Applied Machine Learning

Lecture: 6 Regularization

Ekarat Rattagan, Ph.D.

Slides adapted from Andrew NG, Eric Eaton, Raquel Urtasun, and Patrick Winston

Outline

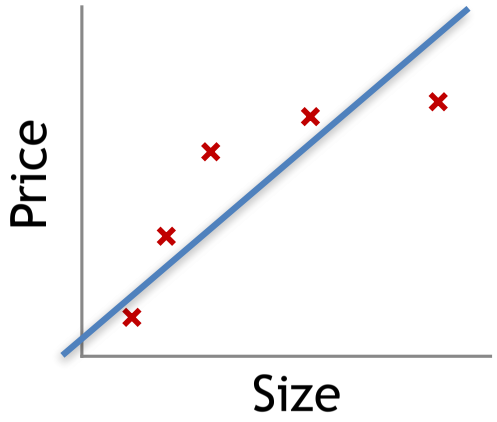
6.1 Overfitting

6.2 Regularization

- Ridge
- Lasso
- Elastic Net

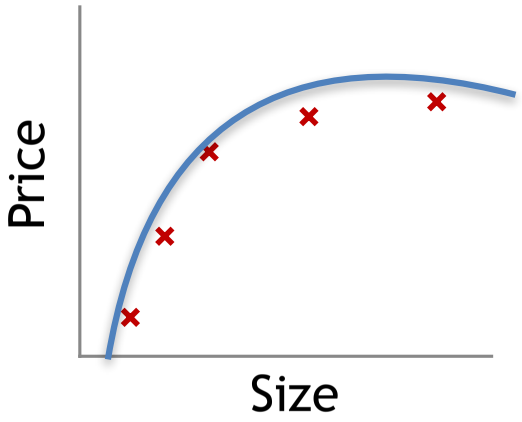
6.1 Overfitting

Example: Linear regression (housing prices)



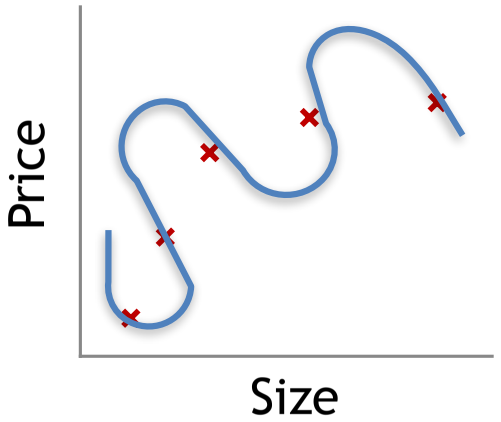
$\theta_0 + \theta_1 x$

Underfit



$\theta_0 + \theta_1 x + \theta_2 x^2$

Good fit



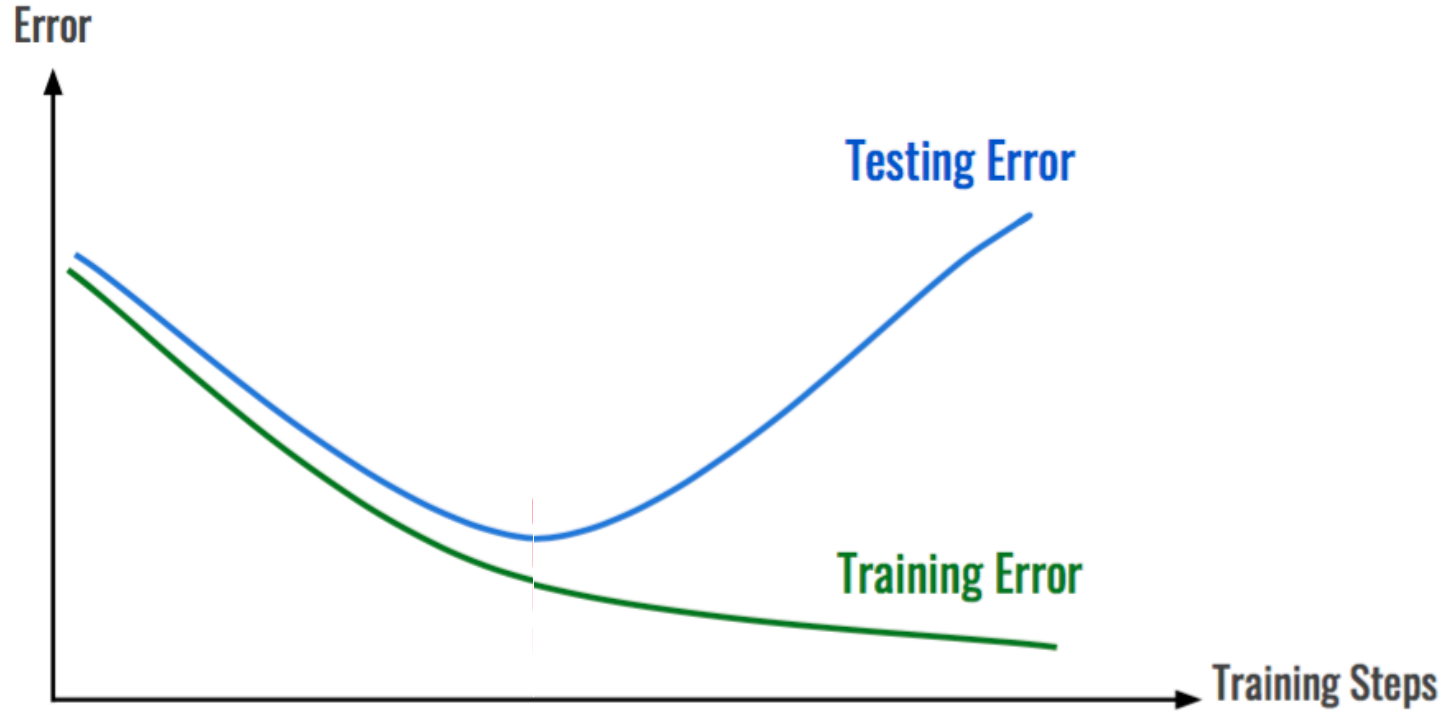
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfit

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples.

