Over Fitting, Under Fitting and Bias and Variance Trade Off

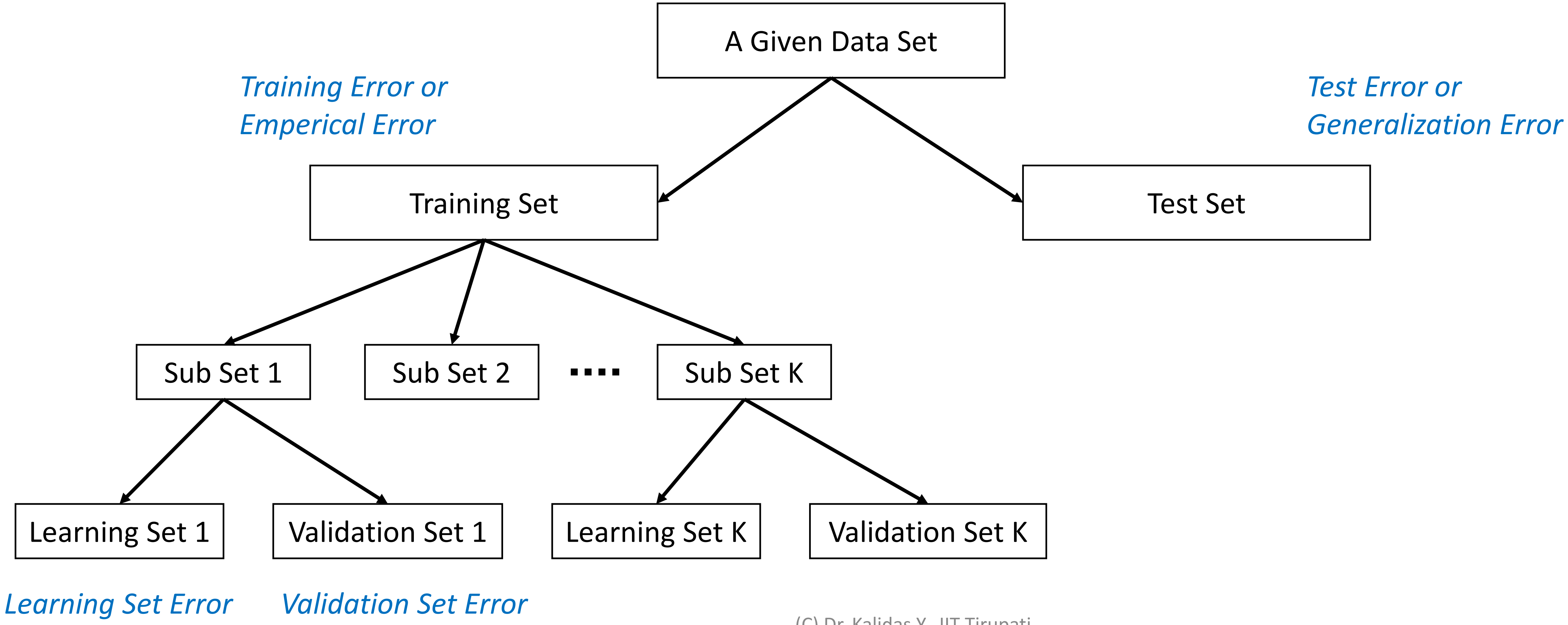
Dr. Kalidas Y., IIT Tirupati

In this lecture, you will understand over fitting and under fitting scenarios and notions of bias and variance

Motivation

- What is the best model?
  - For example, for a given set of points,
  - would **degree 0** polynomial ***fit the best***?
  - Or **degree 1** polynomial would ***fit the best***?
  - Or **degree k**?
- How do we create the 'best-*ness*' criteria?
  - Training error
  - Test error
  - Other forms?
- Which type of loss function to use
  - For example
    - Mean Squared Error
    - Mean Absolute Error
    - Others?
- Model ***Maintenance*** is easy
  - Model performance is high over different arriving data sets in production
- Model is ***too sensitive*** to input changes?
- Model is ***too insensitive*** to input changes?

Subsets of Training Data



compare models **"of a given model type"**

- FOR EACH subset (*whatever be it, for now*)
  - "Build model"… SEVERAL MODELS
- For example, several models of degree 1 polynomial
- For example, several models of degree 2 polynomial
- ….
- For example, several models of degree 1 polynomial **with** mean squared **error function**
- For example, several models of degree 1 polynomial **with** mean absolute **error function**

22) key phrase… “cross validation”

- The “average error” on various subsets of “training set”
- *(you will see more details soon!)*

