Fundamentals of Machine Learning

Mohammad Emtiyaz Khan
EPFL

Aug 25, 2015

Contents

List of concepts

1 Course Goals

Understand (some of) the fundamentals of Machine learning.

Understand the basic set-up to analyze data under a machine-learning framework.

1. Regression.
2. Model: Linear Regression.
3. Cost Function: MSE.
4. Algorithm: Gradient Descent.
5. Classification.

Understand what can go wrong when learning from data and how to correct it.

6. Overfitting.
7. Linear basis function model.
8. Regularization and CV.
9. Bias-Variance Decomposition.

2 Regression

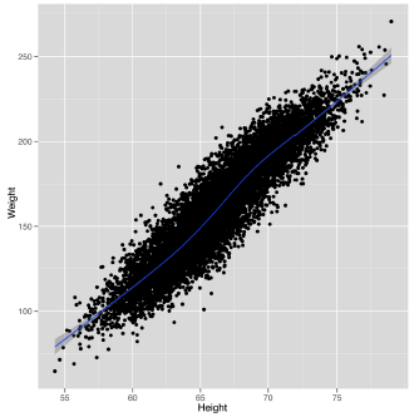
What is regression?

Regression is to relate input variables to the output variable, to either predict outputs for new inputs and/or to understand the effect of the input on the output.

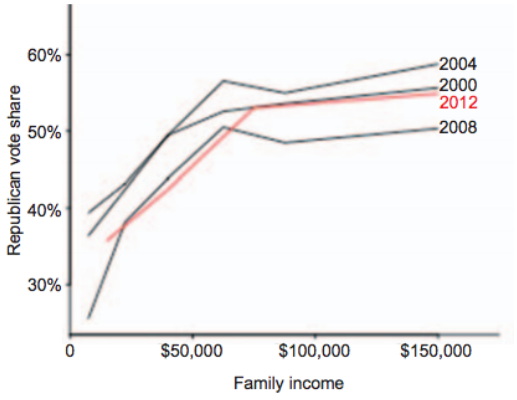
Dataset for regression

In regression, data consists of pairs $(y_n, \mathbf{x}_n)$, where y_n is the n'th output and $\mathbf{x}_n$ is a vector of D inputs. Number of pairs N is the data-size and D is the dimensionality.

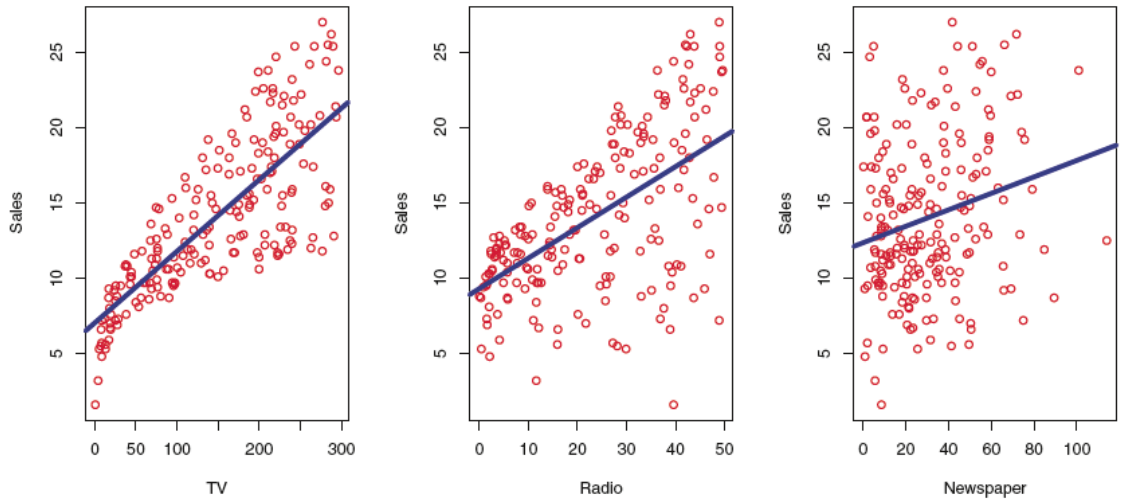
Examples of regression



(a) Height is correlated with weight. Taken from "Machine Learning for Hackers"



(b) Do rich people vote for republicans? Taken from Avi Feller et. al. 2013, Red state/blue state in 2012 elections.



(c) How does advertisement in TV, radio, and newspaper affect sales? Taken from the book "An Introduction to statistical learning"

Two goals of regression

In prediction, we wish to predict the output for a new input vector, e.g. what is the weight of a person who is 170 cm tall?

