Curve fitting exercises

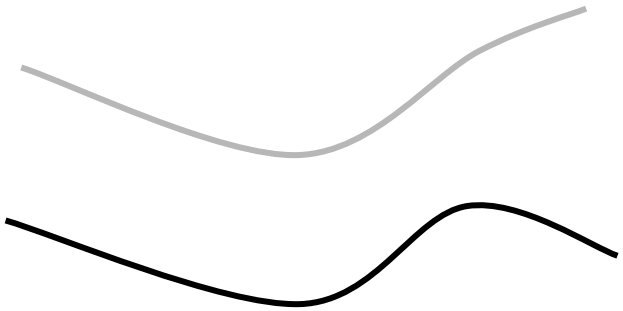
Curtin Institute for Computation

Things to do...

- Install Anaconda (Python 3)
- Python packages required:
  - Numpy
  - Scipy
  - Plotly
  - Matplotlib
  - pickle
  - Theano
  - Tensorflow
- Join CurtinIC slack channel for workshop related discussions **https://curtinic.slack.com**

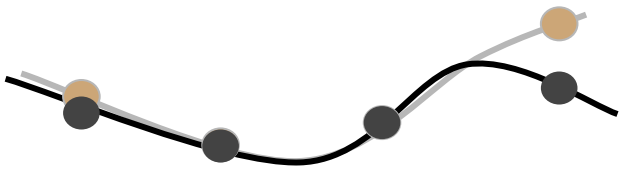
Curve Fitting 101

Is there a **quantitative** way to measure how close the two curves are:



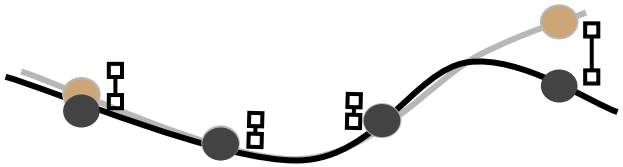
Curve Fitting 101

Now?



Curve Fitting 101

Measure the distances between two points

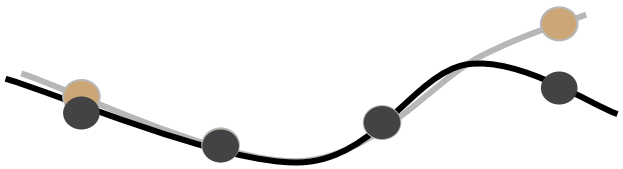


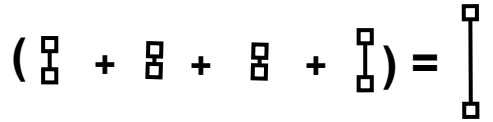
Is there a single value that I could use to compare?x

Please skim through the function - **“mse”** - below.

Curve Fitting 101

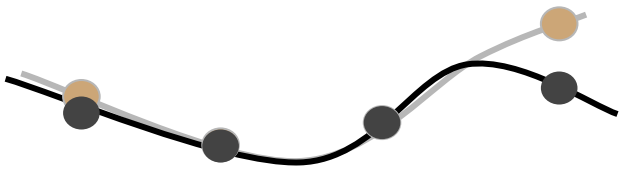
Is there a single value that I could use to compare?

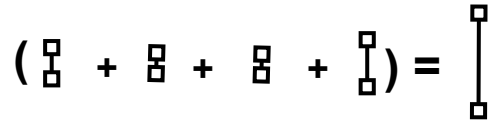




Curve Fitting 101

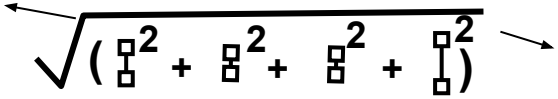
Is there a single value that I could use to compare?





Please skim through the function - **“distance”** - below.
Have a look at the hypothesis: a_0x+a_1x