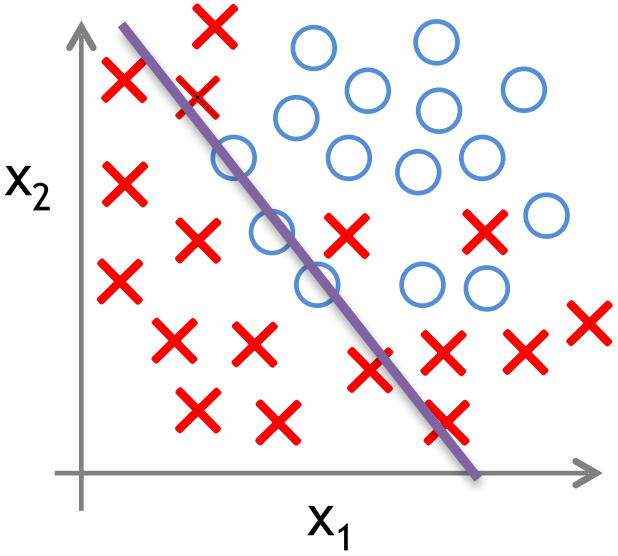
Credit: Andrew NG

Overfitting example: regression using polynomials



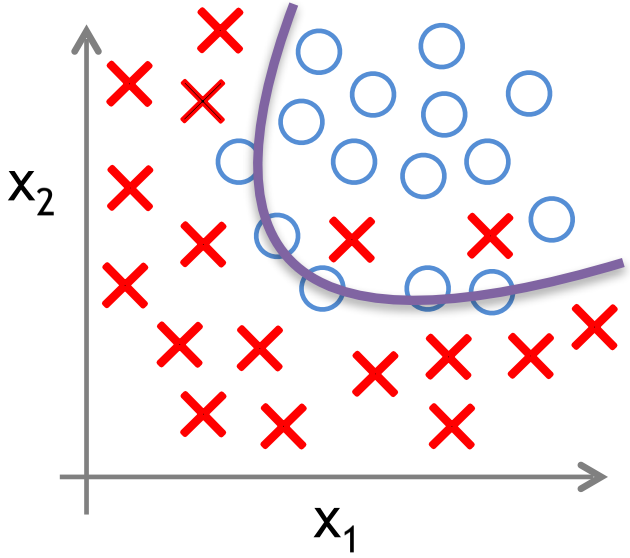
https://mc.ai/early-stopping-with-pytorch-to-restrain-your-model-from-overfitting/

Example: Logistic regression

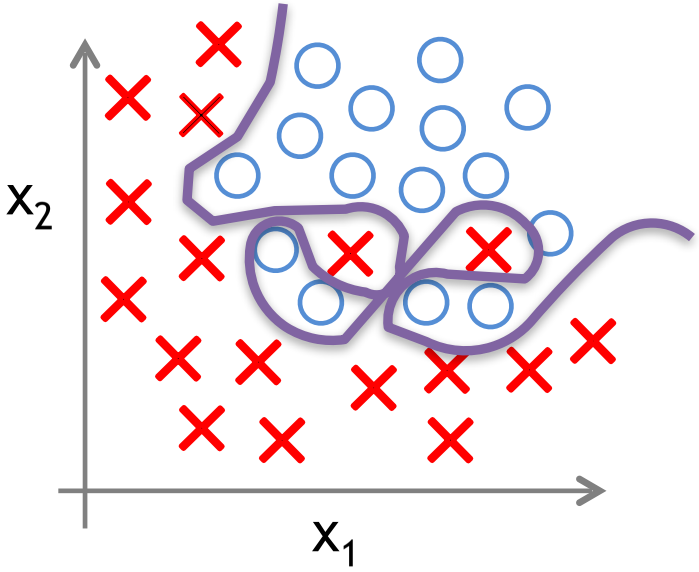


$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$

Credit: Andrew NG

Addressing overfitting:

Options:

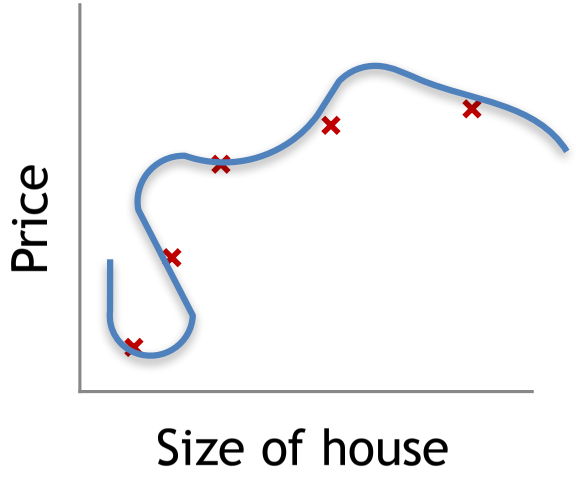
1. Reduce number of features
   - Manually select which features to keep.
   - Model selection algorithm.