In interpretation, we wish to understand the effect of inputs on output, e.g. are taller people heavier too?

The regression function

For both the goals, we need to find a function that approximates the output "well enough" given inputs.

$$y_n \approx f(\mathbf{x}_n), \text{ for all } n$$

Correlation $\neq$ Causation

Regression finds correlation not a causal relationship, so interpret your results with caution.

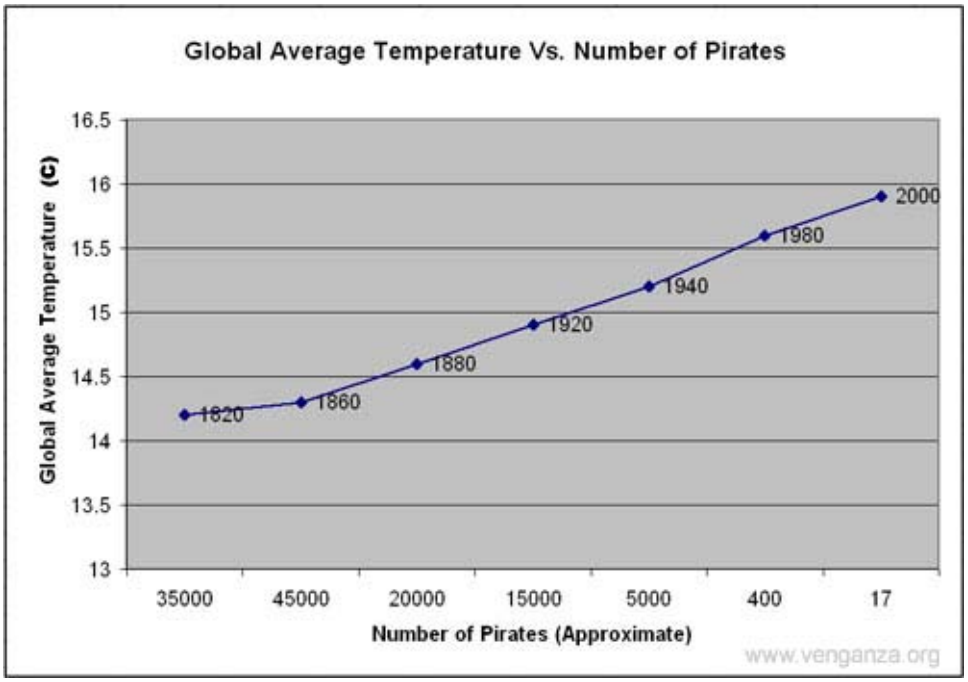


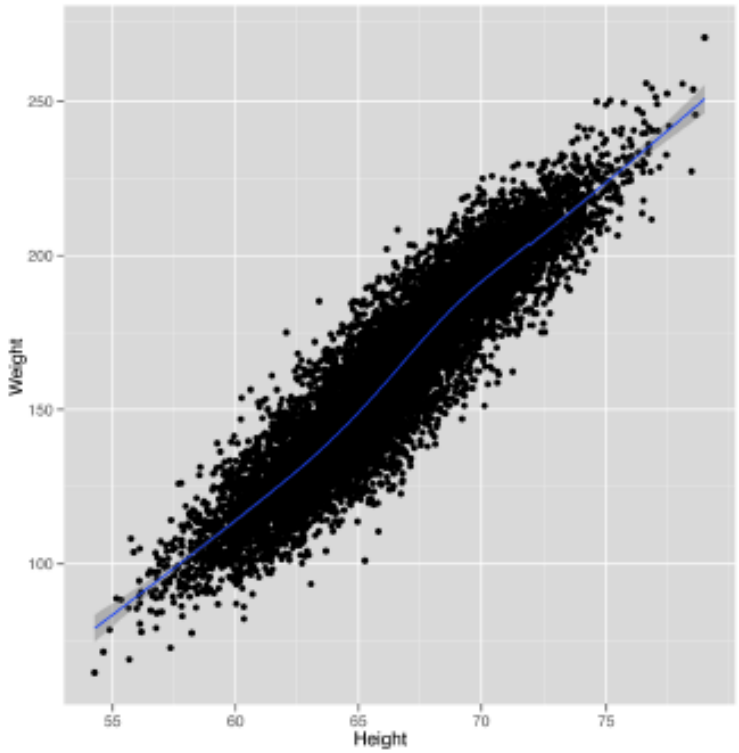
Image taken from http://www.venganza.org/.

