Ridge regression

- Shrinkage methods
  - Ridge Regression
  - Model: $\{w_i\}$'s minimize the residual sum of squares:

$$\min_{\{w_i\}} \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{p} w_i x_i)^2 + \alpha \sum_{i=1}^{p} w_i^2 \equiv \min_{\{w_i\}} ||Xw - y||_2^2 + \alpha ||w||_2^2.$$

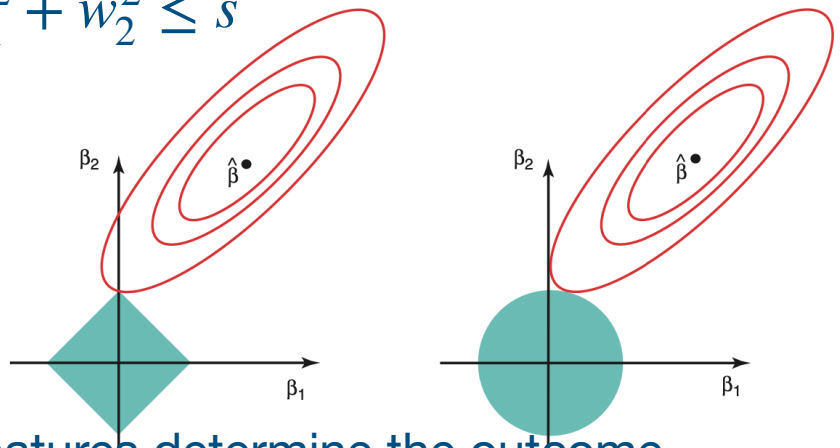
  - α obtained using Cross-Validation.

- Shrinkage penalty applied only on the coefficients w_i, $i = 1 \ldots p$, and not on the intercept w_0.
- Why RR improve over Least Squares (linear regression)?
  - Bias-variance trade-off
  - Small α leads to large variance; large α reduces variance, increases bias

- RR fits models involving all the features; none of the coefficients is zero
  - Even in cases where some of the features may be irrelevant to the target
- LASSO can estimate some coefficients to be exactly zero
  - Thus it performs ***feature selection***, yields ***sparse models***

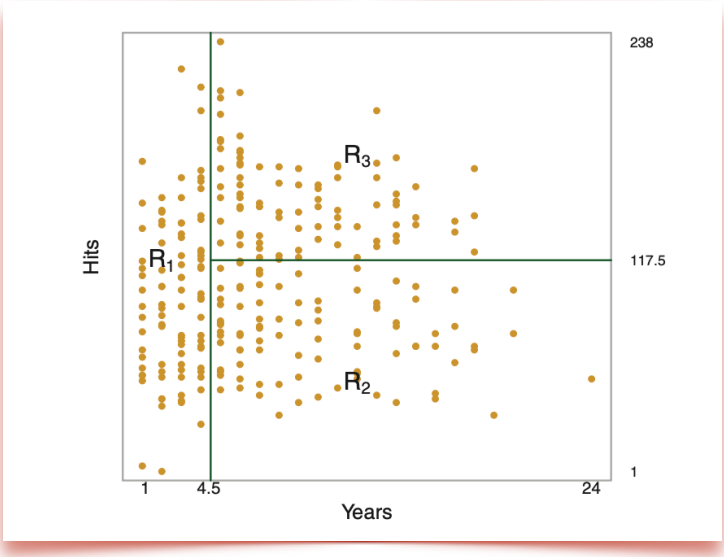
Understanding difference between LASSO & RR

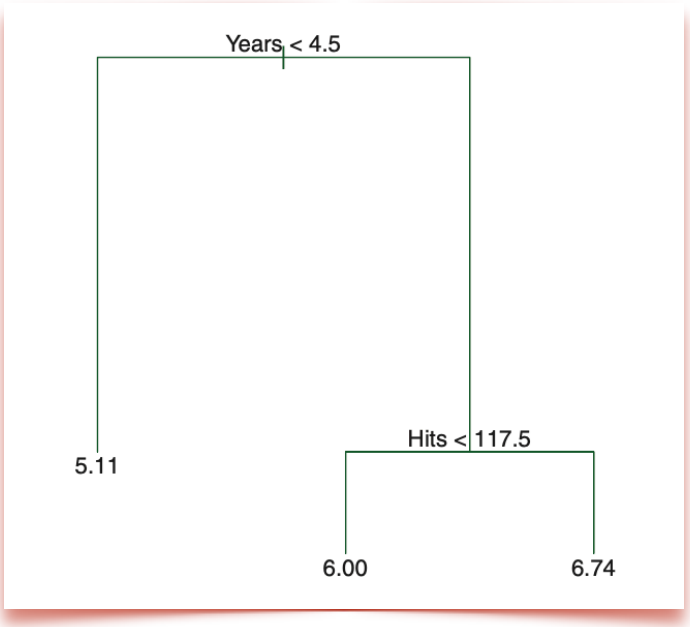
- Alternative formulation for RR and LASSO
  - RR: $\min_{\{w_i\}} \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{p} w_i x_i)^2$ subject to $\sum_{i=1}^{p} w_i^2 \leq s,$
  - LASSO $\min_{\{w_i\}} \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{p} w_i x_i)^2$ subject to $\sum_{i=1}^{p} |w_i| \leq s.$
  - For every α there is a value of s such that these conditions yield the same w_is for RR and LASSO as per the equations on the previous two slides.
  - s is a ***budget*** for the coefficients. For $p = 2$
    - LASSO: $|w_1| + |w_2| \leq s$; RR: $w_1^2 + w_2^2 \leq s$