2. Regularization
   - Keep all the features, but reduce magnitude/values of parameters θ_j.
     - Works well when we have a lot of features, each of which contributes a bit to predicting y.

Credit: Andrew NG

6.2 Regularization

Intuition

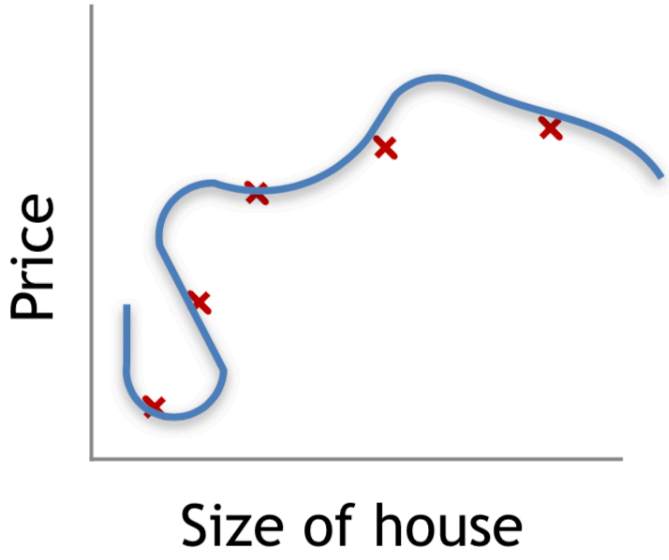


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

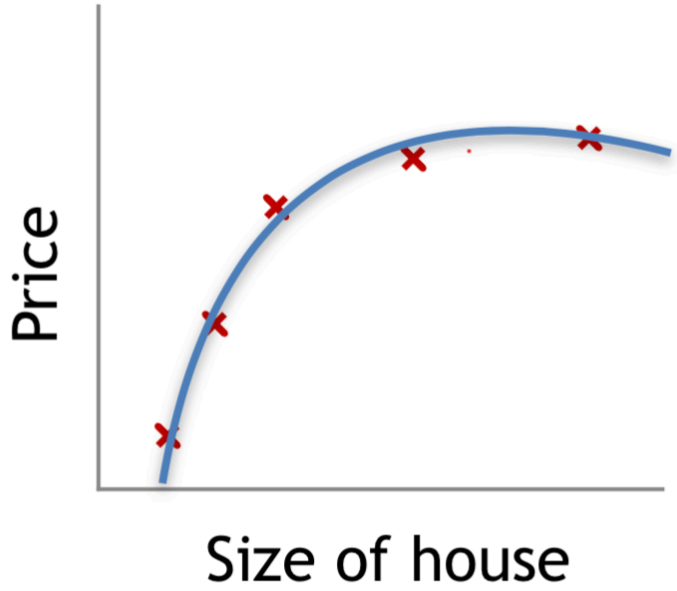
Suppose we penalize and make θ_3, θ_4 really small.
The curve will change.

A good way to reduce overfitting is to regularize the model (i.e., to constrain it)

Intuition



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Suppose we penalize and make θ_3, θ_4 really small.
The curve will change.

A good way to reduce overfitting is to regularize the model (i.e., to constrain it)

Regularized Linear Models

- Ridge regression
- Lasso regression
- Elastic Net

Ridge regression ($squared\ L_2 - norm$)

- Additional term in the cost function to prevent overfitting
  - keep the model weights/coefficients/parameters as small as possible
  - L_2 (Euclidean) norm: A square root of the sum of the squared vector values $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$

Tuning/ridge/penalty parameter controls how much you want to regularize the model

