Aaron Mathankeri

Fundamentals of Machine Learning

Adapted from: Patterns of Recognition
Machine Learning: a Probabilistic Perspective
Elements of Statistical Learning

Today we will analyze a regression problem to introduce fundamental concepts in machine learning. You will see that very complex, sophisticated techniques are typically extensions of these core ideas. We will begin with a brief introduction to **likelihood**.

LIKELIHOOD |

In both probability paradigms (Bayesian and frequentist), the likelihood function plays a central role. In Machine Learning, we use the likelihood as a way to estimate our model. Formally, the likelihood is defined as,

Probability of the Data given our model
P(D | M)

Intuitively, we are quantifying a simple question : what are the chances our guess(model) can produce the data we are observing?

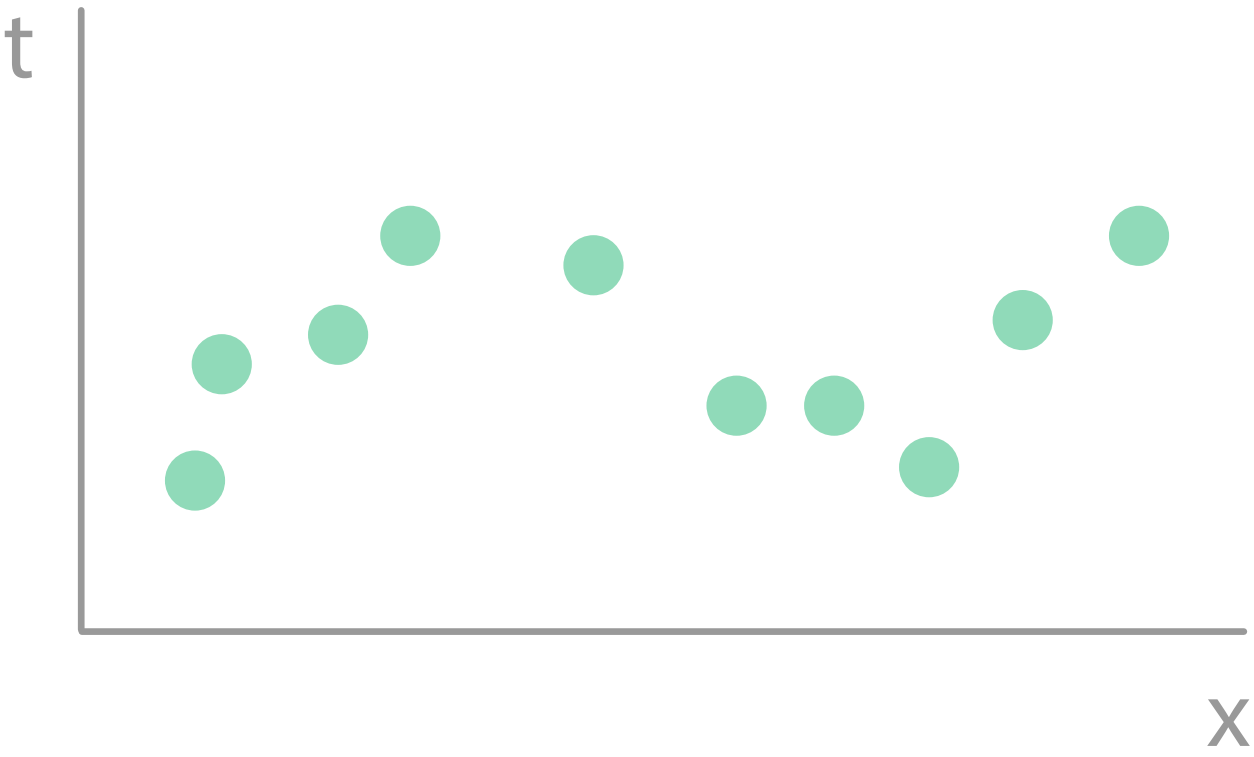
Question:
What is maximum likelihood? How is it defined? What are the advantages and disadvantages of maximum likelihood?

