LA Finalproject: Least Angle Regression

Howard Hsieh

December 2024

1 Introduction: Least Angle Regression (LARS)

The Least Angle Regression (LARS) algorithm Efron et al. 2004 is introduced as a computationally efficient method for model selection in linear regression. It selects a parsimonious subset of predictors from a large set for efficient response prediction, maintaining balance between prediction accuracy and simplicity.

1.1 Classic model-selection method

- Forward selection: Hastie et al. 2007

  1. Given a collection of possible predictors, we select the one that has **the largest absolute correlation** with the response y, say x_{j_1}, and perform a simple linear regression of y on x_{j_1}.
  2. At each step of Forward Selection, the residual vector r becomes the new "response" to ensure the predictors already in the model remain uncorrelated with the new predictor being selected.
  3. Orthogonality prevents overfitting by ensuring that each predictor added to the model contributes uniquely to explaining the variance in y.
  4. Forward selection formula: Incrementally selects predictors x_j by maximizing their absolute correlation with the residuals r, updating coefficients as:

  $$\hat{\beta}_j = \frac{x_j^\top r}{x_j^\top x_j}, \quad r = y - X\hat{\beta}.$$

- LASSO:

  1. Lasso applies an l_1-norm penalty to the regression coefficients, encouraging sparsity by setting some coefficients to exactly zero, effectively performing variable selection alongside regularization.
  2. By shrinking coefficients toward zero, Lasso reduces model variance, improving generalization at the cost of introducing bias, which is particularly useful in high-dimensional settings.

3. The constraints $\sum_{j=1}^{m}|\beta_j| \leq t$ (or equivalent penalty $\lambda\sum_{j=1}^{m}|\beta_j|$ in optimization) governs the balance between model complexity and prediction accuracy, where a smaller t (or larger λ) increases sparsity.
4. Lasso formula: Minimizes the residual sum of squares with an ℓ_1-norm penalty:

$$\arg\min_{\beta} \quad \|y - X\beta\|^2 + \lambda\sum_{j=1}^{m}|\beta_j|.$$

1.2 The relationships between these three methods

Both Lasso and Stagewise are variants of a basic procedure called "Least Angle Regression" (LARS) which "S" comes from "Stagewise and Lasso". LARS is a **greedy** and **efficient algorithm** that moves iteratively along an equiangular direction between predictors, adding predictors one by one based on their correlation with the residuals, which will be fully explained in the next section.

2 The LARS Algorithm

2.1 Notation

- The space spanned by $(\mathbf{x}_1, \mathbf{x}_2)$ is :Linear space $\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2)$
- Features or predictor or covariates : $\boldsymbol{x}_1, \boldsymbol{x}_2,\ \boldsymbol{x}_j, \boldsymbol{x}_k$
- A candidate vector of regression coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_m)$ gives prediction vector $\hat{\boldsymbol{\mu}}$:

$$\hat{\boldsymbol{\mu}} = \sum_{j=1}^{m} x_j\hat{\beta}_j = \mathbf{X}\hat{\beta} \quad [\mathbf{X}_{n\times m} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)]$$

- Residual: $\quad \boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{\mu}}$
- The vector of *current correlations*, which means the correlation between covariates and residuals is:

$$\hat{\boldsymbol{c}} = \boldsymbol{c}(\hat{\boldsymbol{\mu}}) = \boldsymbol{X'}(\boldsymbol{y} - \hat{\boldsymbol{\mu}}) = \boldsymbol{X'r}$$

- The active set, containing indices of variables currently included in the model is $\mathcal{A}$
- Equiangular direction vector among active predictors is $\boldsymbol{\mu}_A$

2.2 The LARS procedure

1. **Initialization**:
    - Start with all coefficients are zero, $(\beta_1, \beta_2 ... \beta_m) = 0$.
    - Compute the initial residual $\boldsymbol{r} = \boldsymbol{y}$, which also means $\boldsymbol{\mu_0} = 0$

- Compute the correlations : $\hat{\boldsymbol{c}}_{\boldsymbol{j}} = \boldsymbol{x}'_j \boldsymbol{r}$,for $j = 1, 2, ...m$

2. **Find the First Variable:**

   - Identify the variable x_{j1} with the largest absolute correlation:

   $$j_1 = \arg\max_j |c_j|.$$

   - Add $\mathbf{x}_{j_1}$ to the active set $\mathcal{A}$.