$$+\lambda\|x\|_2^2$$

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right)$$

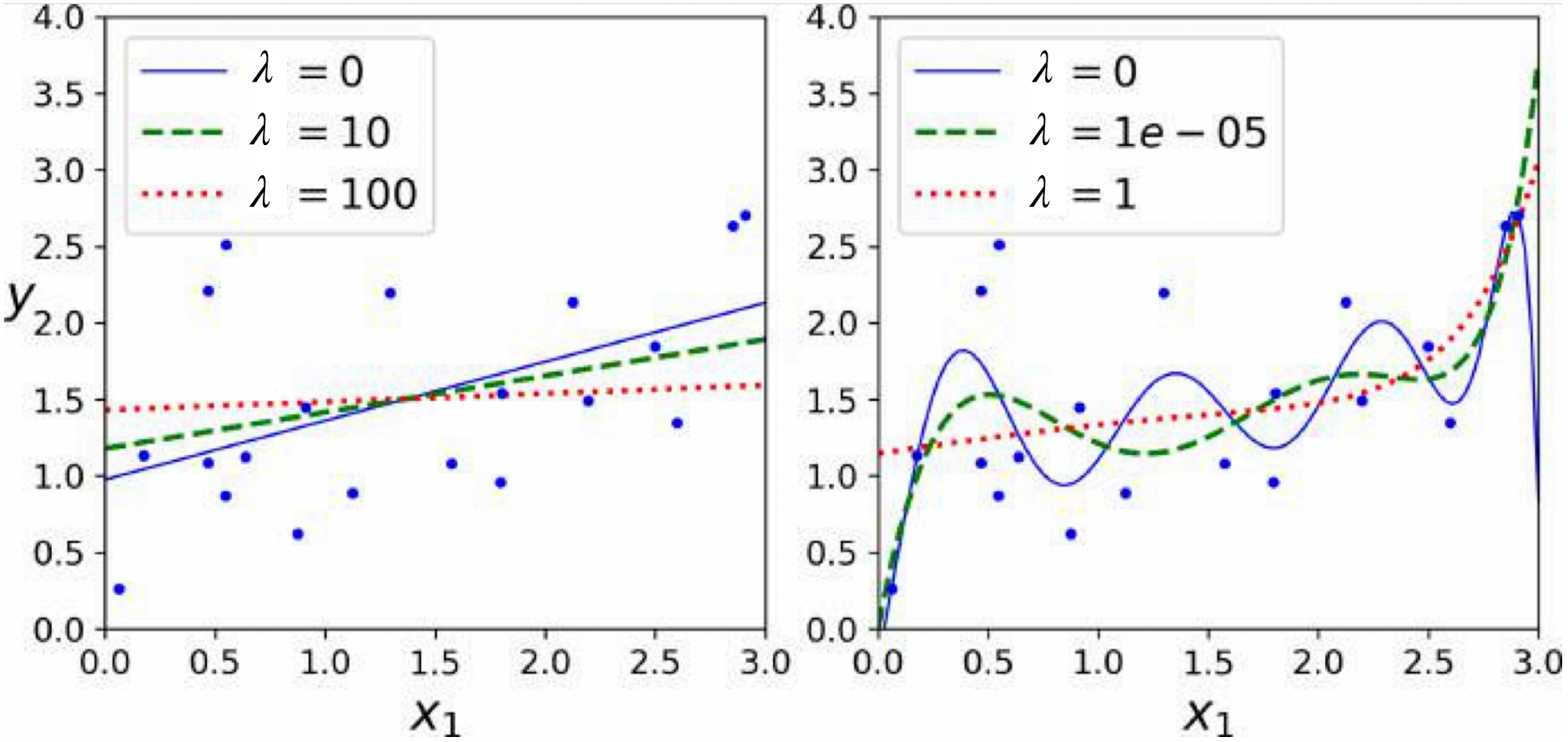
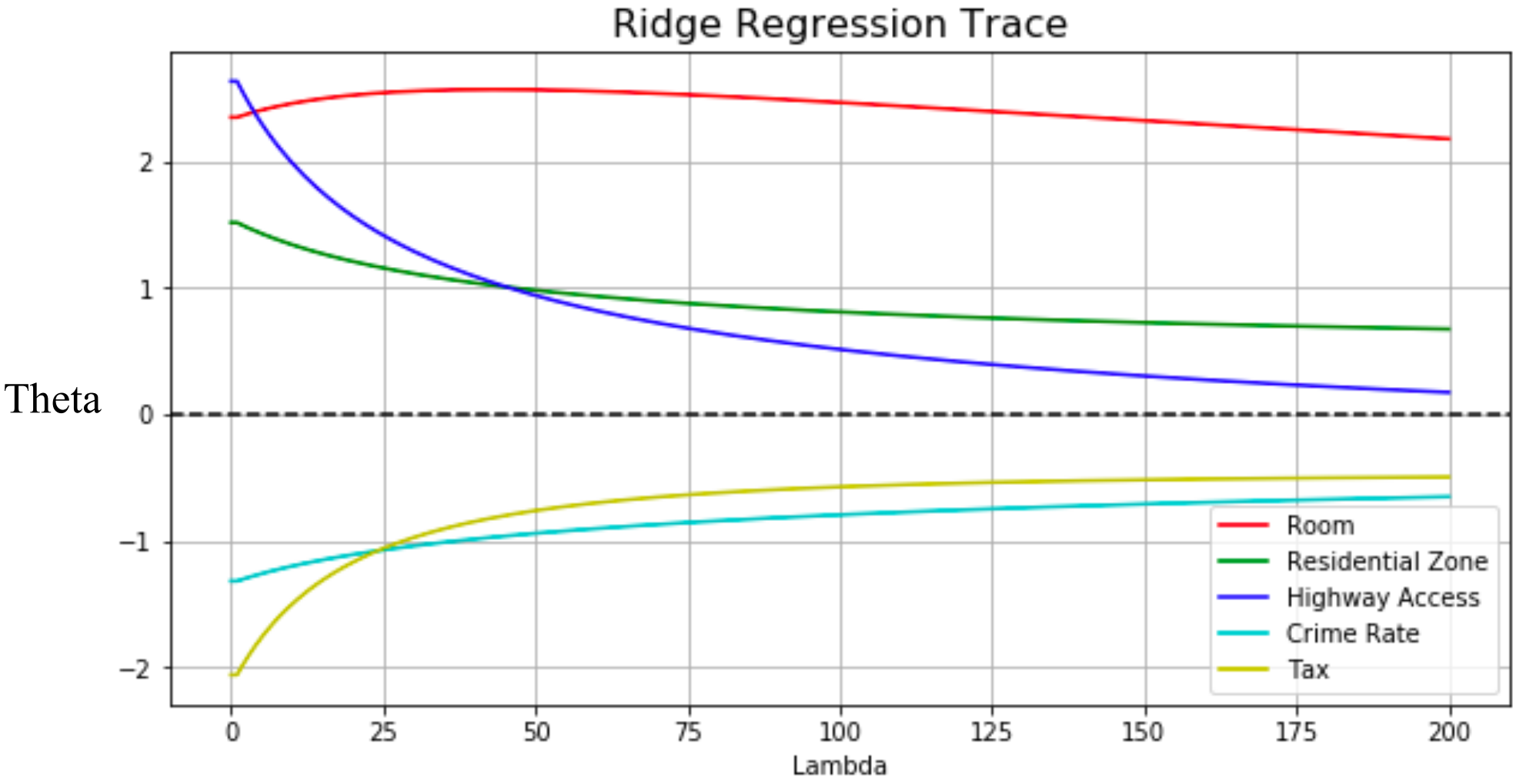