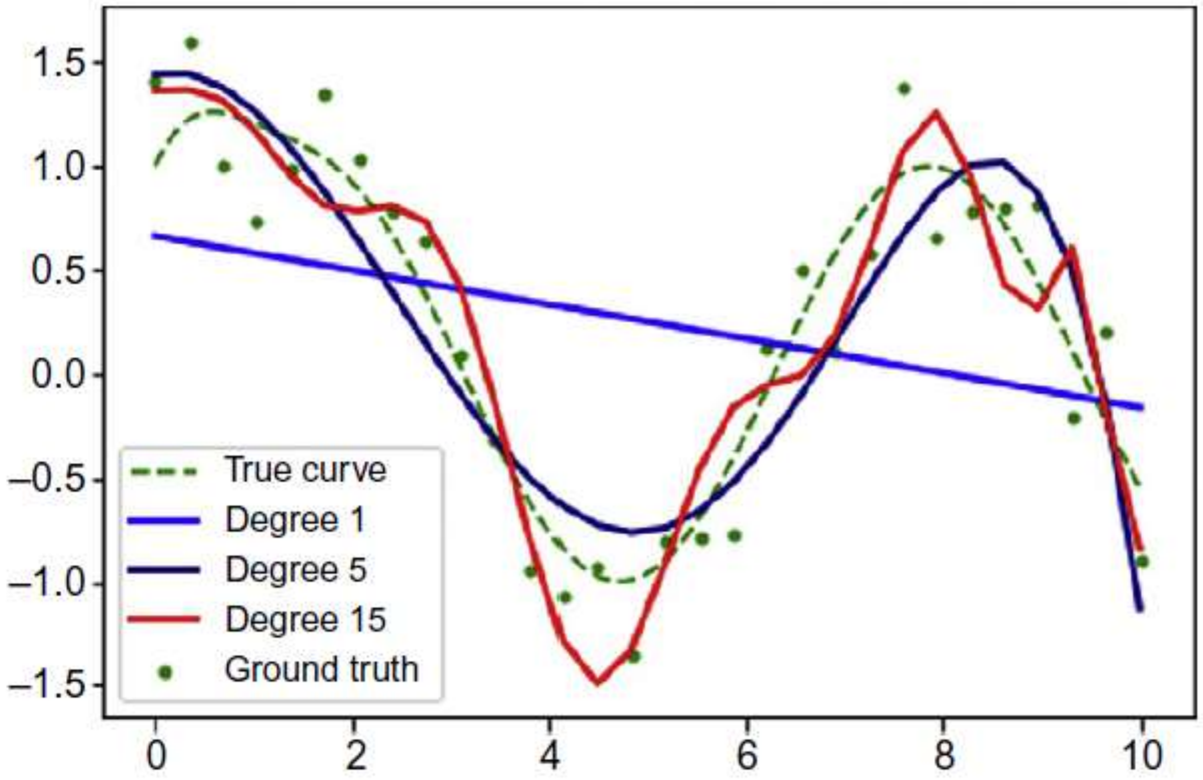


FIG. 4 Polynomial fitting—An underlying function $y = sin(x)+e^{-x^2}$ is used for generation of the data. The original function is shown in dashed ('- -') *green curve*. Random noise from uniform distribution $\xi \in U(-0.5, 0.5)$ is added to the curve. A data set $\{(x, y+\xi)\}$ is constructed and is shown as *green dots*. Polynomials using ridge regression with loss function, $L(w) = ||(y - X^T w)||^2 + ||w||^2$ are fitted with varying degrees 1, 5, and 15 and shown, respectively, in *blue*, *navy*, and *red* colors. The illustration aims to show overfitting nature of the higher degree polynomial, in *red*.

23) key phrases… 3 conceptual functions… split(), train(), test(), predict()

- Conceptual function split(D) – splits a given data set into train and test subsets.

  D_train, D_test = split(D)

- Conceptual function train(D) – Builds a model on a given data set, D

  M = train(D)

- Conceptual function test(M,D) – Computes error score of a given model on a given data set

  error_value = test(M,D)

- Conceptual function predict(M,x) – Computes predicted value for a given x by a given model M

  y’ = predict(M,x)

Algorithm for “cross validation”

- STEP 1 - Choose a model type (for example, degree 3 polynomial)
- STEP 2 – D_train, D_test = split(D)
- STEP 3 – Split training data into several sub-sets and split them
  - FOR i = 1 to k

    LS[i],VS[i] = split(D_train)

  //LS – Learning Set and VS – Validation Set
- STEP 4 – “model evaluation” on each subset
  - avg_error = 0
  - FOR i = 1 to k

    M = train(LS[i])

    avg_error += test(M,VS[i])
  - avg_error /= k
- STEP 5 – RETURN avg_error

Basic version…
there are other variations, we will discuss

Interpreting the “cross validation error”