3. **Move in the Direction of $\mathbf{x}_{j_1}$:**

   - Increase the coefficient β_{j_1} in the direction of $\mathbf{x}_{j_1}$ until another variable $\mathbf{x}_{j2}$ has equal correlation with the residual.
   - Update the residual: $\mathbf{r} = \mathbf{y} - \mathbf{X}\beta$.

4. **Equiangular Direction:**

   $$\mathbf{u}_{\mathcal{A}} = \arg\min_{\mathbf{u}} \|\mathbf{r} - \mathbf{X}_{\mathcal{A}}\mathbf{u}\|_2,$$

   where $\mathbf{X}_{\mathcal{A}}$ is the matrix of active variables.

5. **Update Coefficients:**

   $$\beta \leftarrow \beta + \gamma \mathbf{u}_{\mathcal{A}},$$

   where γ is chosen such that a new variable enters the model or an active variable leaves the model.

6. **Add or Remove Variables:**

   - Continue until all variables are added to the model or a stopping criterion (e.g., minimum error) is met.

2.3 Mathematical Formulation

- **Correlation Update:**

  $$\mathbf{c} = \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\beta).$$

  Variables are selected based on the largest absolute correlation.

- **Equiangular Direction:** For the active set $\mathcal{A}$, compute the direction vector $\mathbf{u}_{\mathcal{A}}$ that is equiangular with all predictors in $\mathcal{A}$:

  $$\mathbf{u}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}},$$

  where $\mathbf{w}_{\mathcal{A}} = (\mathbf{X}_{\mathcal{A}}^\top \mathbf{X}_{\mathcal{A}})^{-1}\mathbf{1}$ and $\mathbf{1}$ is a vector of ones.

- **Step Size:** The step size γ is determined as:

  $$\gamma = \min\left\{\frac{c_{j_1} - c_{j_2}}{\mathbf{u}_{\mathcal{A}}^\top \mathbf{X}}, \text{ for all variables not in } \mathcal{A}\right\}.$$

2.4 Properties of LARS

- **Computational Efficiency:** Requires m steps for m predictors, making it computationally similar to ordinary least squares.
- **Connections to Lasso:** A slight modification of LARS can compute all solutions for the Lasso.
- **Connections to Forward Stagewise Regression:** LARS can approximate Forward Stagewise Regression with fewer computational steps.

3 Degrees of freedom and C_p Estimates

3.1 Definition

For linear regression, the concept of degrees of freedom ties closely to the trace of the projection matrix, which provides insights into how well the model generalizes. A more complex model, with higher degrees of freedom, risks overfitting the data, while a simpler model may fail to capture underlying patterns.

In the context of Least Angle Regression (LARS), degrees of freedom provide a straightforward yet powerful way to gauge model complexity, helping to balance prediction accuracy with parsimony. This is particularly relevant when dealing with high-dimensional data, where efficient model selection is critical.

3.2 Degrees of Freedom in LARS

- In Least Angle Regression (LARS), the degrees of freedom provide a measure of model complexity.
- The degrees of freedom for a k-step LARS estimate can be approximated as:

  $$\mathrm{df}(\hat{\beta}_k) \approx k,$$

  where k is the number of steps taken by the algorithm.
- This approximation is accurate because:
  1. Each step in LARS adds one variable to the active set, increasing flexibility by one degree.
  2. The relationship holds particularly well when predictors are orthogonal or satisfy certain geometric conditions.
- The simplicity of this approximation makes it computationally efficient:
  1. No additional calculations are required to estimate the degrees of freedom.
  2. This efficiency is especially useful when calculating model selection criteria like the Cp statistic.

3.3 Cp Risk Estimate

- The Cp statistic is a key metric used to estimate the prediction error of a regression model.

- It balances the trade-off between model fit and complexity, helping to avoid overfitting.

- For Least Angle Regression (LARS), the Cp statistic is calculated as:

  $$Cp(\hat{\beta}_k) = \frac{\|y - X\hat{\beta}_k\|^2}{\sigma^2} - n + 2k,$$

  where:

  1. y is the observed response vector.
  2. X is the matrix of predictors.
  3. $\hat{\beta}_k$ represents the estimated coefficients at step k.
  4. σ^2 is the variance of the error term.
  5. n is the number of observations.
  6. k is the degrees of freedom, which equals the number of steps taken by the LARS algorithm.