- LASSO better when a small number of features determine the outcome
- RR depends when it depends on a large number of features

Tree based methods



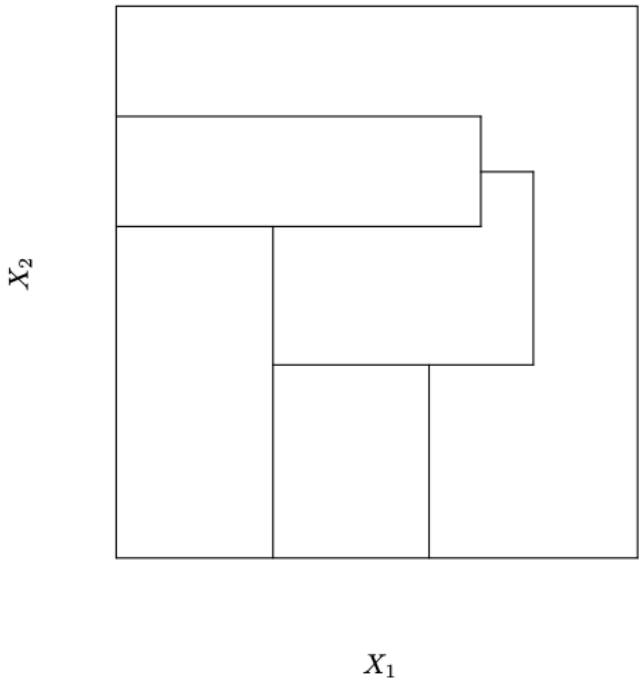


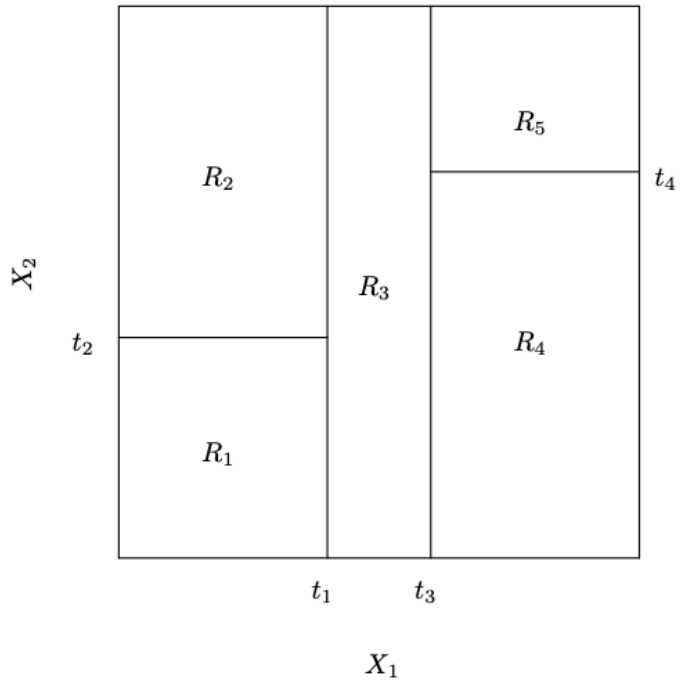
- Regression Trees
  - Salary a function of number of **hits** & number of **years**
  - A series of splitting rules
    - First along the **Years** axis; then along the **Hits** axis
    - $R_1 = \{X \mid \text{Years} < 4.5\}$; $R_2 = \{X \mid \text{Years} \geq 4.5, \text{Hits} < 117.5\}$; $R_2 = \{X \mid \text{Years} \geq 4.5, \text{Hits} \geq 117.5\}$.
    - R_1, R_2, R_3 called **terminal nodes** or **leaves**
    - Points at which feature space is split called **internal nodes**
    - In this example, **Years** is more important in determining salary

- General idea of building a Regression Tree
  1. Divide the feature space into J distinct, non-overlapping regions, $R_1, \ldots R_J$.
  2. For every observation in R_i, prediction is the **simple mean** of the **training observations** in R_i.
- The regions could have any shape.
- But for convenience, they are chosen to be high dimensional boxes so that
- RSS $= \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2$ is minimized

- Creating trees
  - Recursive binary splitting
  - Top-down approach: starts at the *top* of the tree, and successively splits the feature space
  - Choose the feature X_j, and the value s such that splitting the space into regions $\{X \mid X_j < s\}$ and $\{X \mid X_j \geq s\}$ leads to the greatest reduction in RSS.