Figure from Hands-on Machine Learning

Ridge regression



https://github.com/Q-shick/fundamentals_of_data_science

Lasso regression ($L_1 - norm$)

- Least Absolute Shrinkage and Selection Operator (LASSO)
  - Reduce the number of features in a regression model
  - Automatically performs feature selection and outputs a sparse model
  - L_1 (Absolute-value) norm: $\|x\|_1 = |x_1 + x_2 + \ldots + x_n|$

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} |\theta_j|\right)$$

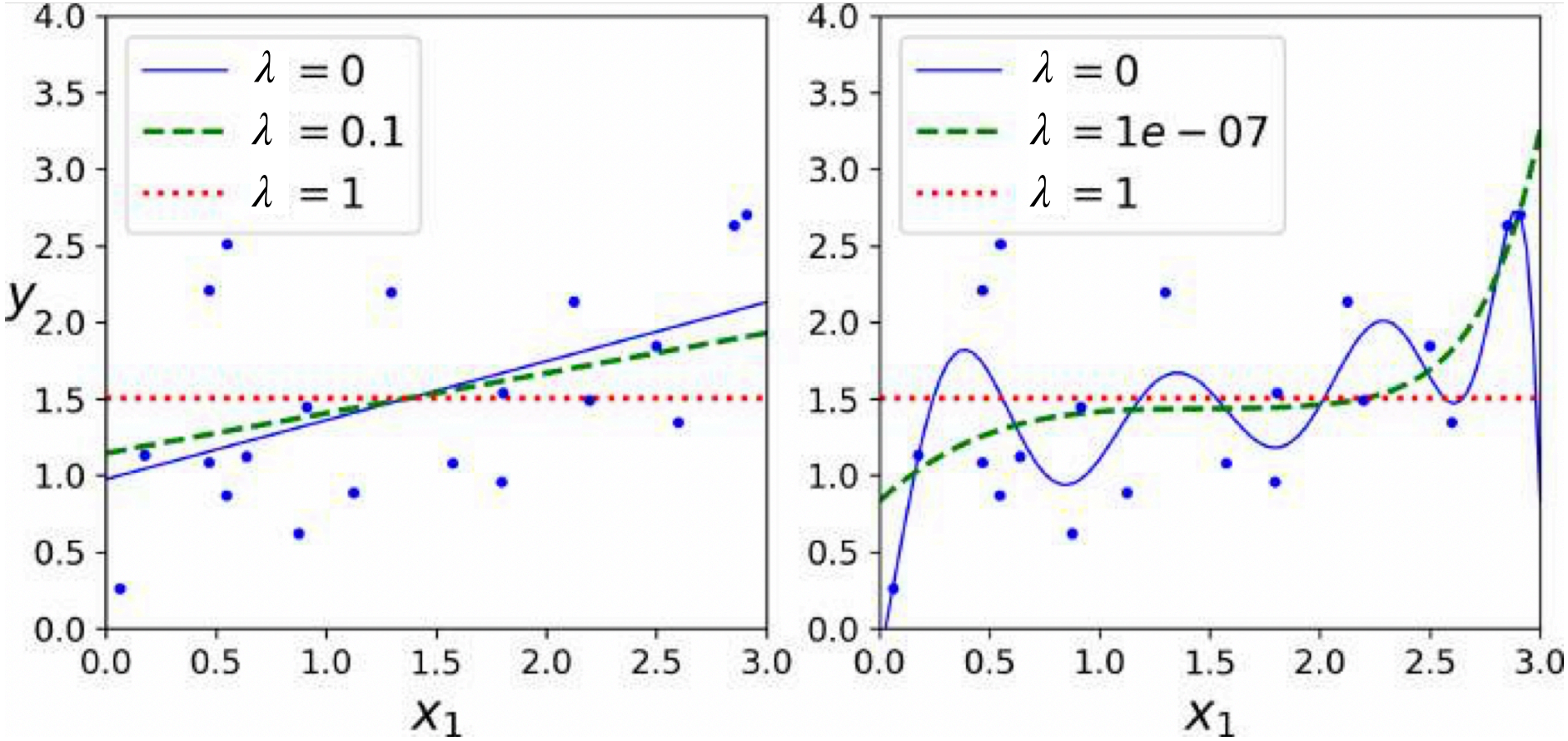
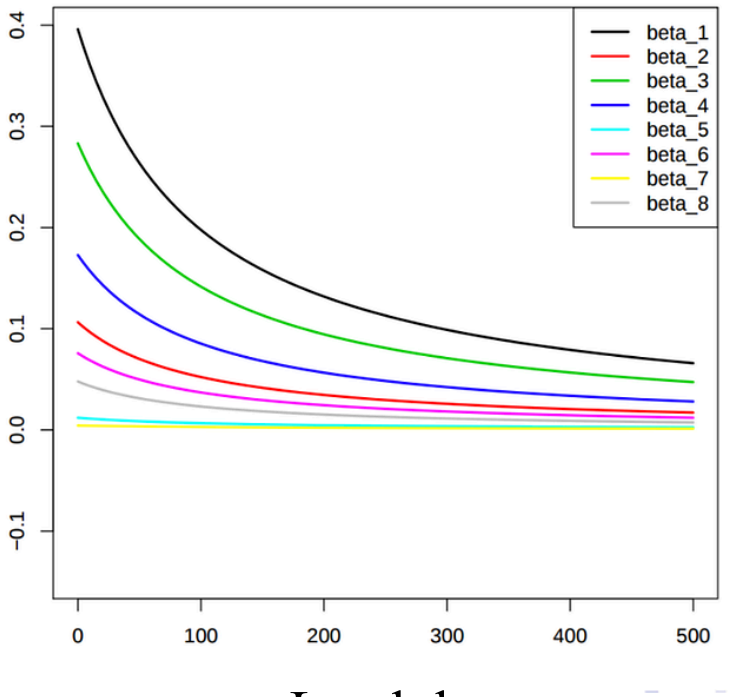