- Low average error

- High average error

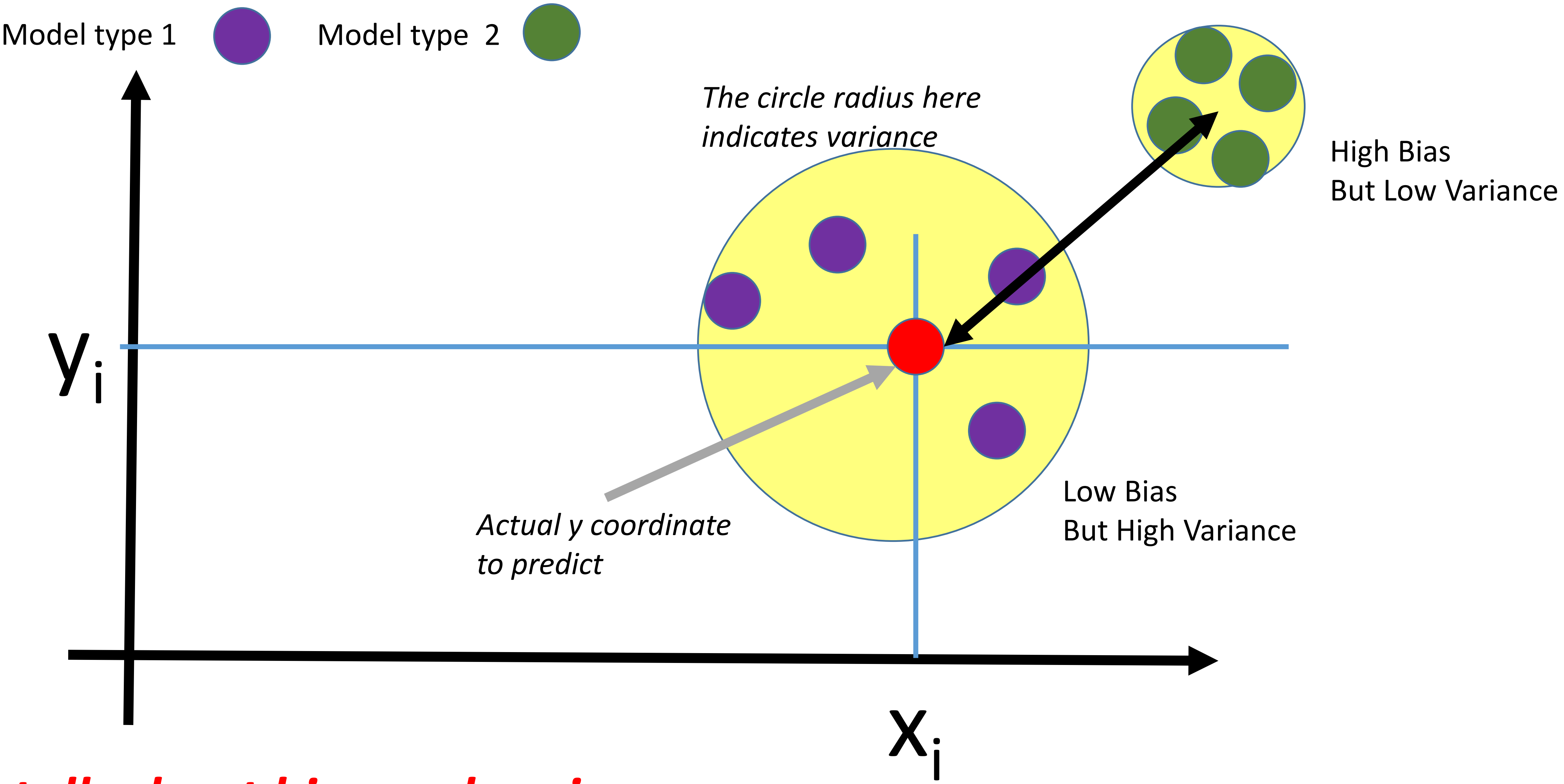
- Questions
  - *On average*, is it good on learning set and did poorly on validation set?
  - *On average*, is it good on both learning set and validation set?
  - *On average*, is it poor on both learning set and validation set?
  - “it” = “*the chosen model type*”

24) key phrase… “Bias”

- Conceptually it relates to “training set error”
- In “cross validation” setting, it relates to “learning set error”
- Inherently, ***if the model too simplistic?***

25) key phrase… “Variance”

- Conceptually it relates to “test set error”
- In “cross validation” setting, it relates to “validation set error”
- Inherently, ***if the model too complex?***



You talk about bias and variance

in the context of comparison of two or more models

26) key phrase… “Under fitting”

the model too simple

27) key phrase… “Over fitting”

the model too complex

“Bias Error” and “Variance Error”

- Conceptually... Cross validation error = Mix of Bias Error and Variance Error

- Using Mean Squared Error (MSE), we can show that…

cross validation MSE error = bias MSE error + variance MSE error

We have a theorem, we will discuss it later in the course!