This question, at its core, is what we are trying to answer. Can we come up with a model that explains the data? We will explore this question with the following regression example.

Polynomial Curve Fitting |

Now suppose we have an input variable, **x**, and wish to use this observation to predict a target variable, **t**. We have a set of data that for every **x** gives us a **t**, as shown in the plot in Figure 1. The million dollar question in Machine Learning is this: can we predict **t** for new values of **x**?

Figure 1 |
Plot of input **x** and target **t**



You can see that the data seems to have some structure. The data possesses and underlying regularity, which we wish to learn, however the observations are corrupted with random noise.

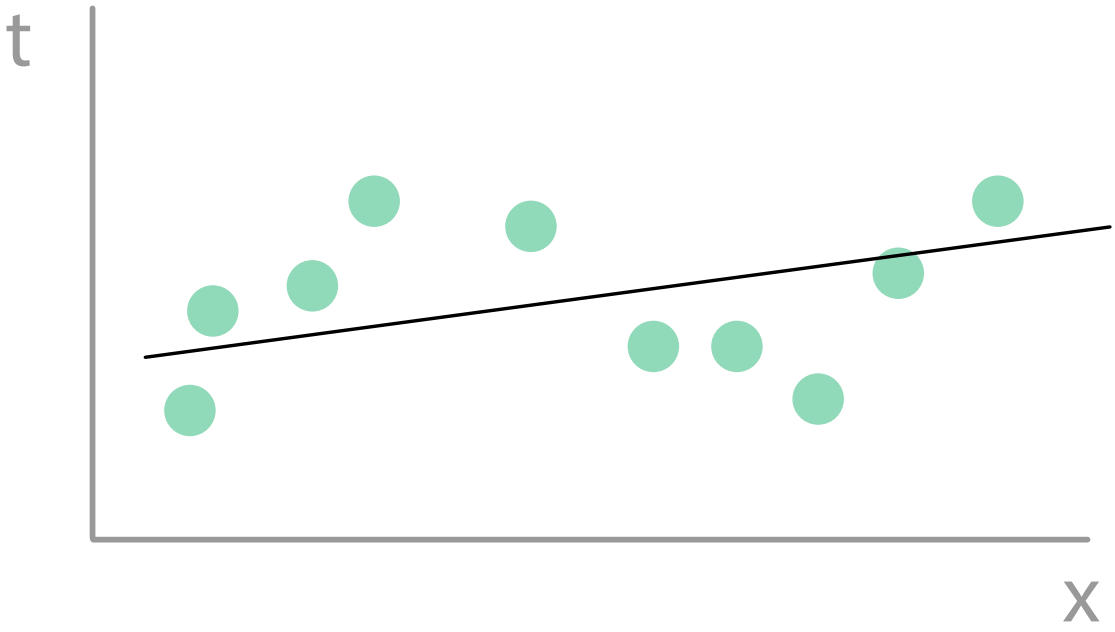
We've seen in previous weeks how to fit data to linear regressions, so let's try that here. Remember, the formula of linear regression is typically:

t = y(x, w) + error

y(x, w) is just a way to map inputs, x, to outputs t. It is parameterized by "w". For example, a line of best fit would have the form,

$y(x,w) = w_0 + w_1 x$

Figure 2 |
Fitting data to linear regression



This is an embarrassing attempt at modelling this data, so lets try to fit this to a polynomial. The mapping function now has the form,

$$y(x, w) = w_0 x^0 + w_1 x^1 + w_2 x^2 + \ldots = \sum_i^M w_i x^i$$

Here M is the order or the polynomial, which means it governs how complex our model will be.

Question
This is still called a linear regression, why? What are basis functions? What is the basis functions for the example above?