Also see Spurious correlations page.

3 Model: Linear Regression

What is it?

Linear regression is a model that assumes a linear relationship between inputs and the output.



Why learn about *linear* regression?

Plenty of reasons: simple, easy to understand, most widely used, easily generalized to non-linear models. Most importantly, you can learn almost all fundamental concepts of ML with regression alone.

Simple linear regression

With only one input dimension, it is simple linear regression.

$$y_n \approx f(x_n) := \beta_0 + \beta_1 x_n$$

Here, β_0 and β_1 are parameters of the model.

Multiple linear regression

With multiple input dimension, it is multiple linear regression.

$$y_n \approx f(x_n) := \beta_0 + \beta_1 x_{n1} + \ldots + \beta_D x_{nD}$$

Learning/estimation

Given data, we would like to find $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_D]$. This is called learning or estimating the parameters.

---

4 Cost Function: MSE

What is a cost function?

Cost functions (or utilities or energy) are used to learn parameters that explain the data well. They define how costly our mistakes are.

Two desirable properties of a cost function

When y is real-valued, it is desirable that the cost is symmetric around 0, since both +ve and -ve errors should be penalized equally.

Also, our cost function should penalize "large" mistakes and "very-large" mistakes almost equally.

Mean square error

It is the most popular choice.

$$MSE := \frac{1}{2N} \sum_{n=1}^{N} [y_n - f(\mathbf{x}_n)]^2$$

An example for $f(\mathbf{x}_n) = \beta_0$.

Convexity

A function is convex iff a line joining two points never intersect with the function anywhere else.

Importance of convexity

A convex function has only one global minimum value.

Sum of convex functions is also convex. Therefore, MSE has only one global minimum value.

Advanced topics

Outliers and other cost functions, Statistical vs computational trade-off.

5 Algorithm: Gradient Descent

Learning/estimation/fitting

The goal is to find the parameter with minimum cost.

$$\min \sum_n [y_n - \beta_0]^2$$

This is an optimization problem.

Grid search

Grid search is the simplest algorithm where we compute cost over a grid to find the minimum.

But, for large number of parameters, grid search has many "for-loops". This implies exponential computational complexity.

Follow the gradient

A gradient (at a point) is the slope of the tangent (at that point). It points to the direction of largest increase of the function.

Batch gradient-descent

Take a step in the direction of the gradient

$$\boldsymbol{\beta}^{k+1} \leftarrow \boldsymbol{\beta}^k + \alpha \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{\beta}} MSE_n(\boldsymbol{\beta}^k)$$

where $\alpha > 0$ is the step-size (or learning rate).

When to stop? When $\mathbf{g}$ is (close to) zero. If second-order derivative is positive, it is a minimum.

Stochastic grad-descent

When N is large, choose a random pair i and take a step.

$$\boldsymbol{\beta}^{k+1} \leftarrow \boldsymbol{\beta}^k + \alpha_i \frac{\partial}{\partial \boldsymbol{\beta}} MSE_i(\boldsymbol{\beta}^k)$$

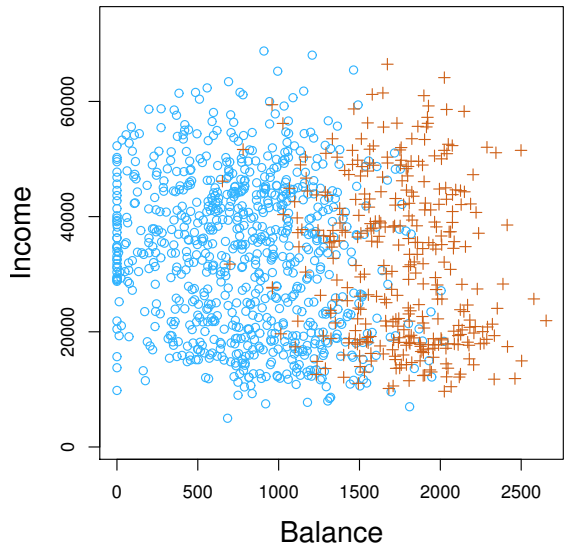
For convergence, decay $\alpha_i \to 0$ using Robbins-Monroe construction.

Least-squares

Least-squares uses the closed-form expression for the minimum. It is also related to maximum likelihood.