*Fit several degree 0 polynomials (**horizontal** lines as below)*

*Fit several degree 1 polynomials (**inclined** lines as below)*

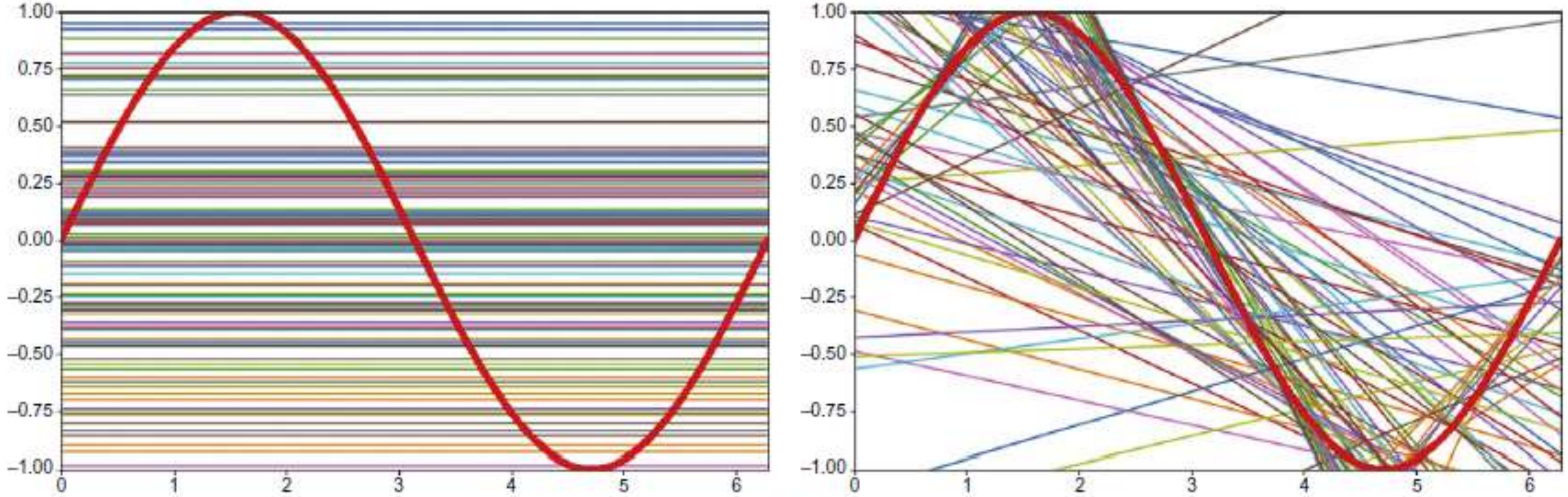


FIG. 8 The *red curve* in the *left* and the *right plots* is the true sine curve from which data is sampled uniformly to create (x, y) tuples. The problem is of regression type where given an x coordinate, the task is to predict the corresponding y coordinate. The *left plot* shows attempts to fit a degree 0 polynomial (i.e., $y = b$ type) and the *right plot* shows attempts to fit a degree 1 polynomial (i.e., of the form $y = a_0 * x + a_1$). Each of the *left* and the *right plots* have 100 models corresponding to the number of subsets sampled from the true sine curve and it corresponds to 100 lines we see here. In this plot, each of the 100 models has just 2 points in its training set. The task is to assess the behavior of the average model for its mean and variance.

Observed...
Higher bias at xi

FOR EACH model M:
y_avg += predict(M,xi)
y_avg = 1/M *y_avg

Observed...
Lower bias at xi

|y_avg – y_actual|

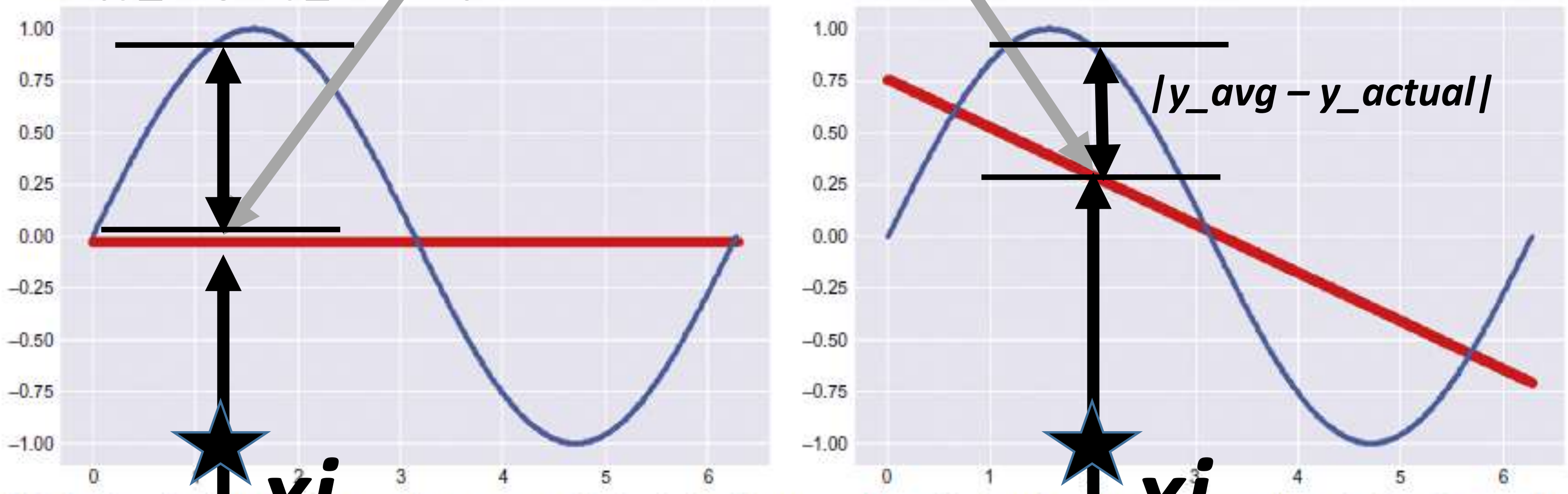


FIG. 9 For each col mn, e x-coordinate, average value is calculated from across the models along that c lum Th averaging mechanism, also termed as the average models fo the degree 0 polynomials (*left plot*) and the degree 1 polynomials (*right plot*) are show in *dark red color*.

