Solving Regression

Jordan Boyd-Graber
University of Colorado Boulder
LECTURE 12

Slides adapted from Matt Nedrich and Trevor Hastie

Roadmap

- We talked about what regression is, but now how to solve these problems
- Gradient Descent for OLS
- Least Angle Regression for LASSO

Plan

Gradient Descent for OLS

Least Angle Regression

Closed Form Estimator

- Possible for ridge regression

$$\left(\mathbf{X}^T\mathbf{X} + \lambda I\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

- But inverting a matrix is hard! Doesn't always scale.
- What if your data don't live in memory?

Closed Form Estimator

- Possible for ridge regression

$$\left(\mathbf{X}^T\mathbf{X} + \lambda I\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

- But inverting a matrix is hard! Doesn't always scale.
- What if your data don't live in memory?
- Stochastic gradient descent

Objective

- Observations should be close to $\vec{\beta} x^\top$

$$\text{Error}(\beta) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \vec{\beta} x^\top \right)^2 \tag{2}$$

- Equivalent to observations from Gaussian

OLS Gradient for 2D

For convenience, write predictions as $mx + b$

$$\frac{\partial}{\partial \mathrm{m}} = \frac{2}{N} \sum_{i=1}^{N} -x_i(y_i - (mx_i + b))$$

$$\frac{\partial}{\partial \mathrm{b}} = \frac{2}{N} \sum_{i=1}^{N} -(y_i - (mx_i + b))$$

OLS Gradient for 2D

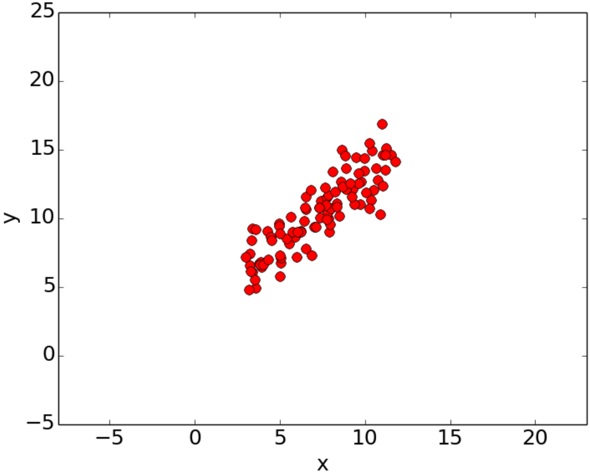
For convenience, write predictions as $mx + b$

$$\frac{\partial}{\partial \mathrm{m}} = \frac{2}{N} \sum_{i=1}^{N} -x_i(y_i - (mx_i + b))$$

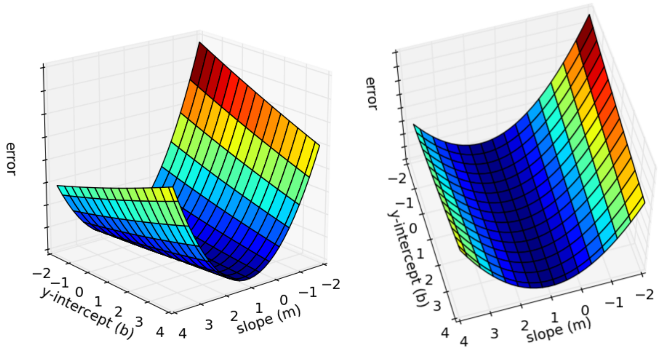
$$\frac{\partial}{\partial \mathrm{b}} = \frac{2}{N} \sum_{i=1}^{N} -(y_i - (mx_i + b))$$