How do we determine what the values of the coefficients, **w**, are? We minimize an error function. A simple choice is the sum of squares error function,

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

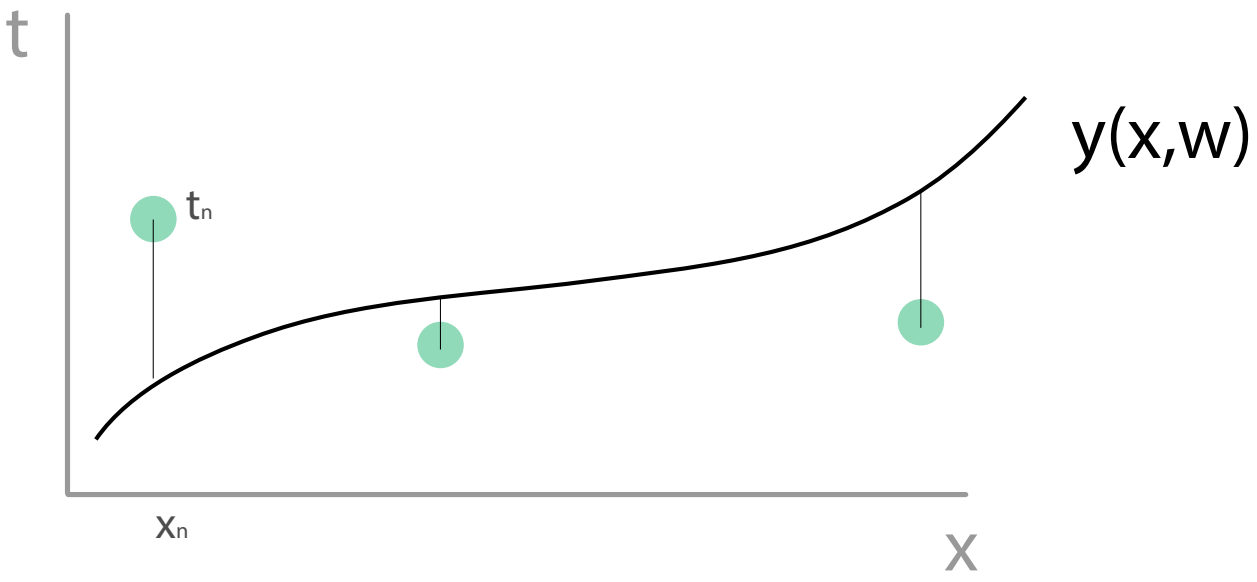
Question
Where does this come from?

Bonus
Derive the previous result. Hint: assume likelihood is normally distributed, with mean = y(x, w), and take log of likelihood

We of course want to minimize the error, so we need to choose **w** for which E(w) is the smallest.

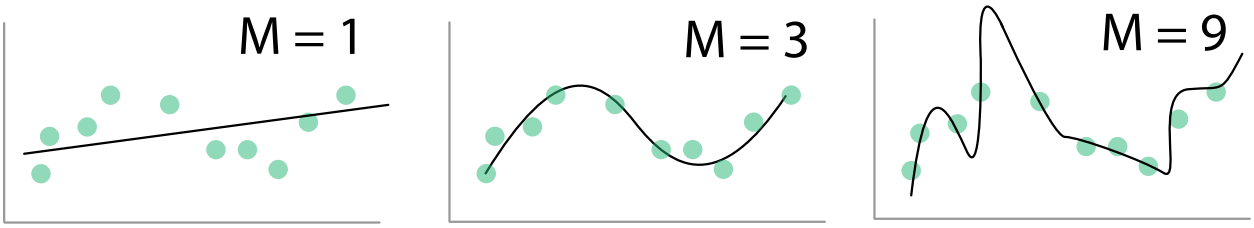
Question
Find minimum of E(w) with respect to w.

Figure 3 |
This error function tries to minimize the length of the vertical bars.



