UCK358E – INTR. TO ARTIFICIAL INTELLIGENCE

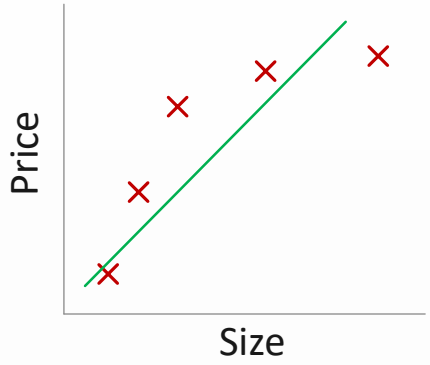
SPRING ‘23

LECTURE 5

REGULARIZATION

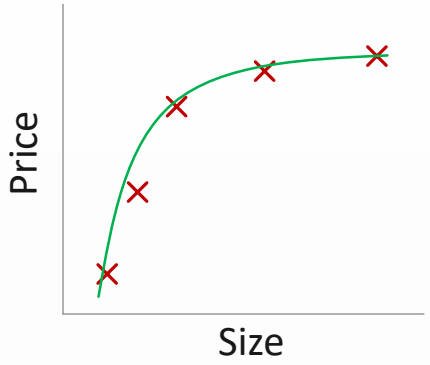
Instructor: Asst. Prof. Barış Başpınar

Example: Linear regression (housing prices)



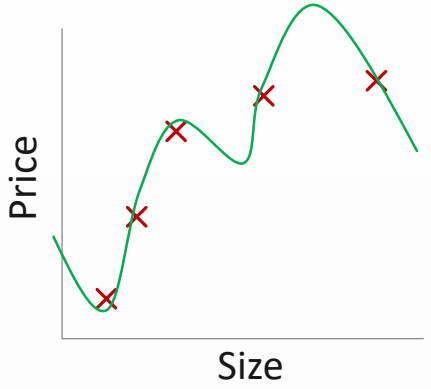
$\theta_0 + \theta_1 x$

Underfit, High Bias



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

Just Right

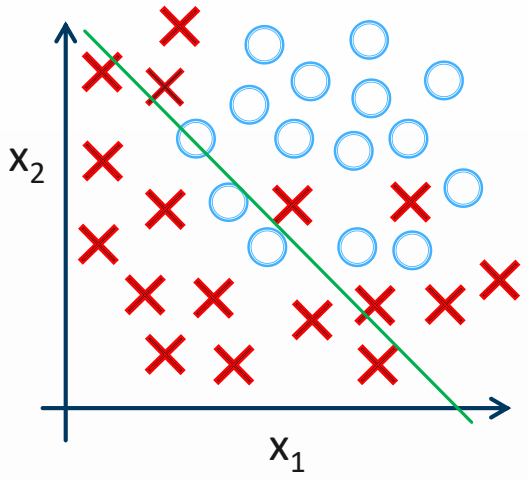


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfit, High Variance

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 (predict prices on new examples).