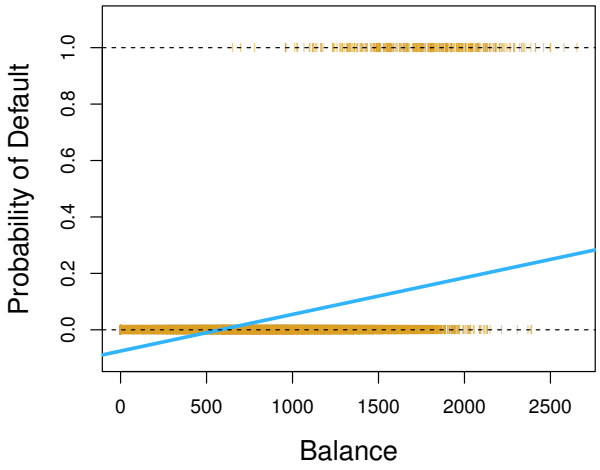
6 Classification

Classification is same as regression but now y_n is binary. Examples: fraud detection, face detection.



Model: Lin Reg

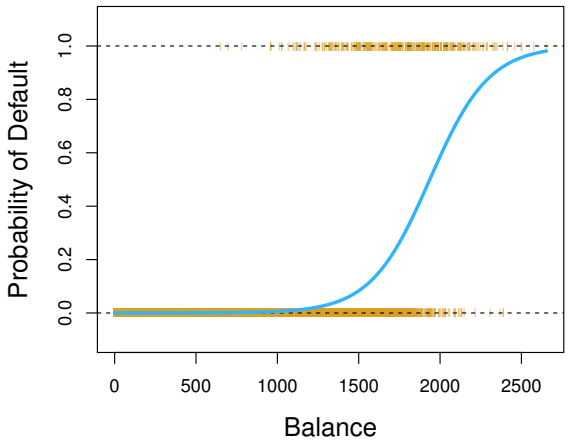
Predict y_n using linear regression. If $f(\mathbf{x}_n) > 0.5$, $y_n = 1$ or else 0.



This is not a good model since it ignores that y_n is binary.

Model: Logistic Reg

We can map $f(\mathbf{x}_n)$ to a value between 0 and 1. We can treat this value as "the probability of $y_n = 1$".



We will use logistic function for this.

$$\sigma(f) = 1/[1 + \exp(-f)]$$

Cost: Log-Likelihood

Instead of minimizing mistakes, we can maximize the likelihood.

$$\max \sum_n y_n \log[1 + \exp(-f_n)] + (1 - y_n) \log[1 + \exp(f_n)]$$

In fact, MSE == maximum likelihood (ML) for linear regression.

7 Overfitting

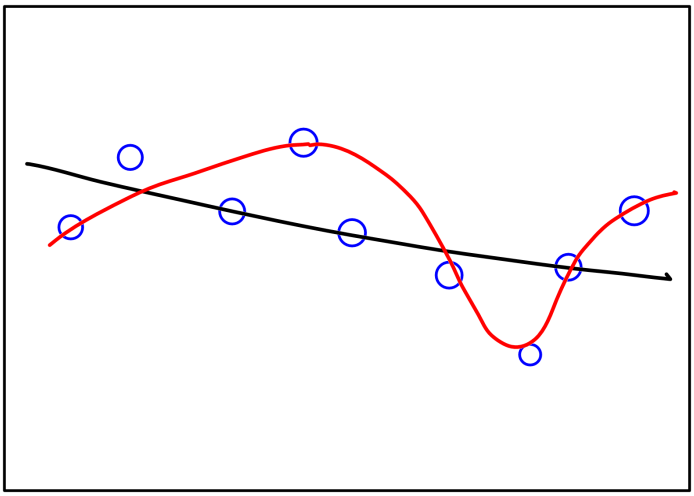
What is Overfitting?

Overfitting is fitting the noise in addition to the signal. Underfitting is not fitting the signal well.

In reality, it is very difficult to be able to tell the signal from the noise.