Calculate average at each xi

Calculate average at each xi

Observed...
Lower variance at xi

FOR EACH model M:
val_list = val_list.append(predict(M,xi))
var_value = variance(val_lis)

Observed...
Higher variance at xi

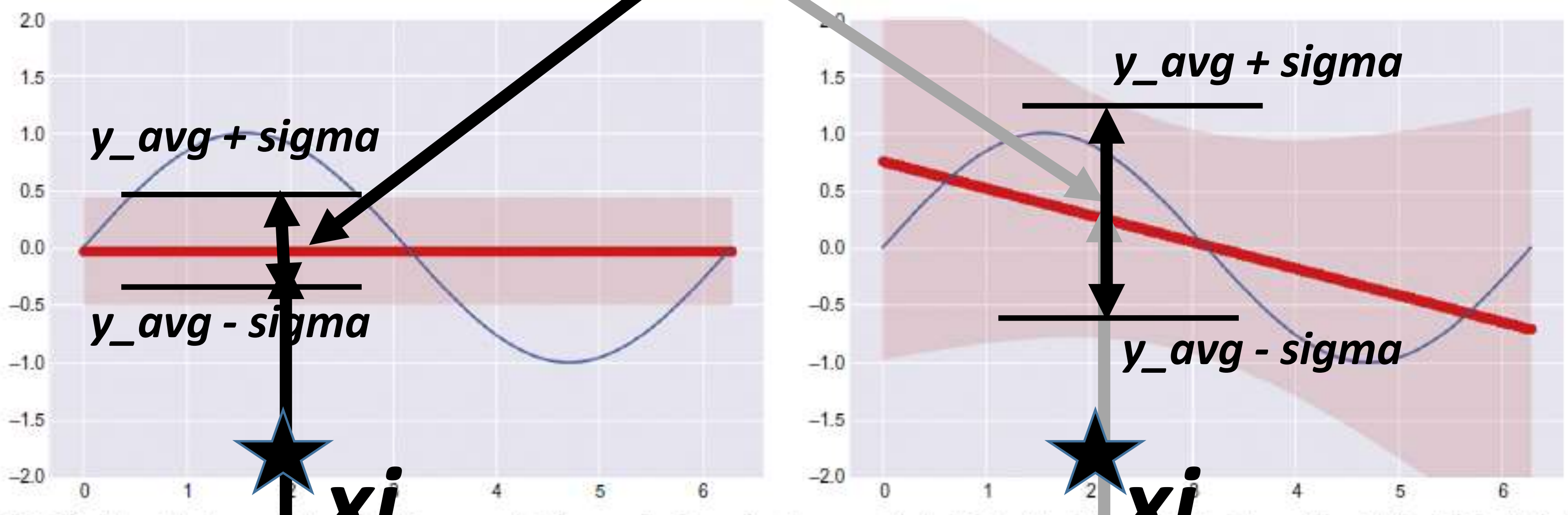


FIG. 10 The plots show variation about the mean $\mu(x)$ along each column, i.e., the x-coordinate. Each of the 100 models (in either of the plots) is build with just 2 points in its training set. Standard deviation $\sigma(x)$ is calculated for each x-coordinate value in a set of values computed by each of the models. A *pink colored line* is drawn between $\mu(x) - \sigma(x)$ to $\mu(x) + \sigma(x)$ to illustrate the amount of variation. One can note that, variation is higher in case of the degree 1 polynomial than the degree 0 polynomial. This is expected behavior because, *the space explored by the degree 1 polynomials is higher*, or in other words, *capacity is higher* for the degree 1 polynomial.