- Key components:

  * $\frac{\|y-X\hat{\beta}_k\|^2}{\sigma^2}$: Measures the residual sum of squares normalized by the error variance.
  * $-n + 2k$: A penalty term that adjusts for the model complexity based on the degrees of freedom.

- A model with a lower Cp value indicates better predictive performance, balancing goodness of fit with simplicity.

3.4 Bootstrap Validation

- Degrees of freedom were validated via bootstrap sampling, demonstrating robustness in simulations and practical scenarios.

3.5 Special Cases

- **Orthogonal Design:** For orthogonal design matrices, the degrees of freedom are directly derived as k.

- **Positive Cone Condition:** Under this condition, $\text{df}(\hat{\beta}_k) = k$ is guaranteed, even for correlated predictors.

4 Properties of LARS and Lasso

4.1 Properties of LARS (Least Angle Regression)

- **Algorithm Overview:**
  - The LARS algorithm starts with all regression coefficients initialized to zero.
  - At each step, it identifies the predictor most correlated with the residuals and moves in its direction.
  - The process continues equiangularly between predictors in the active set until a new variable joins the model.
- **Efficiency:**
  - LARS is computationally efficient, requiring only m steps to compute the full solution path, where m is the number of predictors.
  - Each step involves solving a least-squares problem for the predictors currently in the active set.
- **Connections to Other Methods:**
  - LARS can be modified to compute the entire solution path for the Lasso.
  - It also closely approximates Forward Stagewise Regression but requires fewer computational steps.
- **Degrees of Freedom:**
  - The degrees of freedom for a k-step LARS solution are simply k.
  - This property simplifies the calculation of model selection criteria, such as the Cp statistic.
- **Final Model:**
  - After m steps, LARS reaches the Ordinary Least Squares (OLS) solution, as all predictors are included in the model.
- **Sign Flexibility:**
  - Unlike Lasso, LARS does not enforce sign consistency between coefficients and their correlations.
  - This flexibility allows LARS to explore a broader solution space, which can be advantageous in certain scenarios.

4.2 Properties of Lasso

- **Algorithm Overview:**
  1. Lasso minimizes the residual sum of squares with an additional constraint on the sum of the absolute values of the coefficients (L1 regularization).
  2. Shrinks some coefficients exactly to zero, promoting sparsity in the model.
- **Connection to LARS:**
  1. The full Lasso solution path can be efficiently computed using a modification of the LARS algorithm.
  2. The modification involves adjusting the algorithm when coefficients change sign.
- **Parsimony:**
  - Lasso achieves sparsity by selecting only a subset of predictors, making it suitable for high-dimensional data.
- **Degrees of Freedom:**
  - Unlike LARS, Lasso degrees of freedom correspond to the number of nonzero coefficients in the solution.
- **Final Model:**
  - The Lasso solution path can include steps where variables drop out of the active set and reenter later, unlike the standard LARS.
- **Robustness:**
  - Effective for correlated predictors due to its ability to trade off variance for bias, improving prediction accuracy.

4.3 Key Differences Between LARS and Lasso

- **Sign Constraints:**
  - LARS does not enforce sign consistency between the regression coefficients and their corresponding correlations.
  - In contrast, Lasso ensures that nonzero coefficients maintain the same sign as their correlations, promoting stability in the solution.
- **Sparsity:**
  - Lasso explicitly encourages sparsity by shrinking some coefficients to exactly zero through its ℓ_1-norm penalty.

  - While LARS provides a sparse solution path, it may not set coefficients to zero in the same way as Lasso.

- **Computational Steps:**
  - LARS completes the solution path in exactly m steps, where m is the number of predictors.
  - Lasso may take more steps because coefficients can drop out of the active set and re-enter later during the solution process.

- **Model Complexity:**
  - LARS focuses on creating a full solution path by iteratively adding predictors.
  - Lasso balances model complexity with prediction accuracy by penalizing large coefficients and promoting sparsity.

- **Flexibility:**
  - LARS provides a broader exploration of the solution space without imposing strict constraints on coefficients.
  - Lasso's penalty introduces bias but reduces variance, making it effective for handling high-dimensional data and correlated predictors.