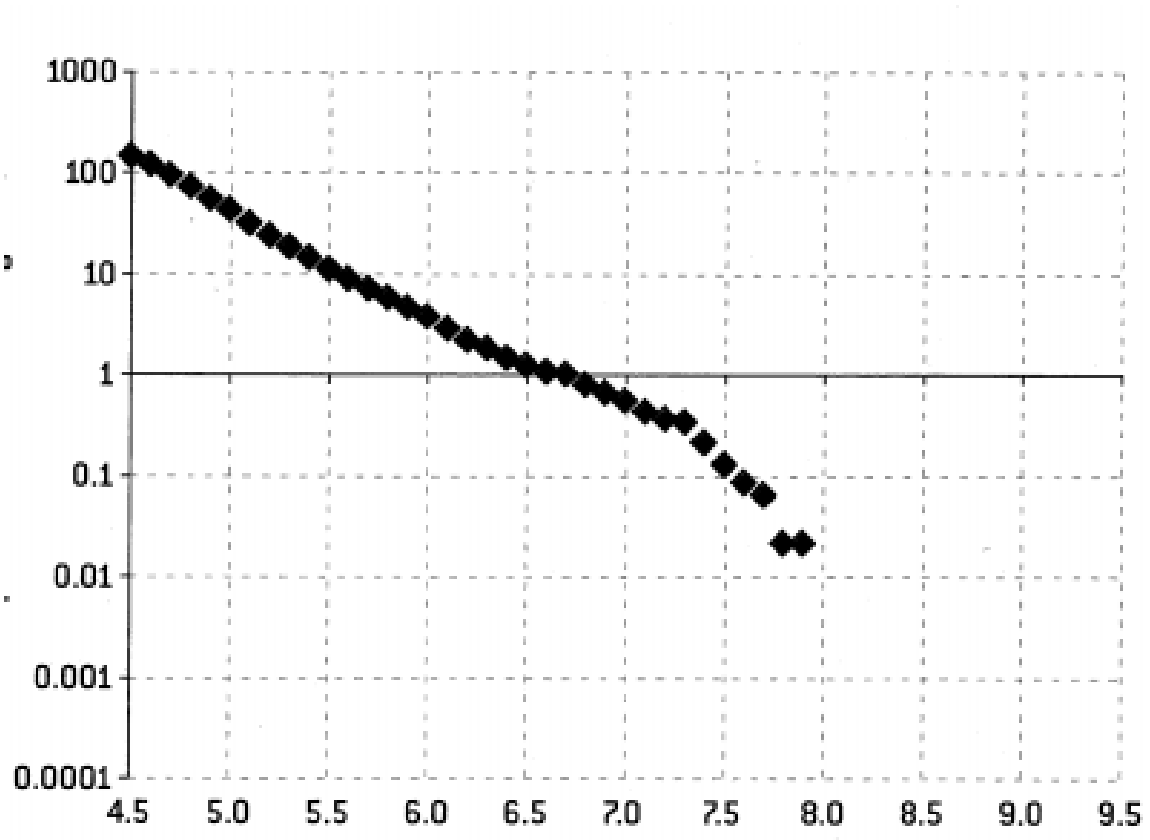
Which is a better fit?

Data is denoted by circle. Which model is a better fit? black or red?



Which is a better fit?

Try a real situation. Below, y-axis is the frequency of an event and x-axis is the magnitude. It is clear that as magnitude increases, frequency decreases. Which model is better fit? blue or red?



Occam's razor

One solution is dictated by Occam's razor which states that "Simpler models are better - in absence of certainty."

Sometimes, if you increase the amount of data, you might reduce overfitting. But, when unsure, choose a simple model over a complicated one.

But how do we do this in practice?

8 Linear basis function model

Why basis function?

Linear model can be too limited and usually underfit. One way is to use "non-linear" basis functions instead.

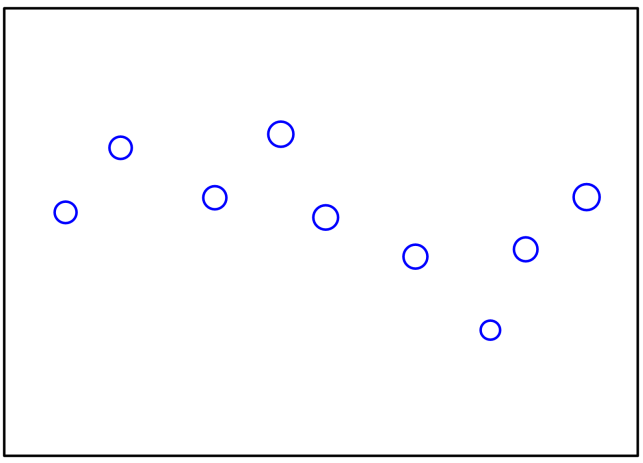
Polynomial basis

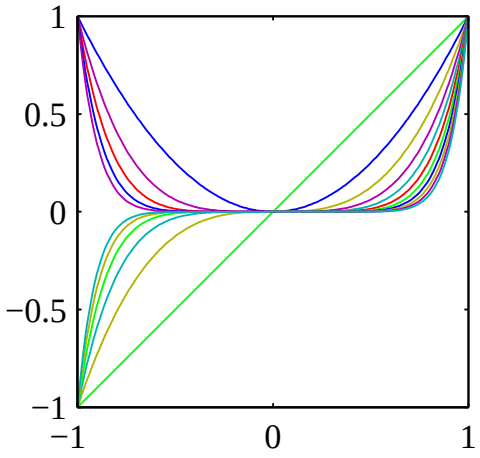
Consider simple linear regression. Given one dimensional input x_n, we can generate a polynomial basis.