Calculate variance at each xi

Calculate variance at each xi

When **learning set size** increases → **Variance decreases**

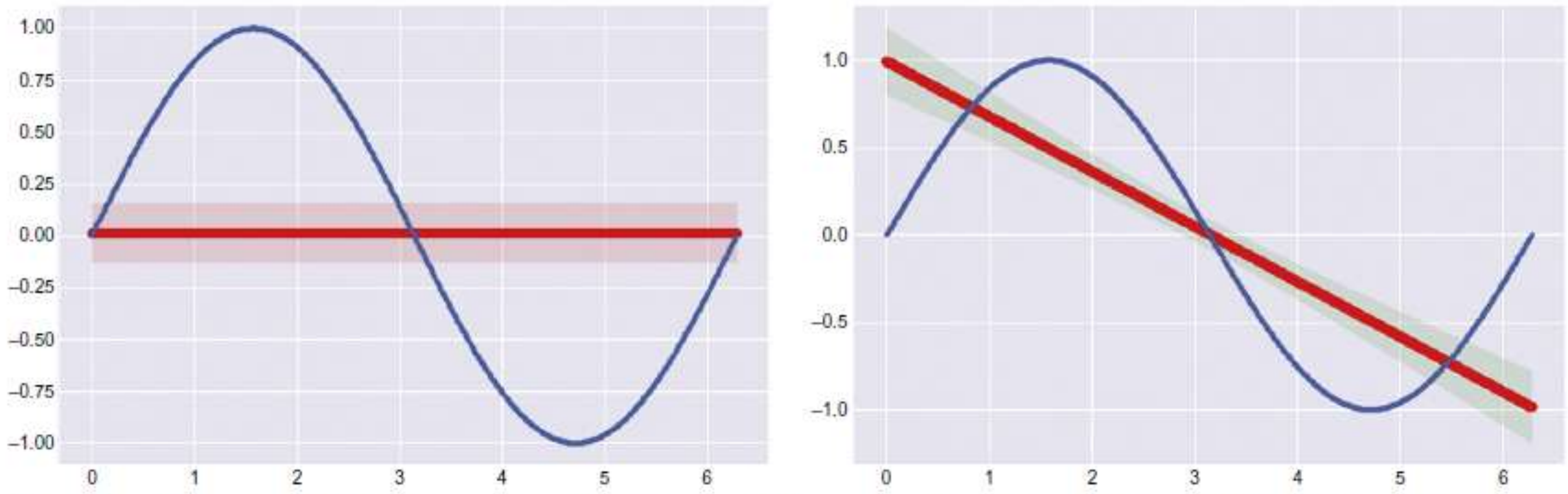


FIG. 11 The plots show the impact adding more training data points to each and every one of the models. Each of the 100 models (in either of the plots) is built with 100 points (*instead of just 2 as in Fig. 10*) points in its training set. As the x-coordinates are uniformly sampled from a given range, the more the size of the sample, the more the similarity between the sample sets. The reduction in variance is due to increasing of the training set sizes of the models that essentially emit similar individual models from each subset.

Diversity in data leads to more stable models

you will find a lot of philosophical statements and discussions of this form in internet

ALGORITHM 6 Bias variance calculation.

Require: X /*Input data set*/

  $\Gamma = []$ /*List of classifeirs*/

  for $i = 1 : M$ **do**

    $X_{sub} = subset(X)$

    $h^* = \arg\min_h L(h(X_{sub}), y)$

    $\Gamma \leftarrow \Gamma \odot h*$

  end for

  /*Define bias and variance as functions over individual classifiers' outputs*/

  Given $(x, y) \in X$ /*x is input and y is true value*/

  $bias(x) := (\mathbf{mean}(\{\gamma(x) : \forall \gamma \in \Gamma\}) - y)^2$

  $variance(x) := \mathbf{var}(\{\gamma(x) : \forall \gamma \in \Gamma\})$

28) key phrase… "Bias and Variance Trade Off"

- We need
  - Low Bias!
  - Low Variance!!
- Fact of life, ***not always possible! → Trade off***
- There are ways of achieving this – Methods of Ensembles
- *There is some 'regularization' to happen (we will discuss!)*

29) key phrase… “Model Selection”

Learning Curves

Caveat: These curves are only to assist in human intuition
We may not be able to quantitatively act on them!

