VARIANCE REDUCTION

LECTURE 3
SECTION 3
JUNE 5TH

INSTITUTE FOR APPLIED
COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY

UNIVERSITY of
RWANDA

MOTIVATION

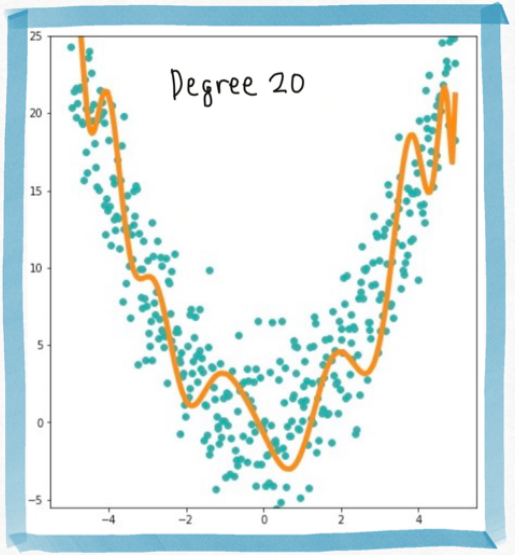
VARIANCE REDUCTION:

From our decomposition of generalization error:

$$\mathop{\mathbb{E}}_{(x,y)\sim P(x,y)}\left[\mathop{\mathbb{E}}_{D\sim P(D)}\left[(y-f_{\hat{w}}(x))^2\right]\right] = \mathop{\mathbb{E}}_{(x,y)\sim P(x,y)}\Big[\underbrace{(f_w(x)-\mathbb{E}[f_{\hat{w}}(x)])^2}_{\text{Bias}} + \underbrace{\text{Var}[\varepsilon]}_{\sigma^2} + \underbrace{\text{Var}[f_{\hat{w}}(x)]}_{\text{Variance}}\Big]$$

we know that using complex functions decreases bias but increases variance and therefore may increase generalization error.

We will study two methods that reduces the variance of complex models.



Low bias: the model is complex enough to capture the trend (no underfitting)

High noise: the noise level is high but we can't reduce this source of error.

High variance: the model is too complex and overfits to the noise. model parameters and predictions will vary a lot when we resample the training set D.

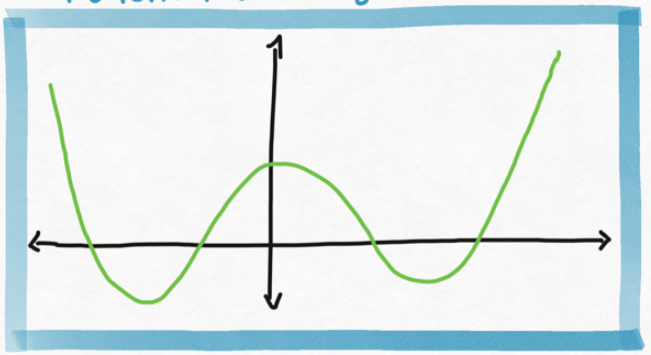
The overall error will be high unless we reduce variance.

REGULARIZATION

REGULARIZATION:

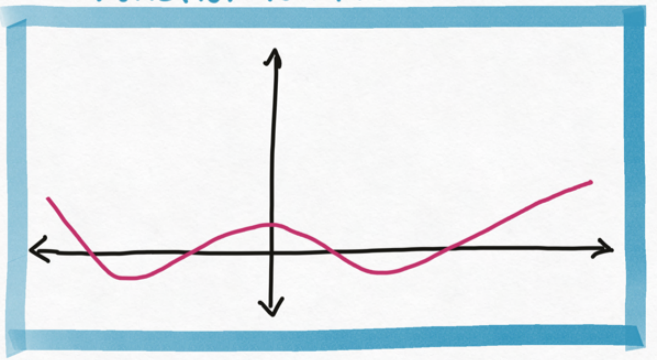
Models that have high variance overfit to the noise in the data, in order to do so, the parameters need to take on very large positive values or very negative values.

Function with high variation



$y = 0.1x^4 + 0.1x^3 - 12x^2 - 10x + 200$

Function with Low variation



$y = 0.001x^4 + 0.001x^3 - 0.12x^2 - 0.1x + 2$

If we want to discourage unnecessary variations we need to penalize large values in the parameters.