$$\boldsymbol{\phi}(x_n) = [1,\, x_n,\, x_n^2,\, x_n^3,\, \ldots,\, x_M^3]$$

Then we fit a linear model

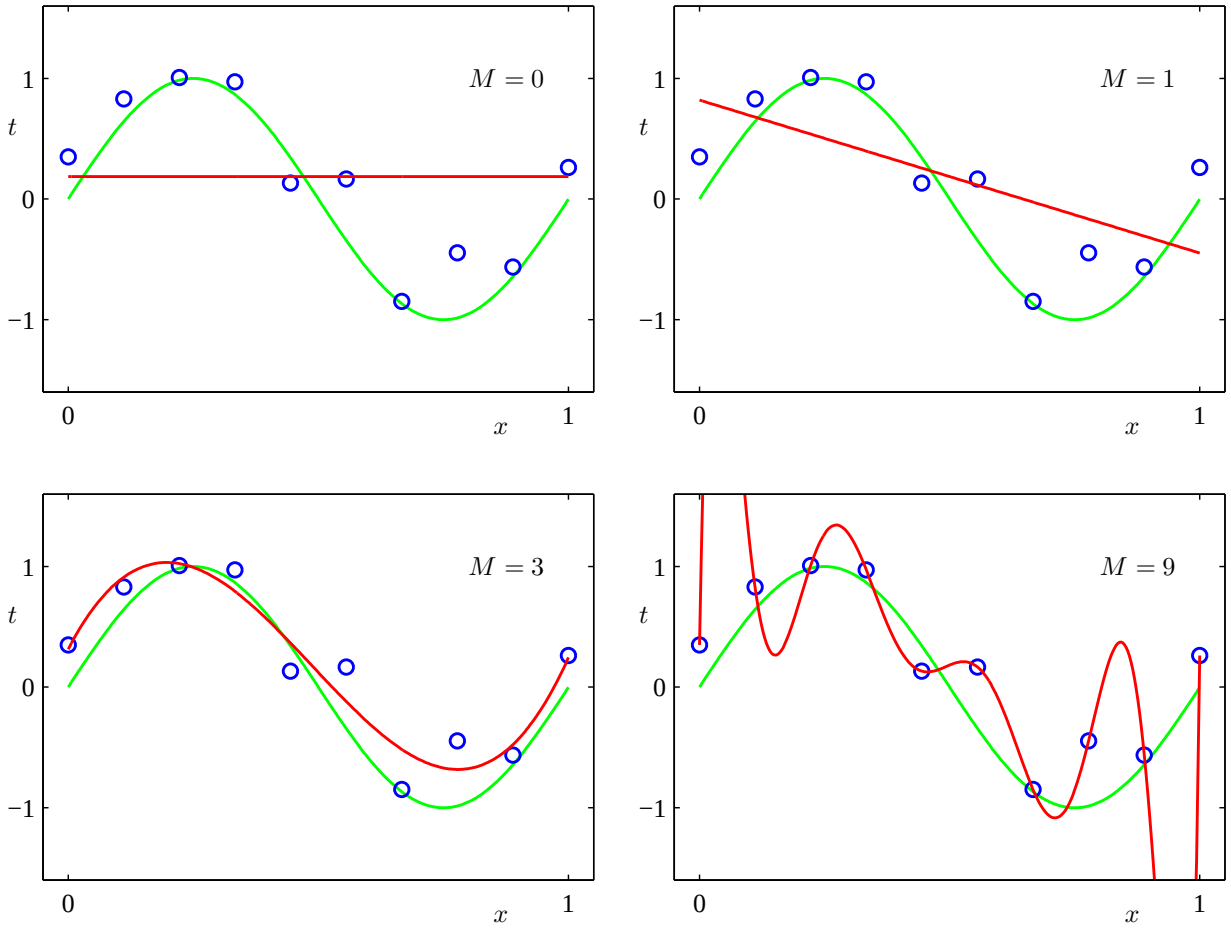
$$y_n \approx \beta_0 + \beta_1 x_n + \beta_2 x_n^2 + \ldots + \beta_M x_M^3$$