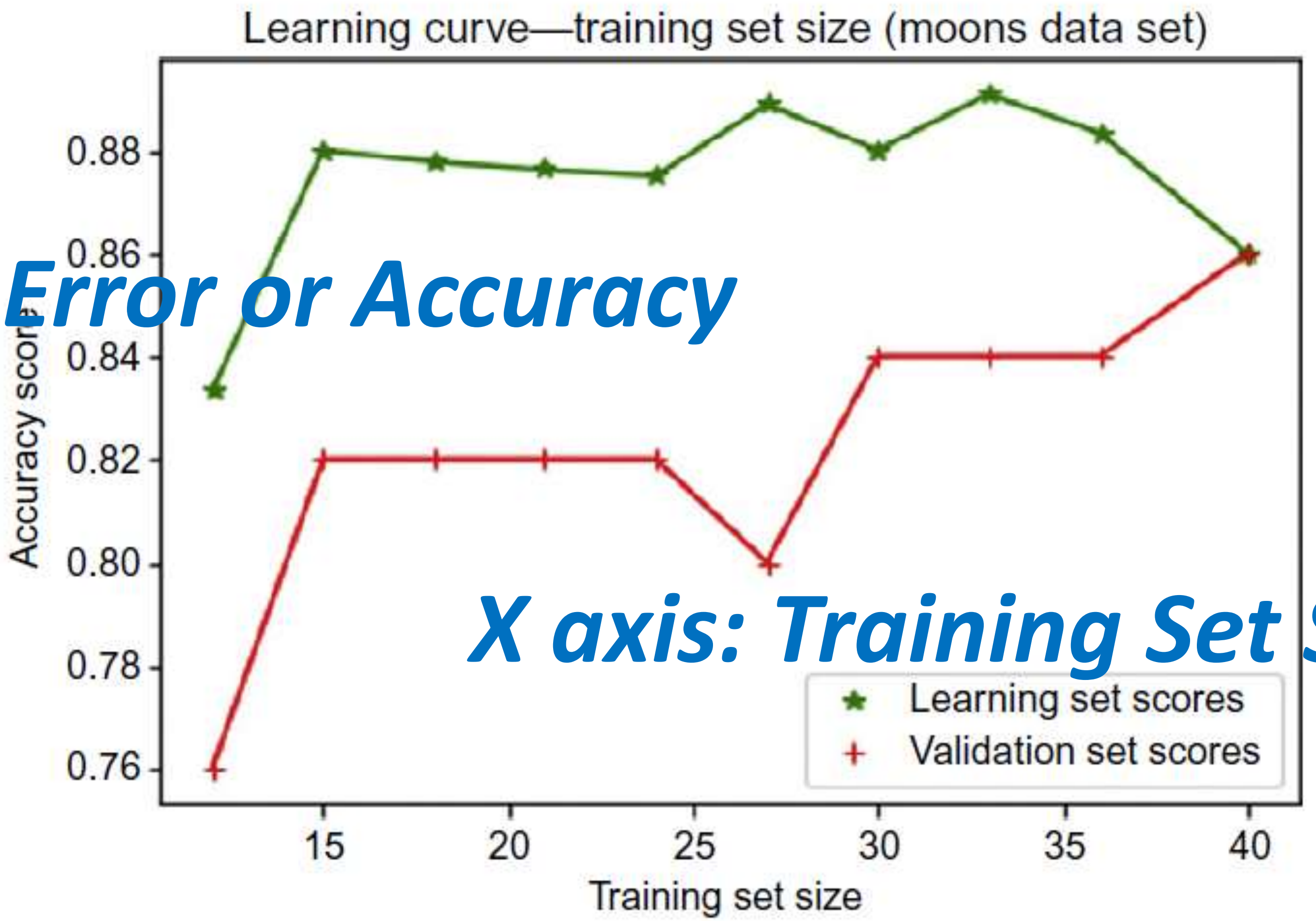


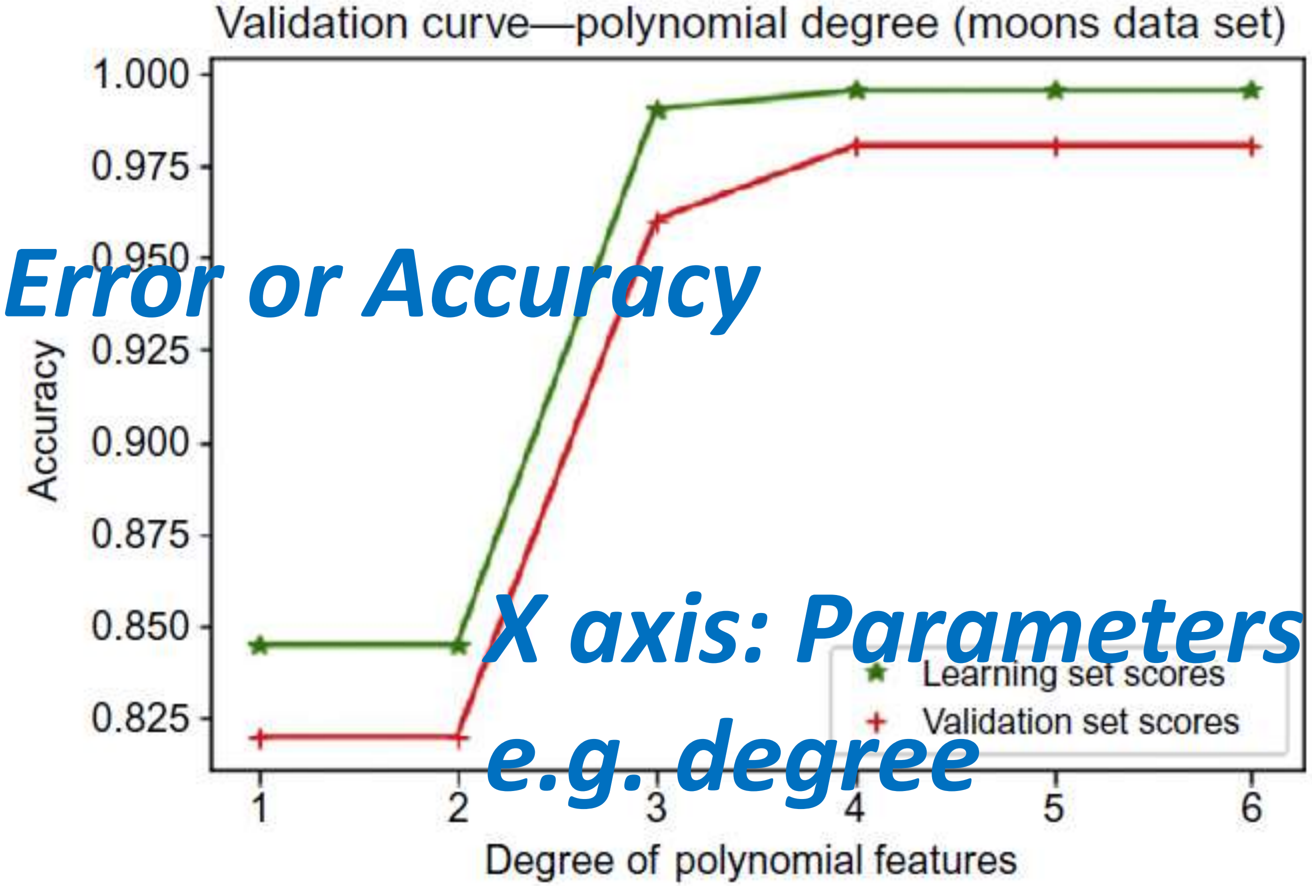
FIG. 12 Learning curve for various training set sizes is shown here for the moons data set and the classifier as used in Fig. 6. Cross validation (CV) with *stratified shuffling* for five random splits for *learning subset* and *validation subset* partitions in 80–20% sizes is carried out on each training data partition. Each execution of the CV results in a mean accuracy score of the learning and the vali-