Possible tweaks: stochastic gradient descent, adding regularization

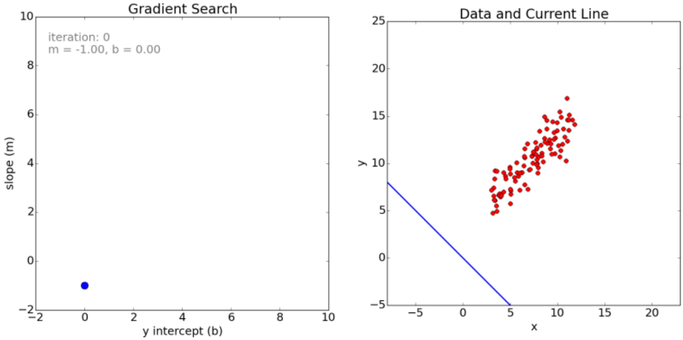
Toy Data



Toy Data

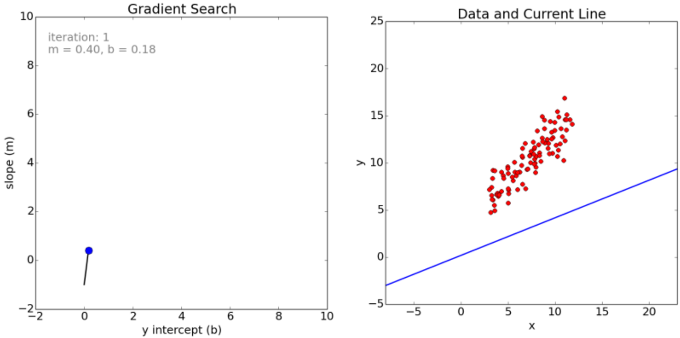

error
y-intercept (b)
slope (m)
−2
−1
0
1
2
3
4
error
y-intercept (b)
slope (m)

Running Gradient Descent



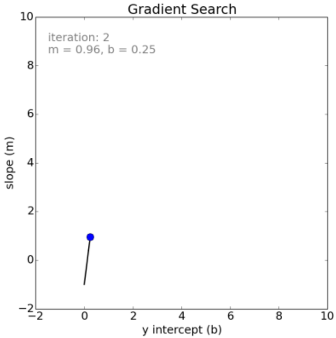
(learning rate is 0.0005)

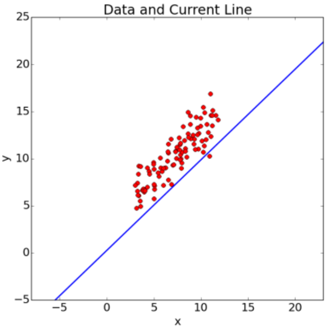
Running Gradient Descent



(learning rate is 0.0005)

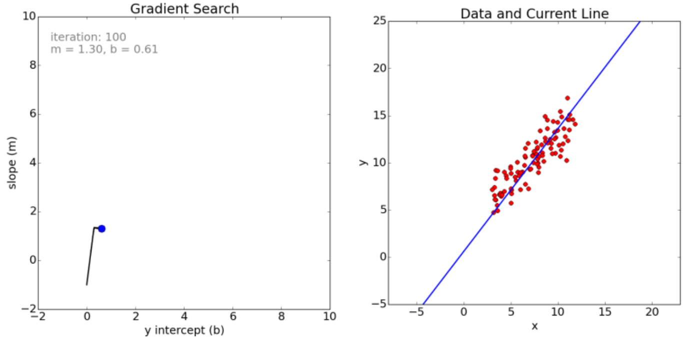
Running Gradient Descent





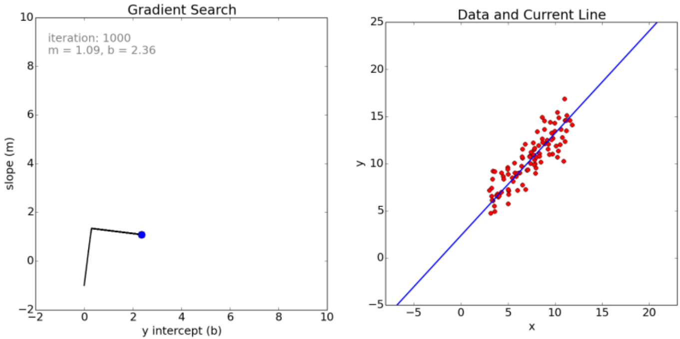
(learning rate is 0.0005)

Running Gradient Descent



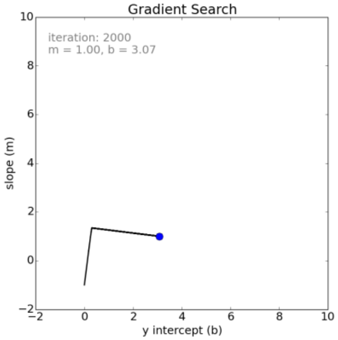
(learning rate is 0.0005)