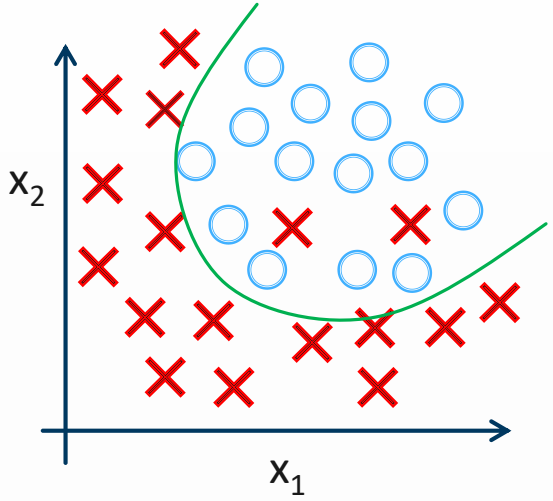
Example: Logistic regression



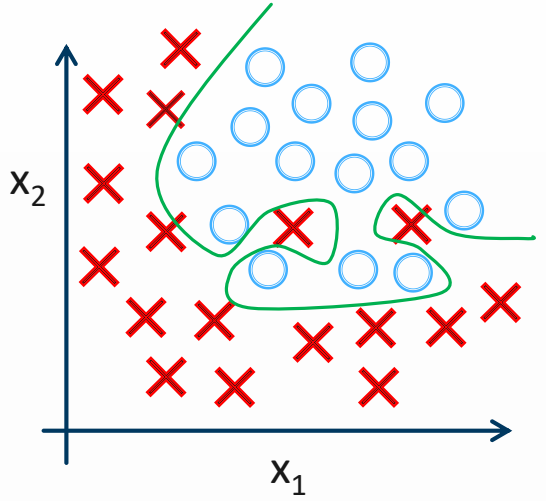
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

(g = sigmoid function)

"Underfit"



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



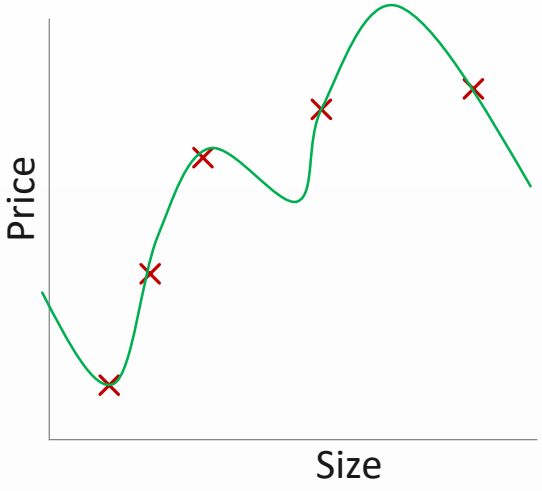
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

"Overfit"

Addressing overfitting

$x_1 =$ size of house
$x_2 =$ no. of bedrooms
$x_3 =$ no. of floors
$x_4 =$ age of house
$x_5 =$ average income in neighborhood
$x_6 =$ kitchen size
$\vdots$
x_{100}

Many features due to available data



Many features due to high-degree polynomial

Overfitting can become a problem when:

- Using high-degree polynomials, or complex models
- Having a lot of features and very little training data

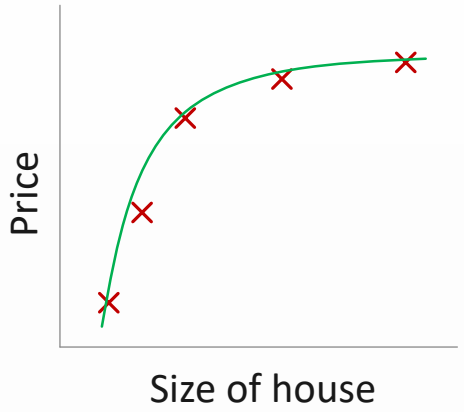
Addressing overfitting

Options:

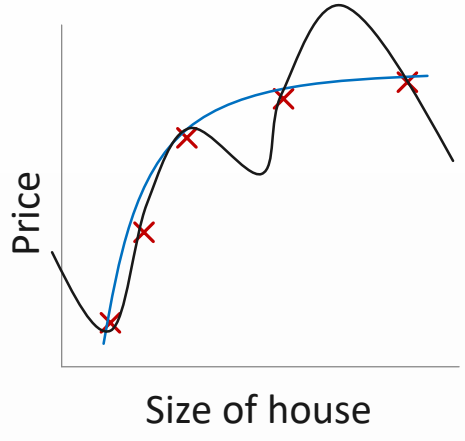
1. Reduce number of features
   - Manually select which features to keep.
   - Model selection algorithm (later in course)

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

Intuition



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



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

Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \quad + 1000\theta_3^2 \; + \; 1000\theta_4^2$$

$$\theta_3 \cong 0, \qquad \theta_4 \cong 0$$

Regularization

Small values for parameters $\theta_0, \theta_1, \ldots, \theta_n$

- "Simpler" hypothesis
- Less prone to overfitting

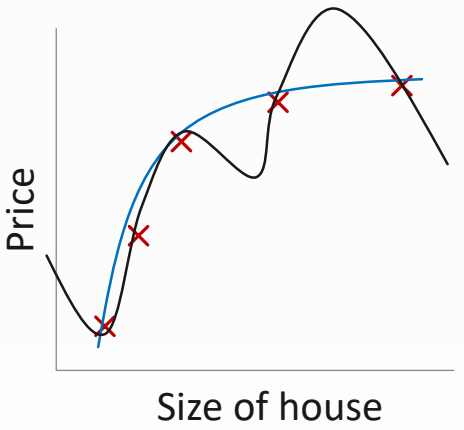
Housing:

- Features: $x_1, x_2, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

Regularization parameter

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

$$\min_\theta J(\theta)$$



Regularization

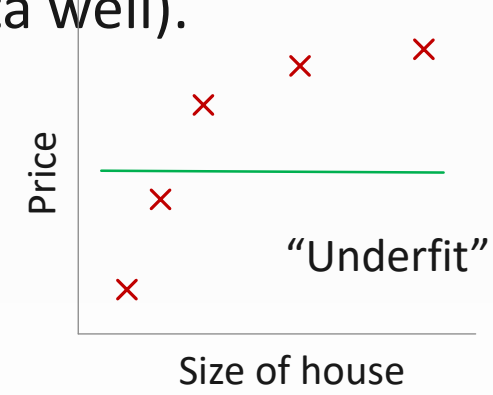
In regularized linear regression, we choose θ to minimize

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

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

- Algorithm works fine; setting λ to be very large can't hurt it
- Algorithm fails to eliminate overfitting.
- Algorithm results in underfitting. (Fails to fit even training data well).
- Gradient descent will fail to converge.

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



Regularized linear regression

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

Gradient descent

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} - \frac{\lambda}{m}\theta_j \right]$$

$$(j = \cancel{0}, 1, 2, 3, \ldots, n)$$

$\}$

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

Regularized logistic regression

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

Gradient descent

Repeat $\{$

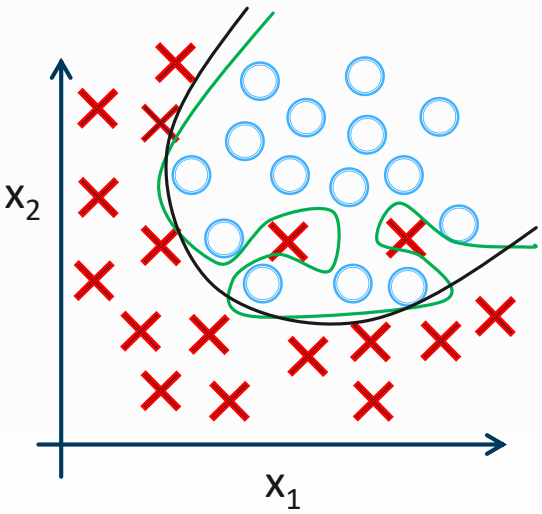
$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} - \frac{\lambda}{m}\theta_j\right]$$

$$(j = \cancel{0}, 1, 2, 3, \ldots, n)$$

$\}$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$



Alternative Regularization Terms

Ordinary Least Squares

$$\hat{y} = w^T \mathbf{x} + b = \sum_{i=1}^{p} w_i x_i + b$$

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{p} ||w^T \mathbf{x}_i - y_i||^2$$

Ridge Regression

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} ||w^T \mathbf{x}_i - y_i||^2 + \alpha ||w||^2$$

(We have already defined this one!)