Figure from Hands-on Machine Learning

Lasso regression

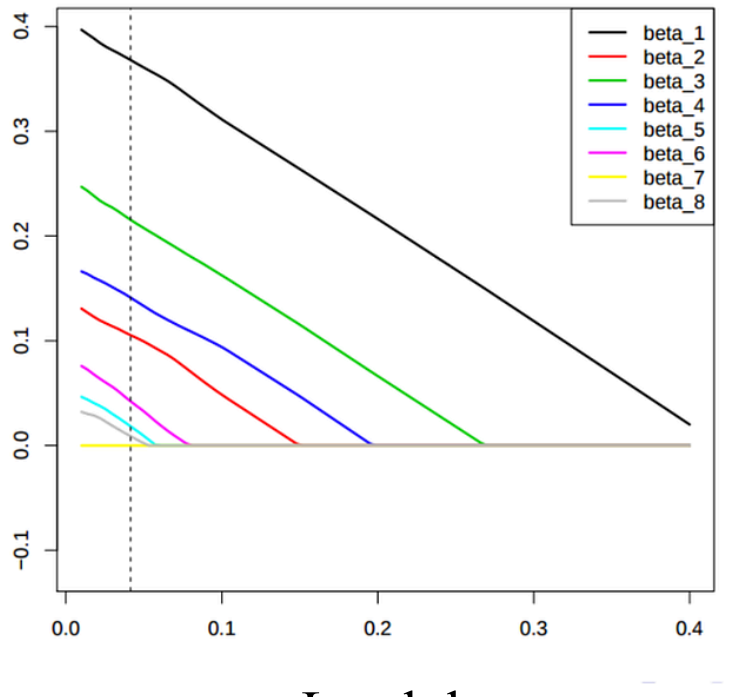
Ridge vs Lasso

Theta



Lambda

Theta



Lambda

https://www.thelearningmachine.ai/lle

When to use what?

Lasso tends to do well if there are a small number of significant parameters and the others are close to zero (hence: when only a few predictors actually influence the response).

Ridge works well if there are many large parameters of about the same value (hence: when most predictors impact the response).

However, in practice, we don't know the true parameter values, so the previous two points are somewhat theoretical. Just run **cross-validation** to select the more suited model for a specific case.
Or... combine the two!