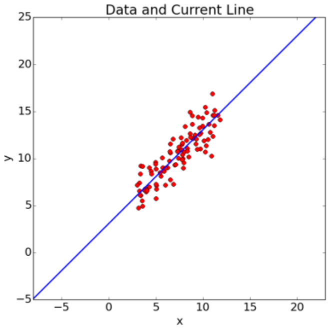
Running Gradient Descent



(learning rate is 0.0005)

Running Gradient Descent





(learning rate is 0.0005)

Plan

Gradient Descent for OLS

Least Angle Regression

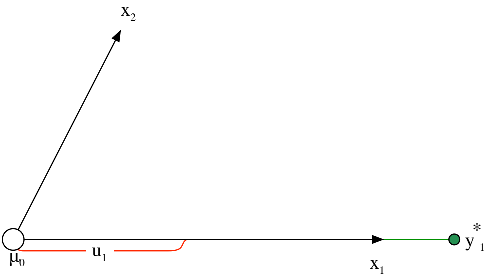
Can we use Gradient Descent for Lasso?

- Objective isn't differentiable
- Combinatorial optimization
- Similar to SMO algorithm for SVMs

LAR Algorithm

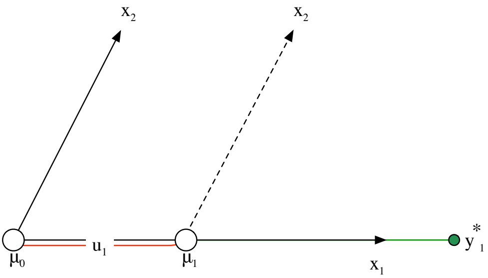
1. Start with $r = y$, $\beta_1, \ldots \beta_p = 0$. Assume x_j are all mean zero and unit variance.
2. Until all predictors have been used and $\langle r, x_j \rangle = 0 \forall j$:
    2.1 Find predictor x_j most correlated with residual r
    2.2 Increase β_j in the direction of sign $\langle r, x_j \rangle$ until some x_k has as much correlation with r as x_j or the sign of β_j changes. Call this distance u
    2.3 Update prediction μ, residual r

Intuition



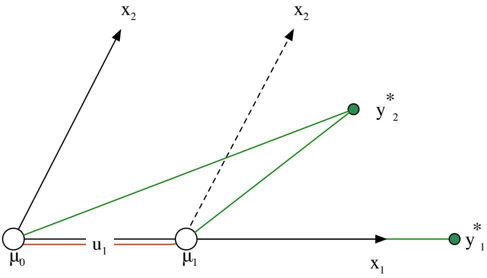
Initially, the prediction is 0, the mean of y (remember, everything is standardized). x_1 is most correlated with y, so we move in that direction (toward the OLS solution of y_1^*). We move a distance u_1 until x_2 has as much correlation with the residual.

Intuition



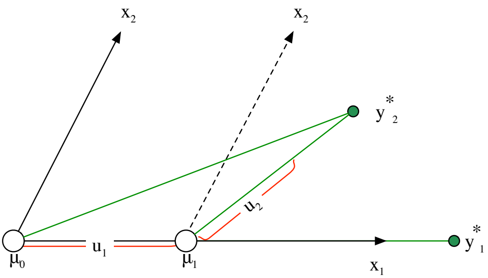
Our new estimate is μ_1, a function of just x_1. Now we need to start using x_2, so we incorporate that into our estimate.

Intuition



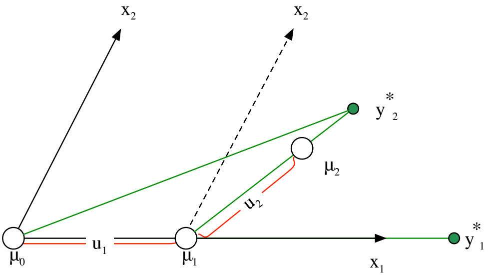
We are now moving toward the OLS solution using these two variables, y_2^*, using a combination of both x_1 and x_2.

Intuition



We move our estimate in that direction until some other variable has higher correlation with the residual. We keep moving closer and closer (but never quite reaching) the OLS solution with the current set of variables.