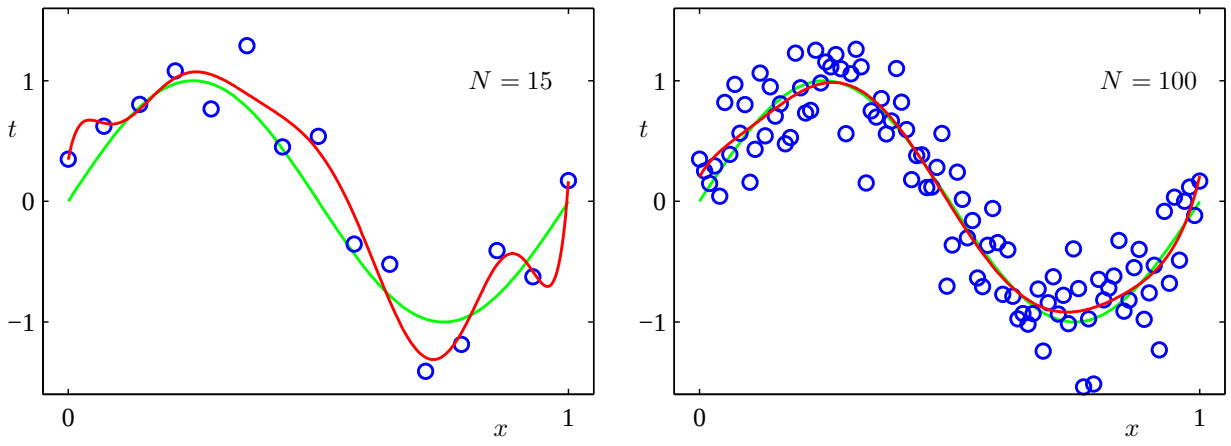


Complex model overfit easily.

Circles are data, Green is truth & red is the fit.



If you increase the amount of data, overfitting *might* reduce.



Question: How do you control the model complexity?

9 Regularization and CV

What is regularization?

We can reduce model complexity by forcing many β_i to be zero. Regularization achieves this by penalizing high values of β_i.

$$\min \sum_{n=1}^{N} [y_n - f(\mathbf{x}_n)]^2 + \lambda \sum_{i=1}^{M} \beta_i^2$$

where second term is the regularizer, $\lambda > 0$ (regularization parameter).

RegularizationParameter

The parameter λ can be tuned to reduce overfitting by reducing model complexity. But, how do you choose λ?

Simulating the future

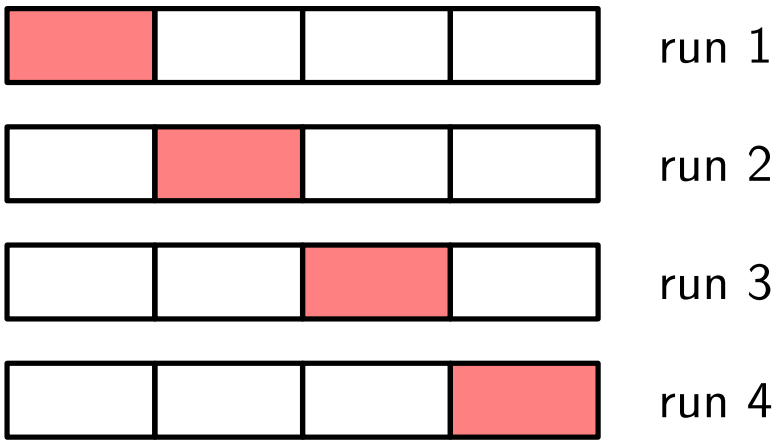
Ideally, we should choose λ to reduce error in our future predictions, i.e. on the data that we have not seen yet. Unfortunately, we do not have the future dataset. However, we can *simulate the future* using data in hand.

Splitting the data

We split the data into train and validation sets, e.g. 80% into training data and 20% as validation data. We can then train on first set and predict the rest to compute test-MSE. This gives us one instant of the future. We can repeat this process many times.

Cross-validation

K-fold cross-validation allows us to do this efficiently. We partition the data into K groups. We train on $K - 1$ groups and test on remaining groups. We repeat this until we have tested on all K sets. We then average the results.



Cross-validation returns an estimate of the *generalization error*.

10 Bias-Variance Decomposition

What is bias-variance?