5 Implementation of LARS Algorithm

5.1 Step 1: Generate Synthetic Data

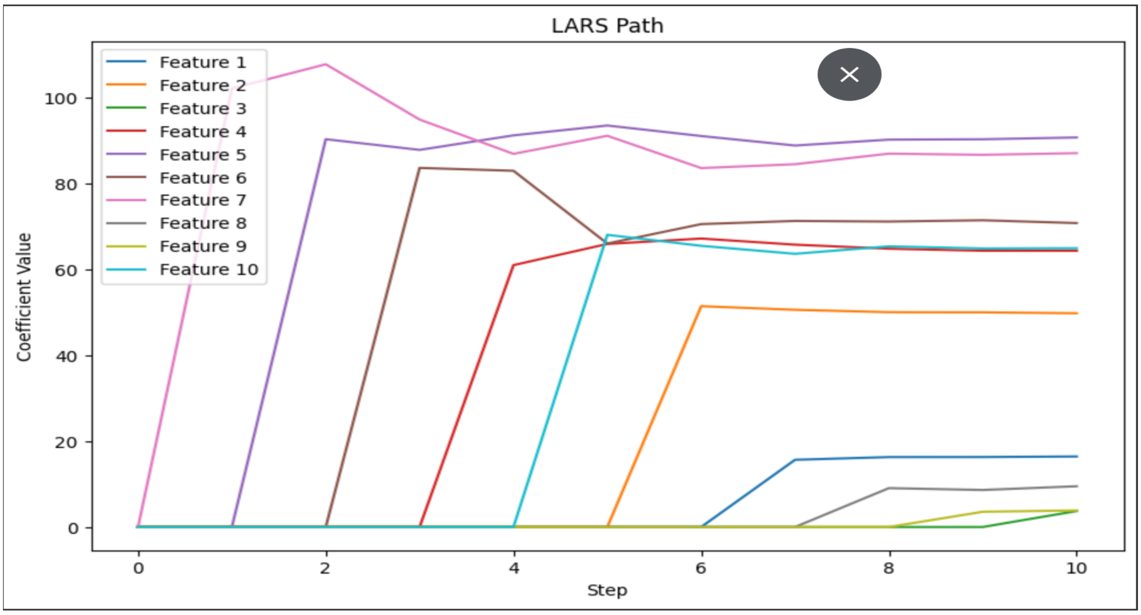
- Use the `make_regression` function from `scikit-learn` to create a synthetic dataset.
- Generate 100 samples and 10 features with added noise to simulate real-world conditions.
- This dataset serves as input to the LARS algorithm for demonstration purposes.

5.2 Step 2: Implement the LARS Algorithm

- **Normalize**: Standardize predictors and center the response variable.
- **Active Set**: Gradually add predictors with the highest correlation to the residuals.
- **Least Squares**: Solve a least-squares problem for predictors in the active set to update coefficients.
- **Update Residuals**: Recalculate residuals after each step to reflect the updated model.
- Repeat the steps until all predictors are included or the maximum number of steps is reached.

5.3 Step 3: Run the Algorithm and Visualize

- Run the LARS implementation using the synthetic dataset created earlier.
- Visualize the coefficient paths as predictors enter the model.
- The plot shows how each predictor's coefficient evolves as the algorithm proceeds.



5.4 Conclusion from the LARS Path Plot

The plot provides a clear visualization of how the coefficients of different features evolve as the Least Angle Regression (LARS) algorithm progresses through its steps. Key takeaways include:

- **Feature Selection Order:** Features are added to the model one by one, based on their correlation with the residuals. For instance, *Feature 5* and *Feature 2* are introduced early, indicating they have strong initial correlations with the response variable.
- **Coefficient Growth:** As the algorithm progresses, the coefficients of the selected features increase gradually. This reflects their growing contribution to the model. Some features, like *Feature 5*, exhibit steep growth initially, emphasizing their significance.
- **Stability and Sparsity:**
  - Certain features, such as *Feature 9* and *Feature 3*, remain near zero throughout the process, suggesting they have weaker correlations with the response or are redundant relative to stronger predictors.

  - Once a feature is added, its coefficient stabilizes as the algorithm ensures equiangular updates with the other active predictors.

- **Model Complexity:** The LARS algorithm completes in 10 steps, which matches the total number of features. By the end, all features are included, and the solution converges to the Ordinary Least Squares (OLS) solution.

This plot effectively illustrates how LARS builds a model incrementally, prioritizing predictors with the strongest contributions first while balancing computational efficiency and interpretability.

6 Optimized version

This is the version optimized by ChatGPT.
LA Finalproject: Least Angle Regression _optimize

References

Efron, Bradley et al. (2004). “Least angle regression”. In.

Hastie, Trevor et al. (2007). “Forward stagewise regression and the monotone lasso”. In.