https://shuaili8.github.io/Teaching/VE445/L3_linear%20regression.pdf

Case study

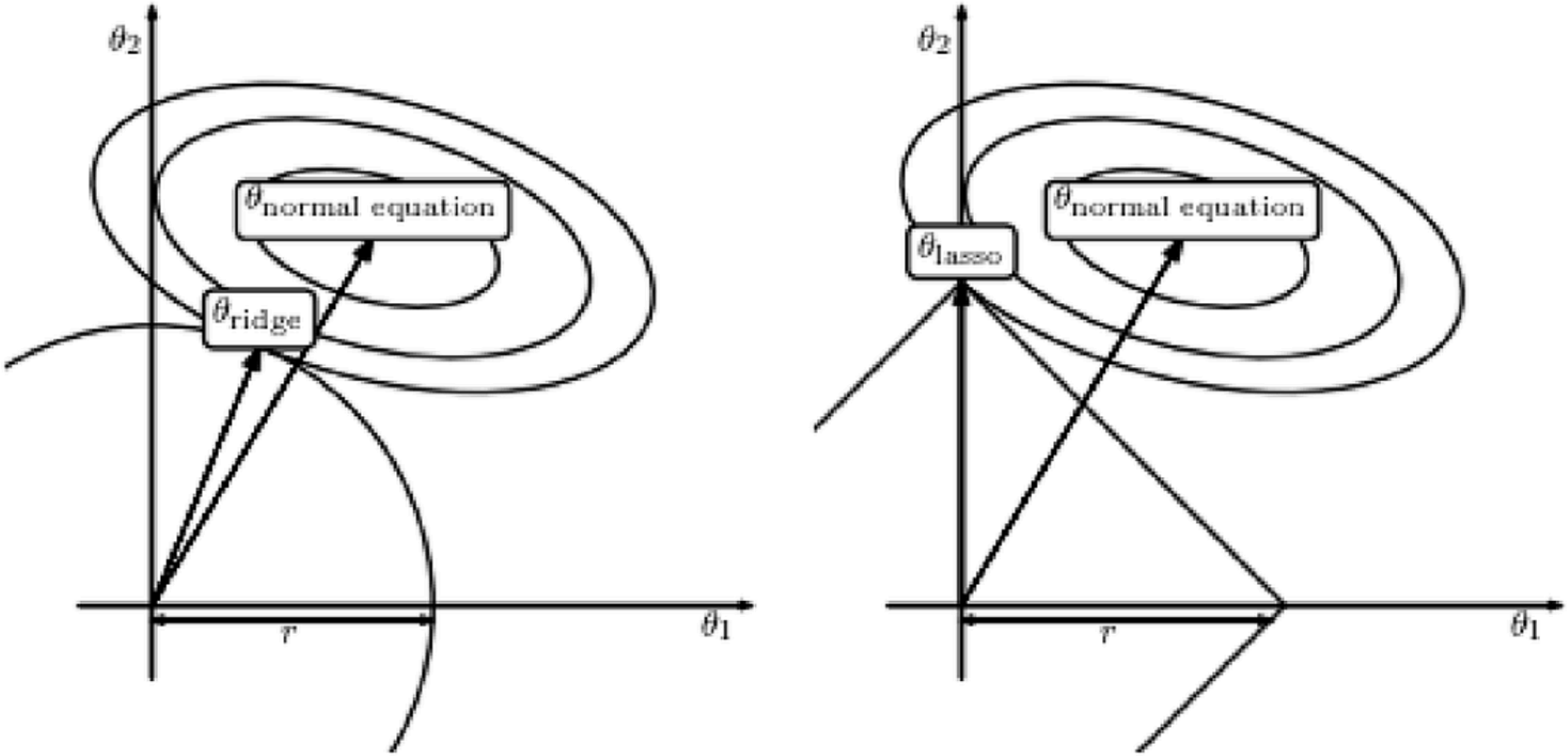
Dmitry Smirnov, Consultant

Answered June 19, 2019

Ridge and Lasso are regularized versions of linear regression. In practice, ridge regression leads to lower weights of coefficients, while lasso regressions leads to more coefficients having zero weights.

Which one fits better - depends on your domain, when I was in neuroscience, we often preferred ridge, because when analysing fMRI data you have multiple sources of signal that are located close to each other and having correlated signals, and ridge would provide similar coefficients for these sources, while lasso will drive a lot of these to zero while preserving some that will have higher weights.

Ridge vs Lasso



https://www.thelearningmachine.ai/lle

Elastic Net

- A simple mix of both Ridge and Lasso's regularization terms, and you can control the mix ratio α, where

$$\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

  - When $\alpha = 1$, Elastic Net is equivalent to Ridge Regression, and when $\alpha = 0$, it is equivalent to Lasso Regression.

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + (1-\alpha)\sum_{j=1}^{n}|\theta_j| + \alpha\sum_{j=1}^{n}\theta_j^2\right)$$