One natural question is how does the test error vary wrt λ? When λ is high, the model underfits, while when λ is small, the model overfits. Therefore, a good value is somewhere in between.

Bias-variance decomposition explains the shape of this curve.

Generalization error

Given training data $\mathcal{D}_{tr}$ of size N, we would like to estimate the expected error made in future prediction. This error is the generalization error. Below is a definition suppose that we have infinite test data $\mathcal{D}_{te}$,

$$teErr(\mathcal{D}_{tr}) := \mathbb{E}_{\mathcal{D}_{te}}[\{y_* - f(\mathbf{x}_*)\}^2]$$

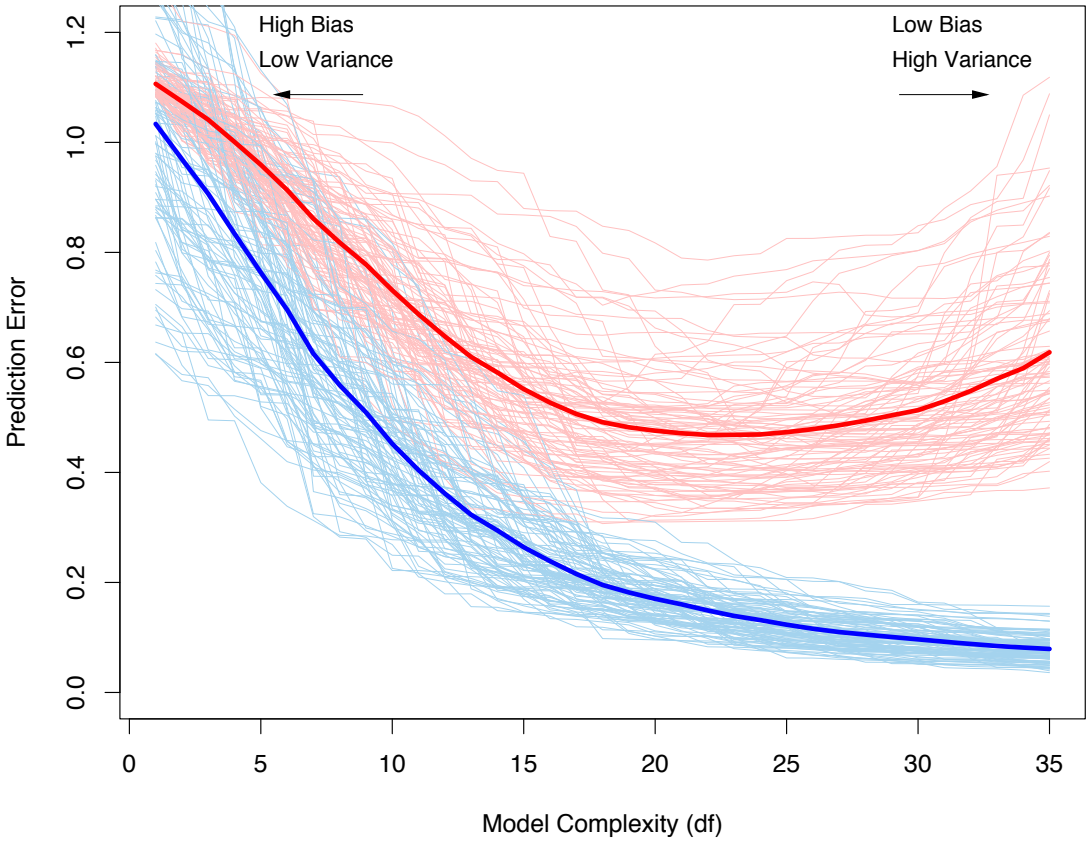
Generalization error is different from the training error which measures how well you fit the data.

$$trErr(\mathcal{D}_{tr}) := \sum_{n=1}^{N}[\{y_n - f(\mathbf{x}_n)\}^2]$$

Errors vs model complexity

As we increase the model complexity, how do these errors vary? The blue line shows training error for a dataset with $N = 50$, while the red line shows the generalization error for that dataset.

Simple model have high train and generalization error since they have a high bias, while complex model have low train but high generalization error because they have high variance.



Bias-variance decomposition

The shape of these curves can be explained using bias-variance decomposition which says that both bias and variance contribute to generalization error. For bias, both model-bias and estimation-bias are important. When we increase model complexity, we increase generalization error due to increased variance.

Regularization increases estimation bias while reducing variance.

Comments about CV

Cross validation estimates both expected train and test error. When learning curve is steep, then CV overestimates the true objective function.