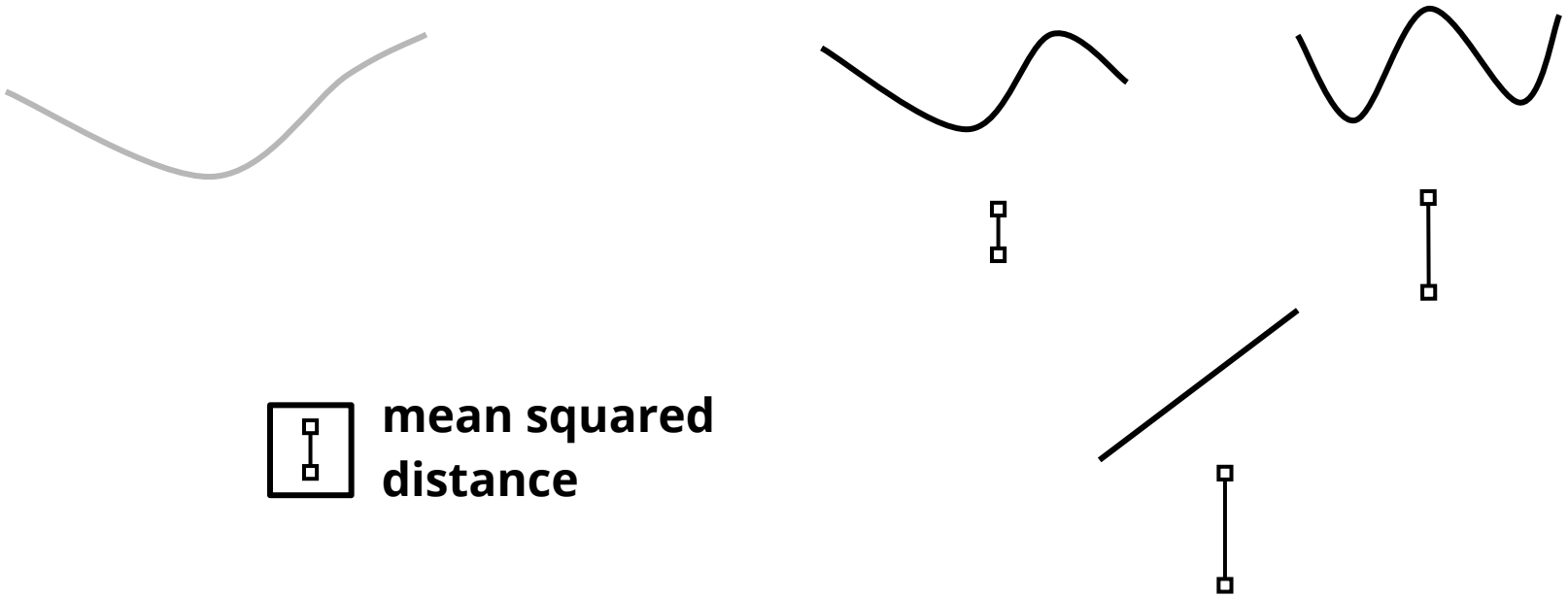
what happens when ▯ lies between (-1,1)?



to account for negative values

Curve Fitting 101

Which of these best estimate the **ground truth**?



mean squared distance

Curve Fitting 101

Now, think of a practical problem

Employment rate, stock price, or even Trump's mood swings

You would then "**hypothesiz**e" a look-alike curve that captures the trend

Curve Fitting 101 - Parametric tuning

Now, what is the best solution?

Values are not representative of the problem

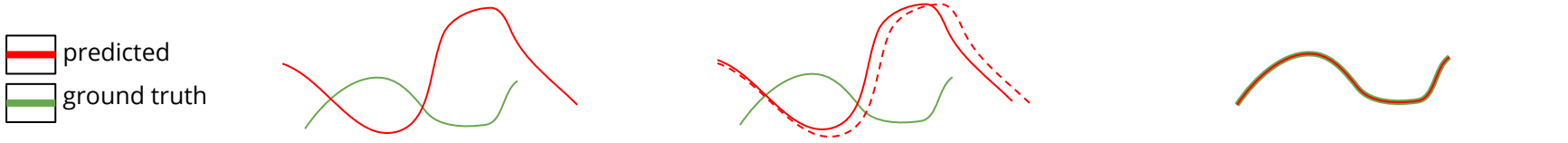
Equation	Distance
sin(1x)	18
sin(2x)	15
sin(3x)	25
sin(1.4x)	12
tan(0.2x)	1000
exp(0.4x)	40000

parameters

Curve Fitting 101 - Parametric tuning

I know that the hypothesis is of the form sin(a*x), how do I find "a"? This is when we would use an optimizer

Optimizer mechanics

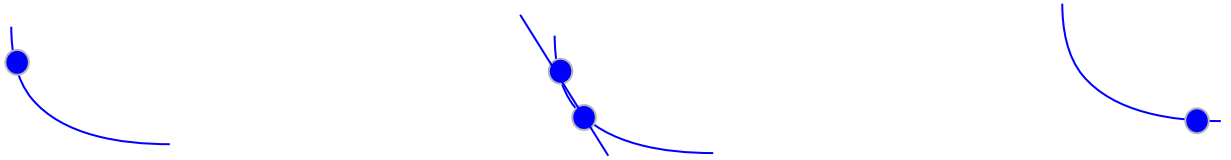


Starts with a guess, computes error (mean distance)

Increments guess, computes error, moves in the direction of the smaller of the two errors

Increments and increments until guess is almost same as ground truth

Loss and its derivatives



Code Curve Fitting 101

Give a nice hypothesis and the optimizer will find great parameters for you

If I know my hypothesis is of the type $\sin(a_0*x)+\tan(a_1*x)$ and would like to find (a_0,a_1), I have to optimize mean distance and find a_0,a_1 where the mean distance is minimum.

Sample code:

opt.minimize(*function that computes error distance, initial values of* (a_0,a_1)*, additional arguments to the function, choice of optimizer-'cg','bfgs'***)**

Exercise 1

Let us visualize the downloaded data, first.

- **Load the data file: trump.npy**
  (Useful functions: ***np.load(filename)***)
- Visualize using the function **iplot**
- Have another look at the function : **distance**, try various hypotheses (*np.sin(a_0(x))+a_1(x), np.sin(a_0(x))+exp(a_1x), and* ***what you think is the right hypothesis***)
- Check if the loss is decreasing, and also check the goodness-of-fit by moving the slider