Elastic Net

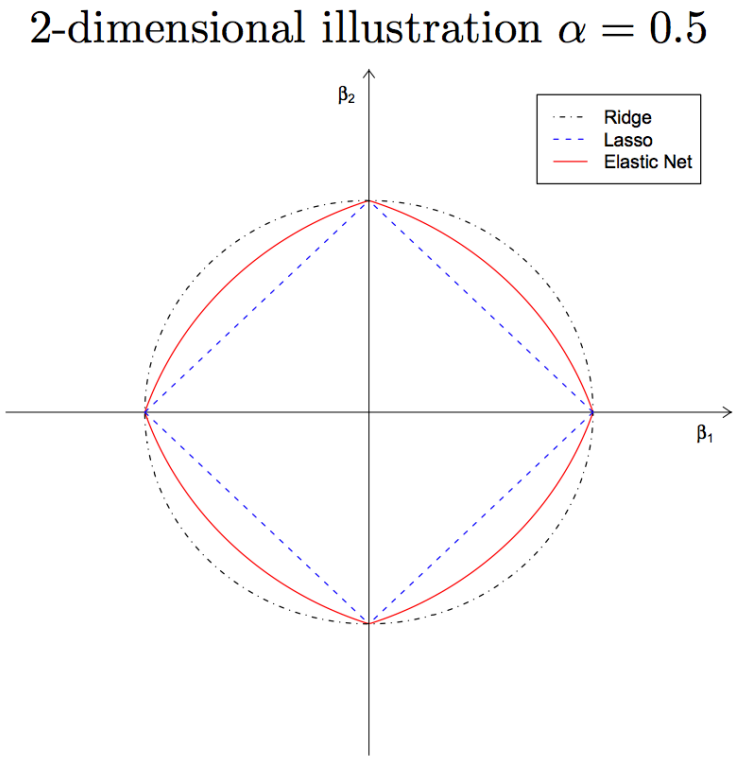
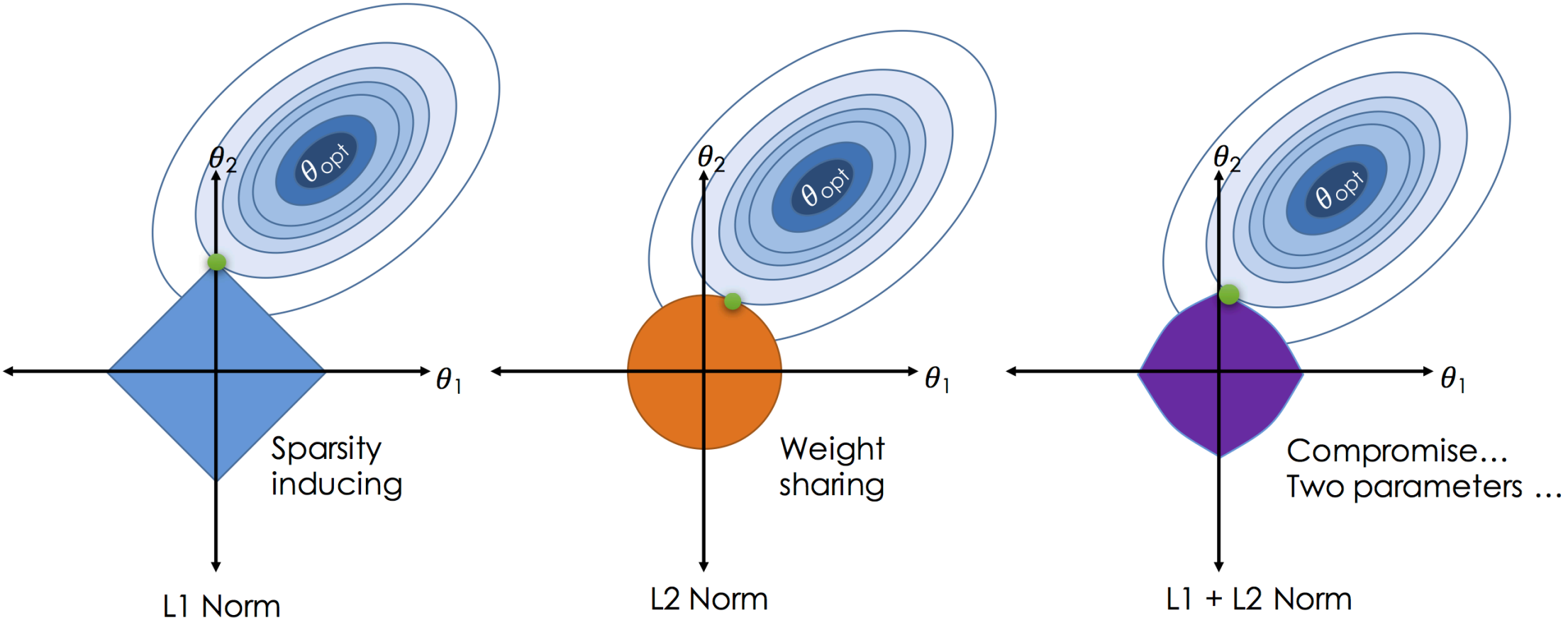


Figure from: Regularization and Variable Selection via the Elastic Net
Hui Zou and Trevor Hastie Department of Statistics Stanford University



https://medium.com/@savannahar68/getting-started-with-regression-a39aca03b75f

When to use what?

So when should you use plain Linear Regression (i.e., without any regularization, Ridge, Lasso, or Elastic Net) ?

It is almost always preferable to have at least a little bit of regularization, Ridge is a good default,

but if you suspect that only a few features are actually useful, you should prefer Lasso or Elastic Net since they tend to reduce the useless features' weights down to zero.

In general, Elastic Net is preferred over Lasso since Lasso may behave erratically when the number of features is greater than the number of training instances or when several features are strongly correlated.

Stochastic Gradient descent

Ridge regression

repeat until convergence {

$$\theta_0 = \theta_0 - \alpha \left((h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} + \lambda \theta_0 \right)$$

$$\theta_1 = \theta_1 - \alpha \left((h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)} + \lambda \theta_1 \right)$$

$$\vdots$$

$$\theta_n = \theta_n - \alpha \left((h_\theta(x^{(i)}) - y^{(i)}) x_n^{(i)} + \lambda \theta_n \right)$$

}

Stochastic Gradient descent

Lasso regression

The Lasso cost function is not differentiable at $\theta_j = 0$, where $j = 0, 1, \ldots, n$ but Gradient Descent still works fine if you use a sub-gradient vector instead when any $\theta_j = 0$.

repeat until convergence {

$$\theta_j = \theta_j - \alpha \left[(h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \lambda \; sign(\theta_j) \right] \quad , for \; j = 0 \ldots n$$

}

$$\begin{pmatrix} \text{sign}\,(\theta_1) \\ \text{sign}\,(\theta_2) \\ \vdots \\ \text{sign}\,(\theta_n) \end{pmatrix} \quad \text{where} \;\; \text{sign}\,(\theta_j) = \begin{cases} -1 & \text{if } \theta_i < 0 \\ 0 & \text{if } \theta_i = 0 \\ +1 & \text{if } \theta_i > 0 \end{cases}$$

Stochastic Gradient descent

Elastic Net $\alpha = \dfrac{\lambda_2}{\lambda_1 + \lambda_2}$

repeat until convergence {

$$\theta_j = \theta_j - \alpha \left[h_\theta(x^{(i)}) - y^{(i)} x_j^{(i)} + (1 - \alpha)\ sign(\theta_j) + \alpha \theta_j \right]$$

}

Reference

- Elastic Net

  Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." Journal of the royal statistical society: series B (statistical methodology) 67, no. 2 (2005): 301-320.

  https://www.youtube.com/watch?v=TJer4ibUZ0I