Now, which order of M should we choose? How complex should our model be? [CONCEPT: Model Selection]
Let's look at a few different choices...

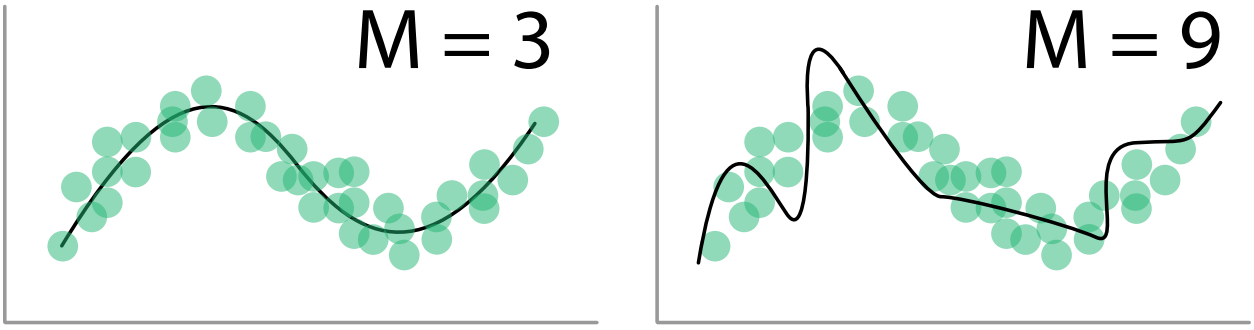
Figure 4 | different M values



M =1 is a pathetic fit. M = 3 gives us a great fit. But M =9 is an excellent fit. In fact it goes through every single point.

What happens when we introduce new data?

Figure 5 | M = 3 and M = 9



Which one is a better fit now? The higher order polynomial doesn't have the success it did with the training data. This behaviour is known as over-fitting, and will plague us throughout our careers.

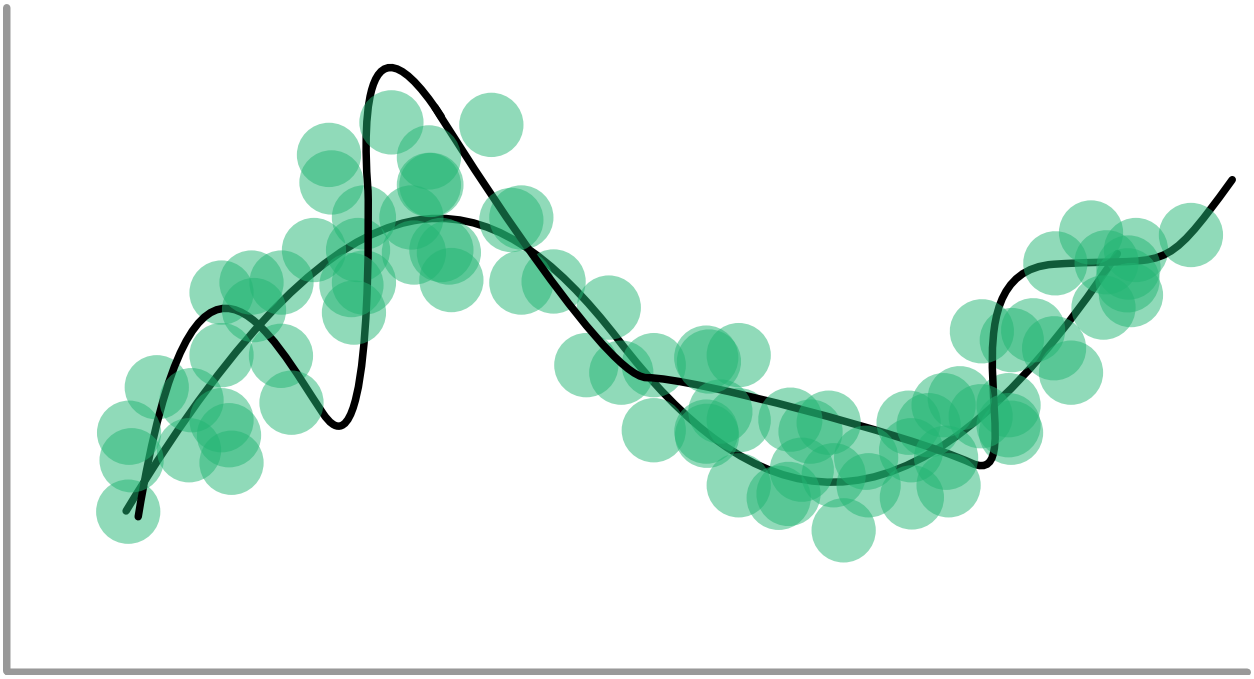
GOAL: Achieve generalization by making accurate predictions for new data

