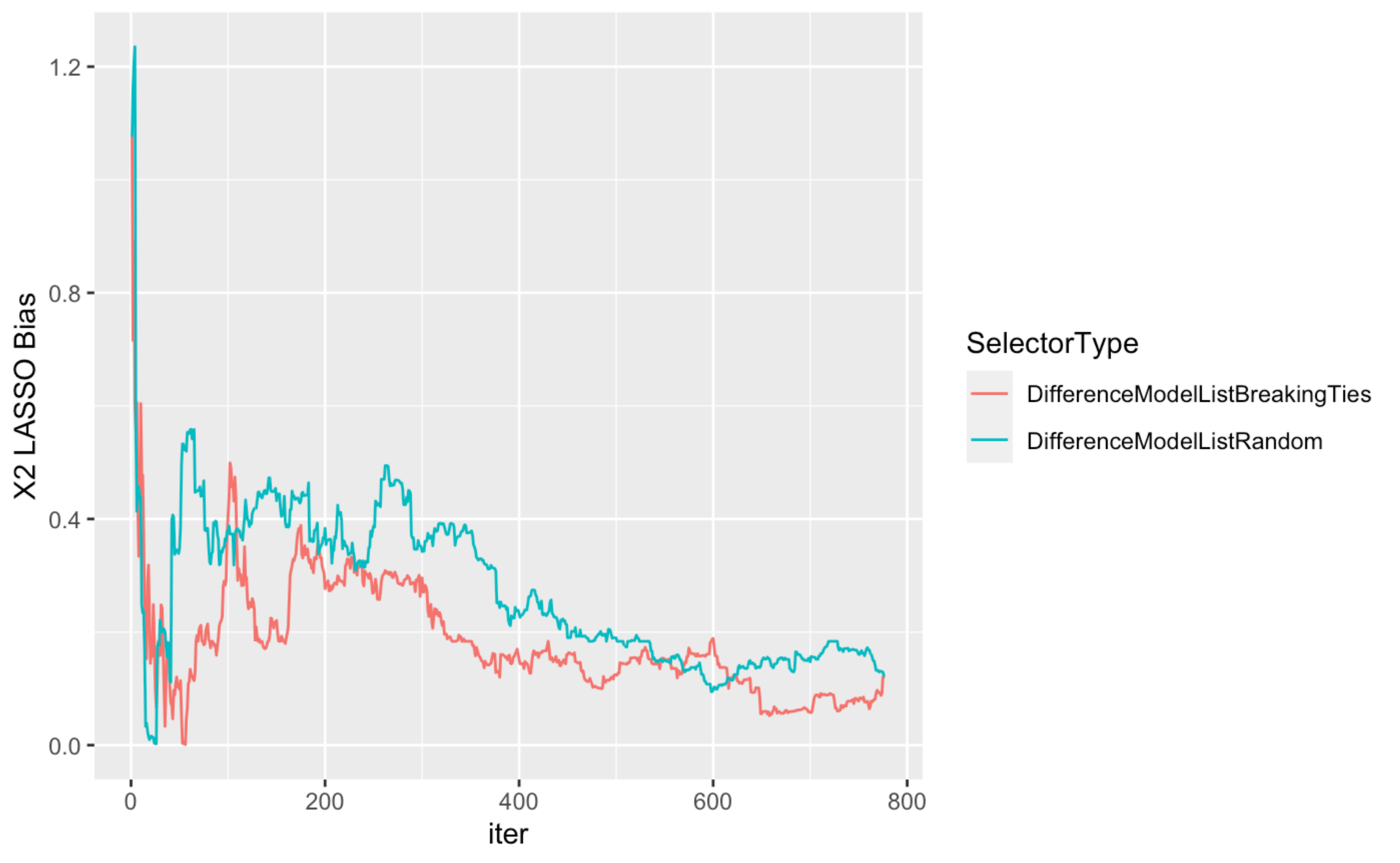
This is cool! Prediction accuracy is decent.

but… LASSO will return bad coefficient estimates when covariates are correlated!

LASSO Biased Coefficient Estimates

- Recall X_2 and X_4 are mildly negatively correlated ($\rho = -0.5$)
- The right table shows that LASSO underestimates X_2 and overestimates X_4.
- The figure shows that even with all samples, LASSO still biases X_2 in the active learning process.

	X1	X2	X3	X4
True	-0.62645	0.18364	-0.83563	1.59528
LASSO	-0.64020	0.06202	-0.78615	1.75861



Takeaway:
Even with decent prediction, LASSO returns biased estimates of regression coefficients in adaptive decision making.

Next Steps

- Extend:

  1. Multi-classes: Easy - should be done this week

  2. Imbalanced samples: Also easy - should be done this week

- Expecting the same results with multi-classes and imbalanced samples:

  - LASSO provides decent prediction

  - But wrong coefficients

- Should set up the framework for considering model uncertainty by enumerating the Rashomon Set

Main Methodology Idea

- At each step of the active learning process, a predictor is retrained based on the updated training set.
- The predictor then provides the probability of an observation being in each class.
- Predicted observation uncertainty is measured (in our case by Breaking Ties):

$$\epsilon_i := \max_{c \in \mathscr{C}} \mathbb{P}(\hat{y}^n_{tr,i} = c \,|\, \mathbf{x}^{(n)}_{tr,i}) - \max_{c \in \mathscr{C} \backslash c^+} \mathbb{P}(\hat{y}^n_{tr,i} = c \,|\, \mathbf{x}^{(n)}_{tr,i})$$

- Unlabelled observations are then matched with the most uncertain observations and then recommended for oracle labelling.
- However, this measure of uncertainty does not consider predictive model uncertainty.
- Instead, we should consider the uncertainty of each observations across the Rashomon Set: $\epsilon_i^{(m)}$ for model m in the Rashomon set
- We then weigh $\epsilon_i^{(m)}$ by the probability of observing each model, say $p(m)$
- Suggestions for oracle labelling is then based on this model-weighted uncertainty metric: $p(m) \cdot \epsilon_i^{(m)}$