Intuition

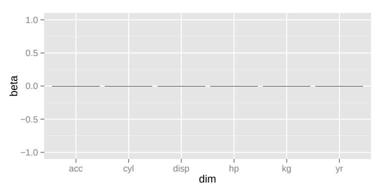


MPG Dataset

- Predict mpg from features of a car
  1. Number of cylinders
  2. Displacement
  3. Horsepower
  4. Weight
  5. Acceleration
  6. Year

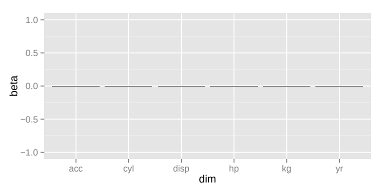
Example of LARS

β

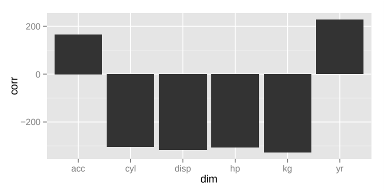


Example of LARS

β



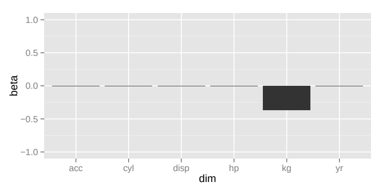
Correlation



The weight of the car is has the highest (negative) correlation with the weight, so we add that to the active set.

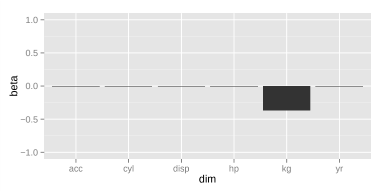
Example of LARS

β

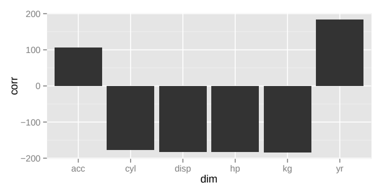


Example of LARS

β



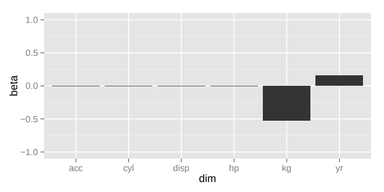
Correlation



After making predictions with only the weight, the year is the most (positively) correlated, so it gets added to the active set.

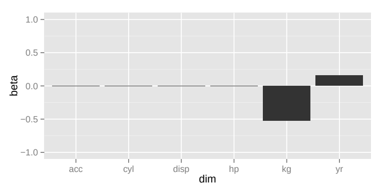
Example of LARS

β

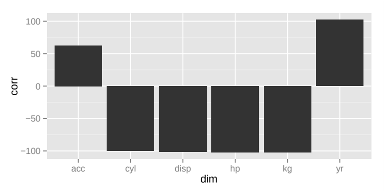


Example of LARS

β



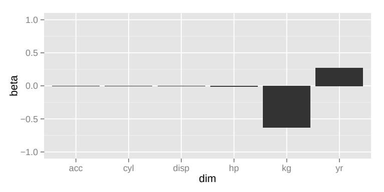
Correlation



At this point, the correlations are getting fairly small. Horsepower wins, but only contributes a tiny amount.

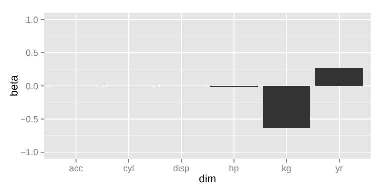
Example of LARS

β

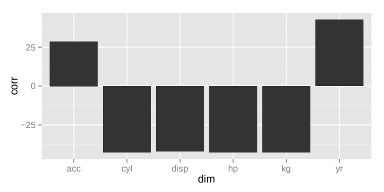


Example of LARS

β



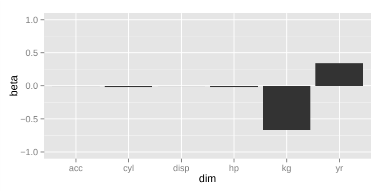
Correlation



Same story with the number of cylinders . . .