This result in figure 5 may seem paradoxical to the math savvy.

Question:
Why? How is it that a polynomial of order 9 is a worse fit than a polynomial of order 3?

One of the things that you will notice is that larger datasets allow complex models to fit the data.

Figure 6 | data points with M = 3 and M = 9



The more data we start off with, seems to allow a more complicated model. However once we add new data, these complicated models do not work as well. This is a very unsatisfying artifact of model fitting. The size of the data determines how many parameters we can use. It would be more appropriate to choose the complexity of the model based on the complexity of the problem, not the amount of data. This is a overfitting problem that is a property of maximum likelihood.

Question

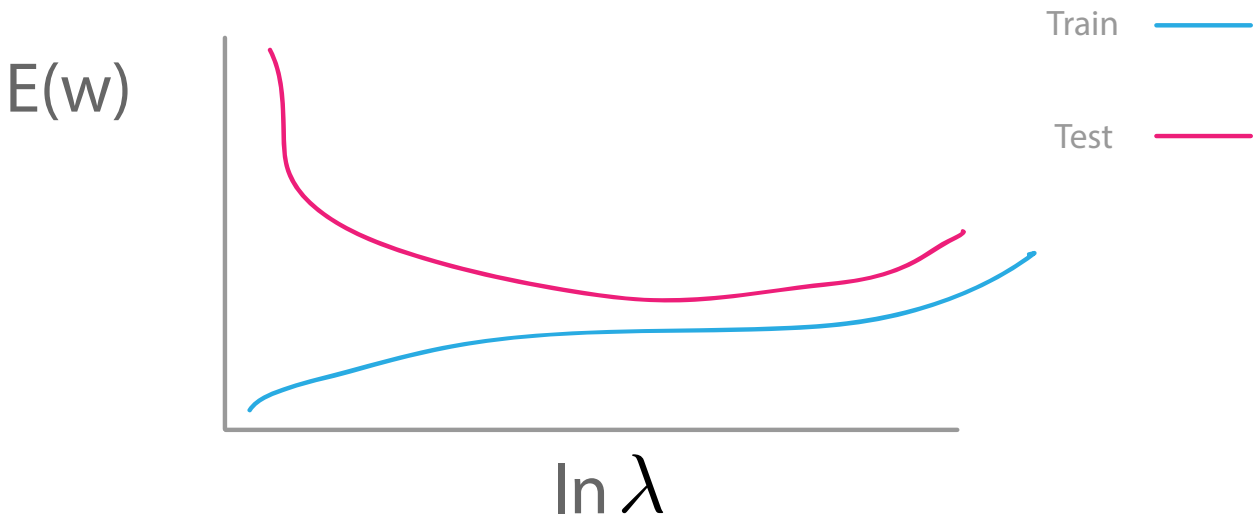
Is there a solution? Is there an alternative approach that avoids this problem?

Regularization |
A way to control this over-fitting issue is “regularization.” We introduce a penalty term into the error function to discourage large coefficient values.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\mathbf{w}^2$$

Question
What is lambda? What is its purpose?