- Recursive binary splitting leads to complex trees, overfits training data
- Smaller tree, fewer splits may work better for test set: lower variance at the cost of a little higher bias
- Tree pruning
  - Grow a very large tree
  - Prune it to a subtree
  - What is the best subtree? One that gives the lowest test error
  - Can be determined by cross validation

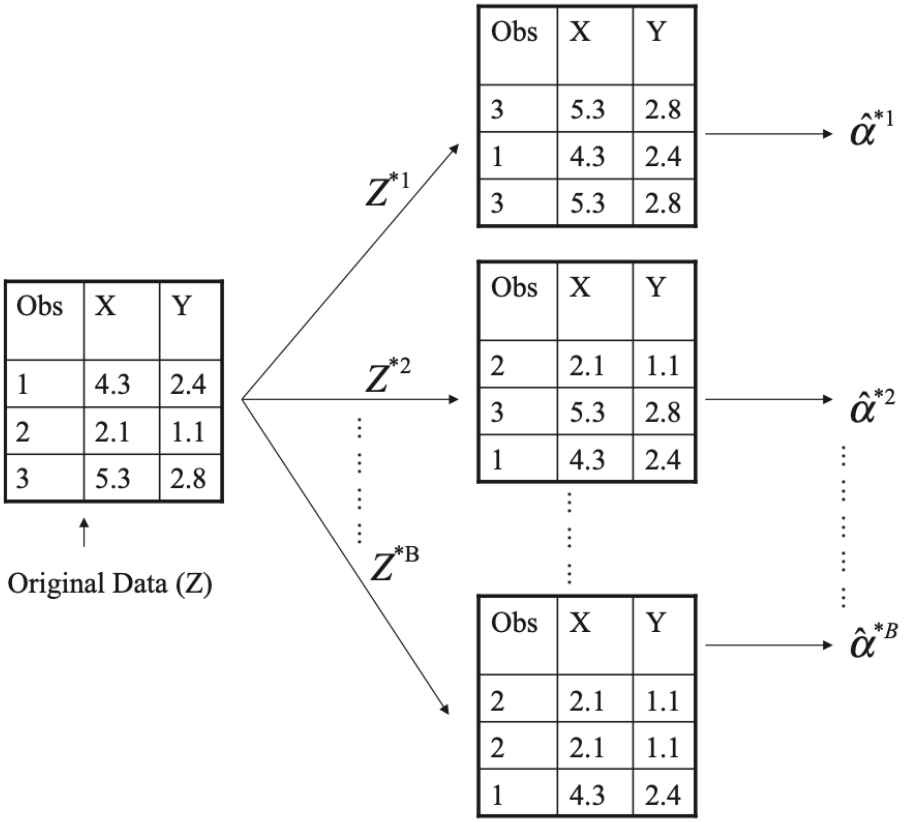




- Classification trees
  - Same idea for splitting along features, and creating regions/boxes
  - What to minimize?
  - Gini index $G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$; K is number of classes; p_{mk} number of training examples in the m-th region from the k-th class, OR
  - Entropy $S = -\sum_{k=1}^{K} p_{mk} \log p_{mk}$

- In general, decision trees have large variance.
  - Take a data set, split into two halves, train to decision trees; these could be very different
  - How to reduce variance?

- Remember bootstrapping



- Draw N training samples of size n from the original training data set
- Consider N independent random variables $Z_1, Z_2, \ldots, Z_N$ each with variance σ^2
- $\bar{Z} = \frac{1}{N}(Z_1 + Z_2 + \ldots + Z_N)$ is a random variable with variance σ^2/N.

- Averaging over many models, variance can be reduced
- If predictions of a quantity f by N tress trained on N different bootstrapped data sets are $\hat{f}^1(x), \hat{f}^2(x), \ldots, \hat{f}^N(x)$, final result is the average
  - $\hat{f}_{\text{bag}}(x) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}^i(x).$
- This is called **bagging**, as applied to regression models
- For classification models, record prediction from each model, take a **majority vote**, or **average of classification probability**

- Random forest
  - Small tweak over bagging which decor relates the trees

- Trees are correlated, particularly if a few of the features dominate splitting
- Variance reduction by $1/N$ applies only to independent random variates, not correlated ones. Therefore, attempt to *decorrelate* the trees.
- Each time, a *random sample of m features* is chosen as candidates for split
  - Typically, $m \approx \sqrt{p}$; most of the features are not even considered for a split at each step

```
class sklearn.ensemble.RandomForestRegressor(n_estimators=100, *,
criterion='squared_error', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=1.0,
max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True,
oob_score=False, n_jobs=None, random_state=None, verbose=0,
warm_start=False, ccp_alpha=0.0, max_samples=None,
monotonic_cst=None)                                          [source]
```