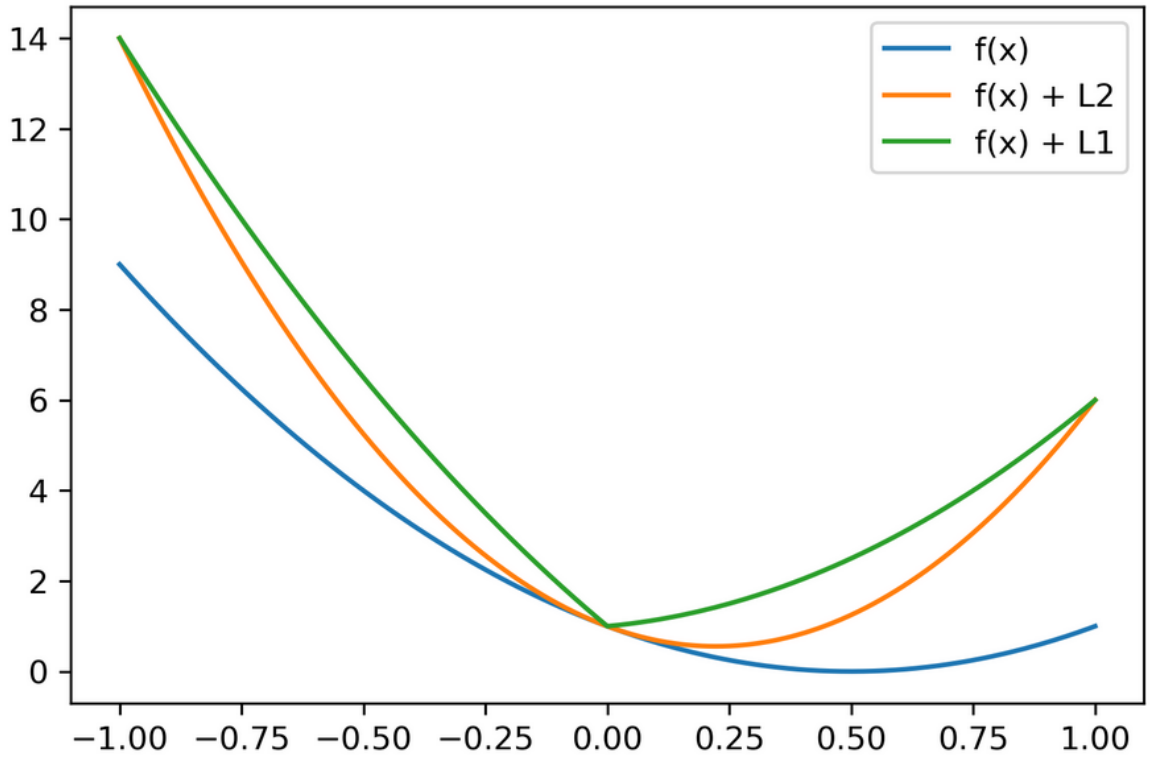
Lasso Regression

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} ||w^T \mathbf{x}_i - y_i||^2 + \alpha ||w||_1$$

- Shrinks w towards zero like Ridge
- Sets some w exactly to zero - automatic feature selection!

Alternative Regularization Terms

Understanding L1 and L2 Penalties



$$f(x) = (2x-1)^2$$
$$f(x) + L2 = (2x-1)^2 + \alpha x^2$$
$$f(x) + L1 = (2x-1)^2 + \alpha|x|$$

Alternative Regularization Terms

Ridge regression

- In ridge regression, though, the coefficients (*w*) are chosen not only so that they predict well on the training data, but also to fit an additional constraint.
- We also want the magnitude of coefficients to be as small as possible; in other words, all entries of *w* should be close to zero.
- Intuitively, this means each feature should have as little effect on the outcome as possible (which translates to having a small slope), while still predicting well.
- The particular regularization term used by ridge regression is known as L2 regularization.
- Ridge is a more restricted model, so we are less likely to overfit.
- The optimum setting of alpha depends on the particular dataset we are using. Increasing alpha forces coefficients to move more toward zero, which decreases training set performance but might help generalization.