Figure 7 | different values of lambda



Model Selection |

We’ve seen that with the maximum likelihood(maxL) approach, performance on a training set is not a good indicator on predictive performance because maxL favours excessively complex models.

If data is available, it is typically good practice to train on a portion of the data and test on the remaining portion, i.e. validation set. Cross validation, as was

mentioned last week, is a popular method for training, exploring the “leave-one-out” technique.

Question
What are the drawbacks of Cross-validation?

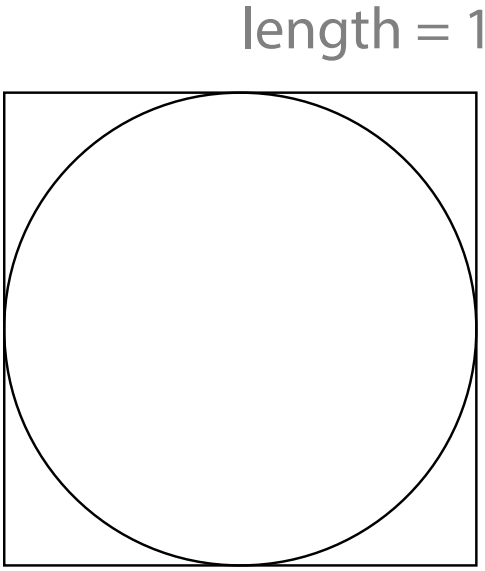
Historically, information metrics are used, e.g. AIC and BIC. They attempt to correct for ML bias by penalizing model complexity.

Question
How is AIC/BIC defined? What is the meaning of the parameters in the equations. What is the difference between the two models?

Dimensionality |

When we start dealing with data in higher dimensions, our intuition breaks down. My favourite illustration of this, is the area paradox.

Figure | Area Paradox



Question:
Where is most of the area of located? Inside or outside the circle?
What is the formula for area of a square?
What is the formula for area of circle(up to a constant)?
Generalize the formulas to just the dimensions(s).

What happens when A_{sq} for D = 10?
What happens when A_{cir} for 10 = 10?

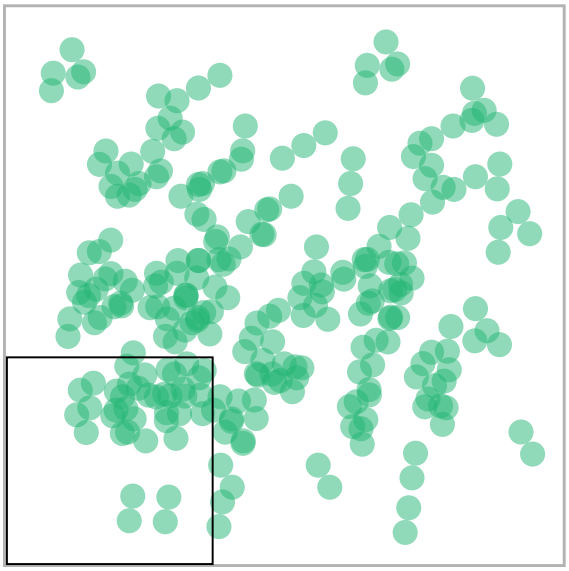
This may seem like math magic but it presents a serious challenge in Machine Learning.

Imagine we want to classify a set of data points using K-Nearest Neighbours (KNN).

Question
What is KNN?

We pick a point, “x” and we grow our square until it has at least K neighbours

Figure



In 2 dimensions, this is quite trivial. Lets explore this in 10 dimensions.

Question
If we ant to capture just 10% of the data, what fraction of the entire data space must we cover?

Hint: formula r^(1/D). Set r = .10 and D = 10.

This doesn’t prevent us to develop techniques for high dimensional input space.

1. Real data is confined to lower “effective” dimensions
2. Real data exhibits local smoothness property, meaning small changes in x lead to small changes in t