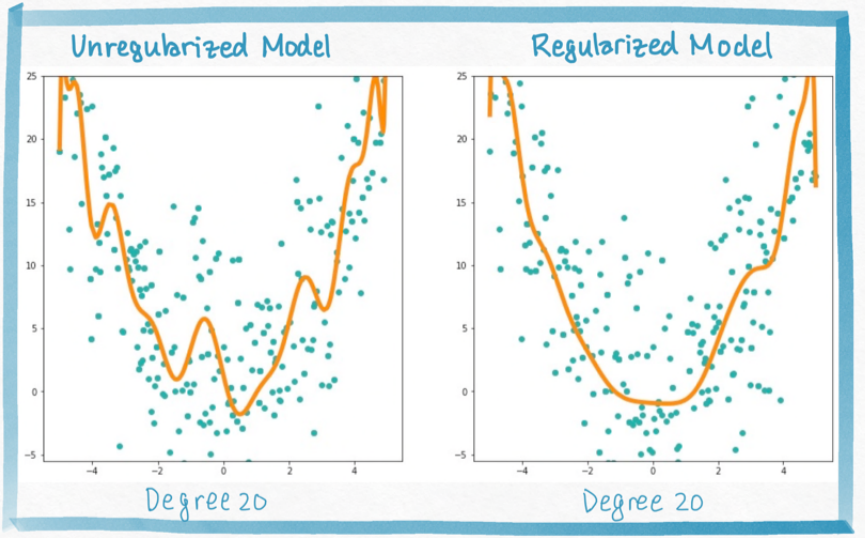
L_p REGULARIZATION:

We want our model $f_{\hat{w}}(x)$ to simulataneously have low MSE on the training set (fit the training data well) and we want the parameters $\hat{w}$ to have small values (to prevent overfitting).

So we train with the objective:

$$\hat{w} = \arg\min_{w} \text{MSE}(w) + \underbrace{\lambda \|w\|_2}_{\text{regularization term}}, \quad \|w\|_2 = \sum_{d=1}^{D} w_d^2, \quad \lambda \in \mathbb{R}$$

Training with this objective is called l_2 or Ridge Regression.



TUNING THE REGULARIZATION PARAMETER:

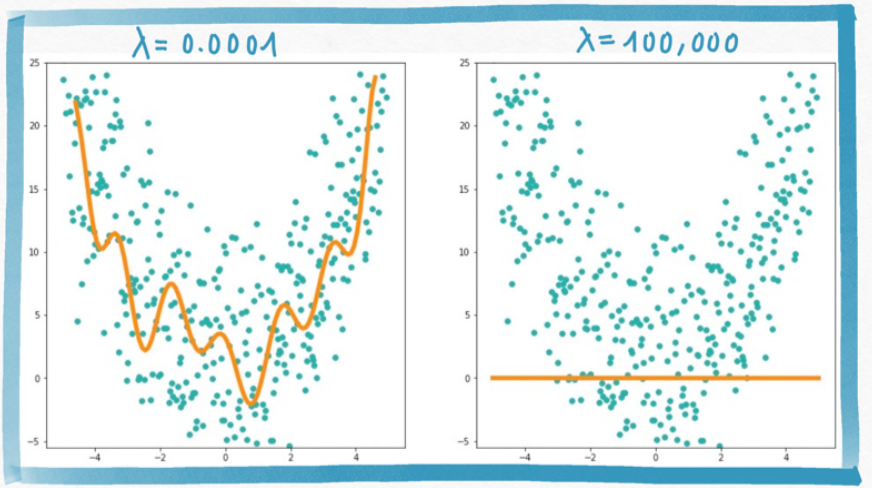
The constant λ in the Ridge objective must be picked before traing starts and controls how much we want to penalize large parameter values.

$$\hat{W} = \arg\min_{W} MSE(W) + \underbrace{\lambda \|W\|_2}_{\text{regularization term}}$$

The constant λ is called a hyperparameter, a parameter that must be fixed before training starts.

When λ is set to be too small, the regularization penalty has no effect.

The function overfits.



When the regularization term is set to be too large, the penalty term overpowers the MSE term.

The function doesn't attempt to fit the data. All the parameters go to zero to minimize the MSE term.

THE PROBLEM WITH RIDGE REGRESSION:

From Lecture #2 we know that minimizing MSE is equivalent to maximizing log-likelihood of the training data. Thus, linear and polynomial have a probabilistic interpretation.

Probabilistic models explicitly models the noise (ε). So we can explicitly check if the data satisfy our modeling assumptions. For example, we can check that our residuals are distributed like ε.