Alternative Regularization Terms

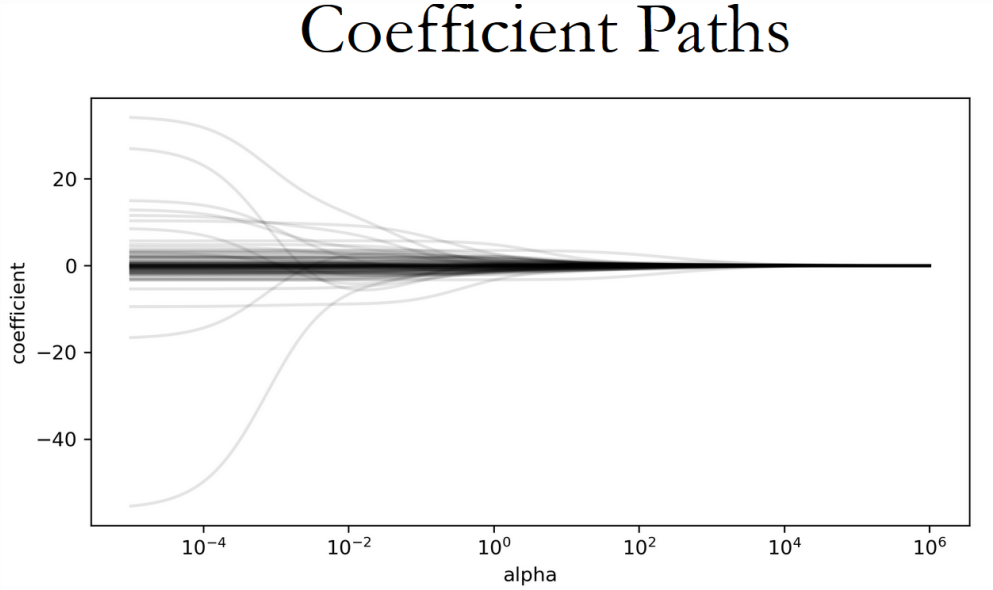
Lasso regression

- An alternative to Ridge for regularizing linear regression is Lasso. As with ridge regression, using the lasso also restricts coefficients to be close to zero, but in a slightly different way, called L1 regularization.
- The consequence of L1 regularization is that when using the lasso, some coefficients are *exactly zero*. This means some features are entirely ignored by the model.
- This can be seen as a form of automatic feature selection. Having some coefficients be exactly zero often makes a model easier to interpret, and can reveal the most important features of your model.
- Two important points:
  - Solution numeracy: Because L2 is Euclidean distance, there is always one right answer as to how to get between two points fastest. Because L1 is taxicab distance, there are as many solutions to getting between two points
  - Computational difficulty: L2 has a closed form solution because it's a square of a thing. L1 does not have a closed form solution because it is a non-differenciable piecewise function, as it involves an absolute value. For this reason, L1 is computationally more expensive

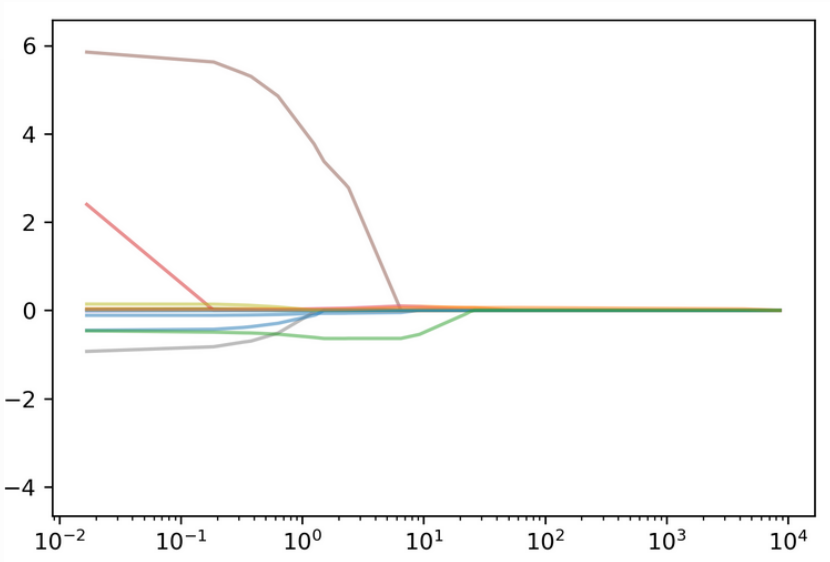
Alternative Regularization Terms

- In practice, ridge regression is usually the first choice between these two models.
- However, if you have a large amount of features and expect only a few of them to be important, Lasso might be a better choice.
- Similarly, if you would like to have a model that is easy to interpret, Lasso will provide a model that is easier to understand, as it will select only a subset of the input features.

Ridge



Lasso



References

- A. Ng. Machine Learning, Lecture Notes.
- I. Goodfellow, Y. Bengio and A. Courville, “Deep Learning”, 2016.