We will discuss accuracy in classification methods

Validation Curves

Caveat: These curves are only to assist in human intuition
We may not be able to quantitatively act on them!

Y axis: Error or Accuracy

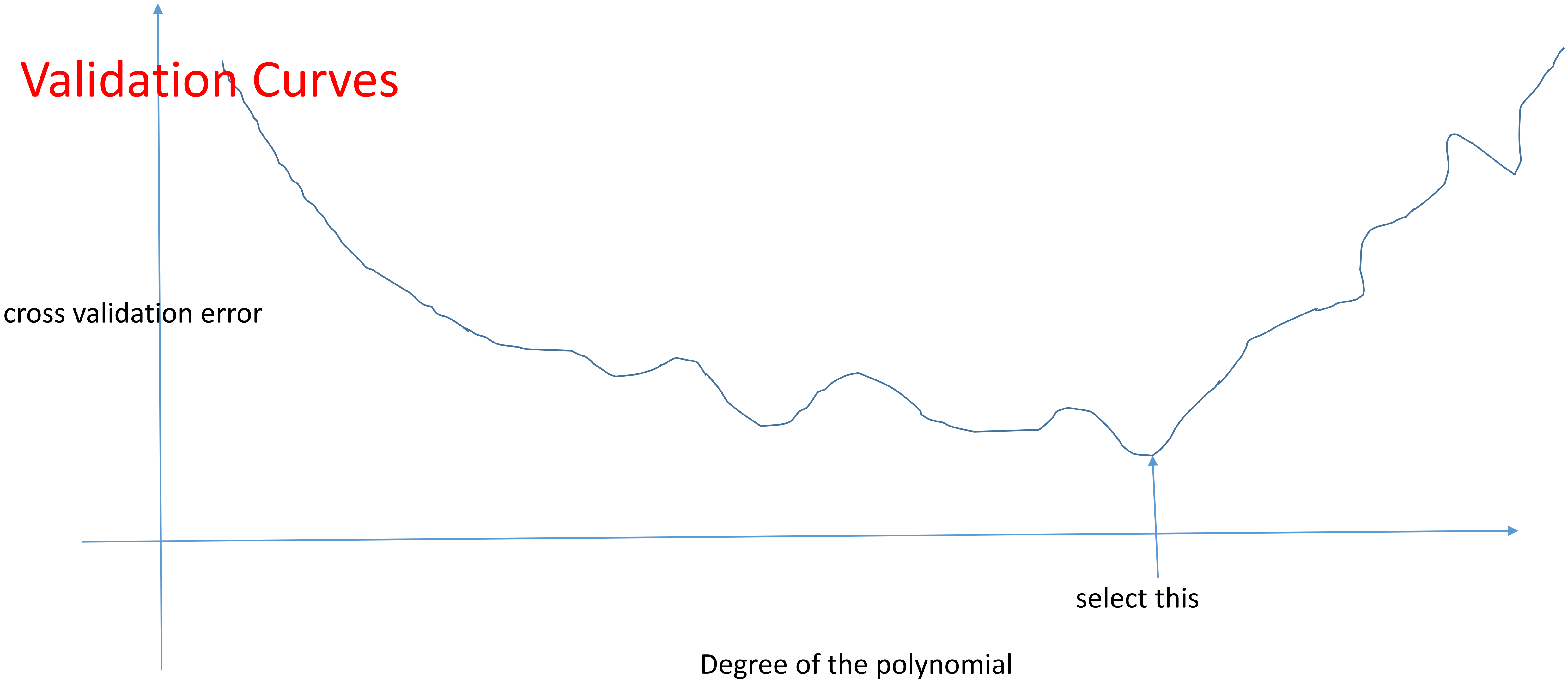


X axis: Parameters e.g. degree

FIG. 13 Validation curve for various degrees of polynomial features expanding the input dimensionality is shown here for the moons data set and the classifier as used in Fig. 6. The data that is expanded for feature dimensionality is of the form $(x_1, x_2, label)$, which is expanded into a higher dimension k as the set of all distinct terms of the form $\{x_1^a \times x_2^b | \forall(a,b) : 0 \leq (a+b) \leq k\}$.

We will discuss accuracy in classification methods

Validation Curves



30) key phrase… "Model Selection"

Model Selection???

Simple…!
Select that model which gives…
least average cross validation error