Since Ridge Regression optimizes MSE with an extra term, the objective is no longer equivalent to the maximum likelihood objective!

$$\hat{W} = \arg\min_W \underbrace{MSE(W)}_{\text{related to log-likelihood}} + \underbrace{\lambda \|W\|_2}_{\text{not related to log-likelihood}}$$

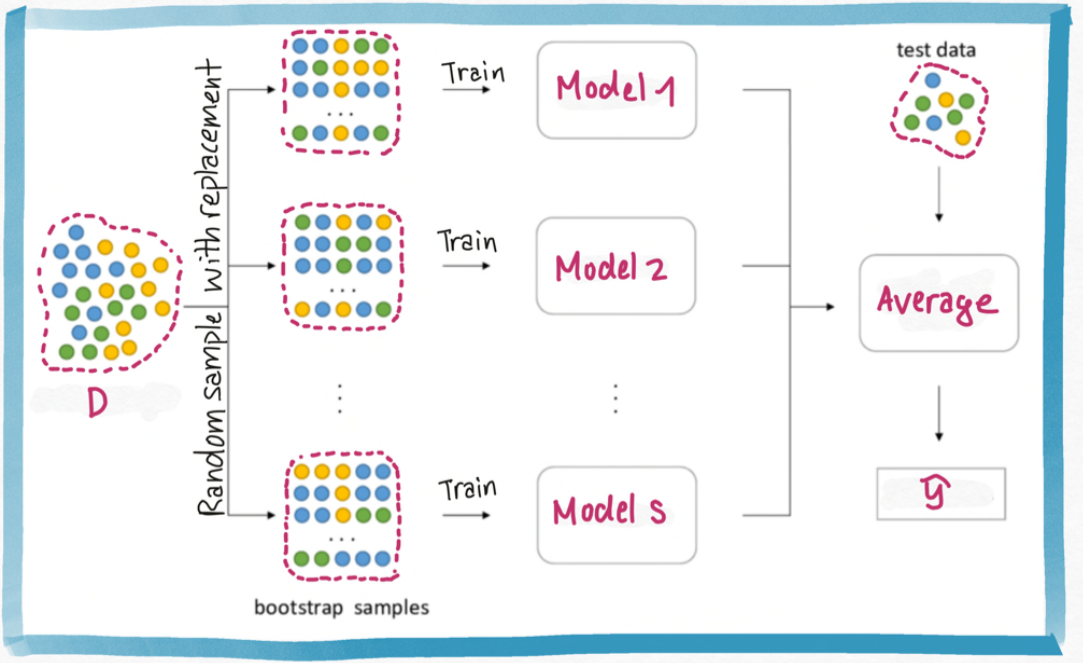
Later, we will give a probabilistic interpretation of Ridge Regression.

ENSEMBLING

ENSEMBLING:

Another way to reduce the variance of complex models is to ensemble them: we train a number of bootstrap complex models and then average their predictions.

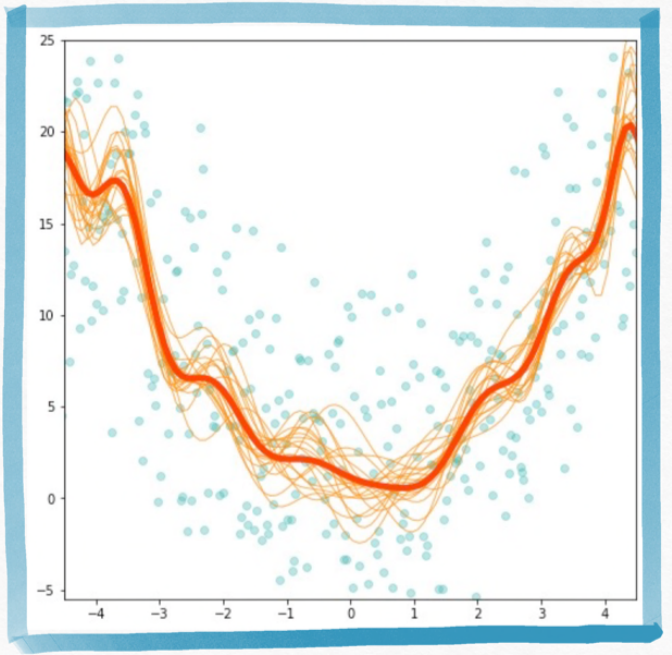
This method is called Bagging (Bootstrap Aggregating).



WHY DOES BAGGING REDUCE VARIANCE?

When we fit complex models, they will capture the noise in the data: sometimes the model will predict values that are higher than it should, sometimes the model will predict values that are lower than it should.

But when we average their predictions, these errors (too high and too low) will cancel out.



/ Bootstrap models

/ Average prediction of the ensemble