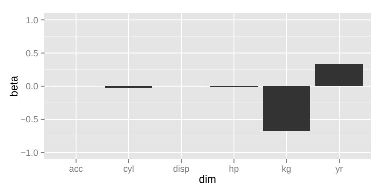
Example of LARS

β

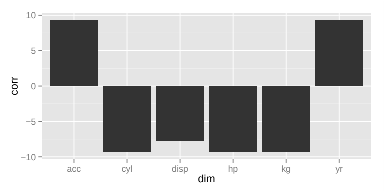


Example of LARS

β



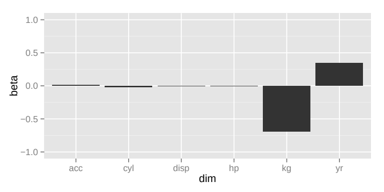
Correlation



and acceleration.

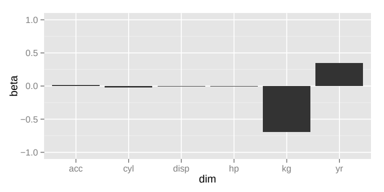
Example of LARS

β

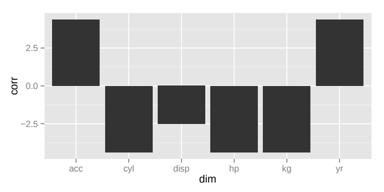


Example of LARS

β



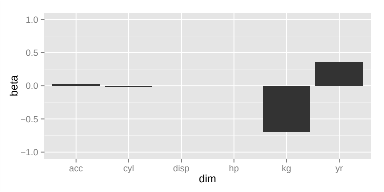
Correlation



Now the year is again the most correlated. But take a look at displacement; it's negatively correlated (about -2.5).

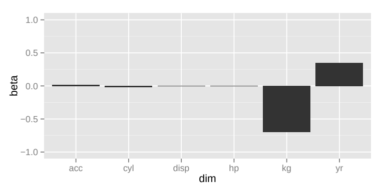
Example of LARS

β

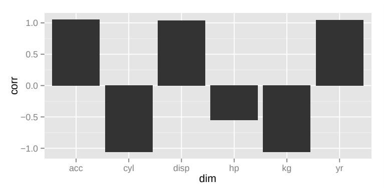


Example of LARS

β



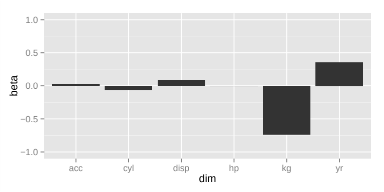
Correlation



After accounting for the other variables, it's positively correlated.

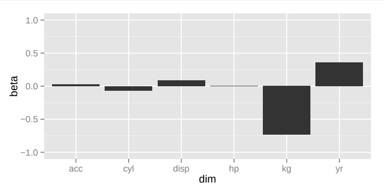
Example of LARS

β

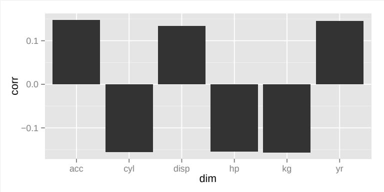


Example of LARS

β



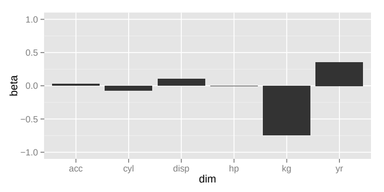
Correlation



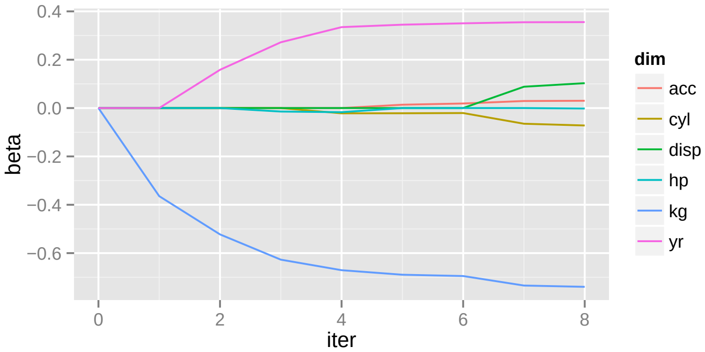
Now we have our final model.

Example of LARS

β



Coefficient Trajectories



Benefits of LARS

- Interpretation of boosting for continuous problems
- About as difficult as computing OLS for each group of variables
- No combinatorial optimization
- Finds all Lasso solutions

Recap

- Objective function for regression
- Algorithms for OLS and regularized regression
- Like classification, a workhorse method for continuous data