SciML - Machine learning

Mark Asch - IMU/VLP/CSU

2023

Program

Recall machine learning (ML) techniques for scientific applications (see Basic Course for details):

1. General principles of ML.
   (a) Under- and Over-fitting
   (b) Bias-Variance Tradeoff
   (c) Recall-Precision Tradeoff
   (d) Cross Validation

2. Supervised learning:
   (a) Regression.
   (b) Classification.

3. Unsupervised learning:
   (a) Clustering.
   (b) Trees.

4. Surrogate models and dimensionality reduction.

5. The use of machine learning in scientific applications.

6. The challenges of applying machine learning to scientific applications.

A simple example that brings out many points

- Basic fact: most of ML is about pattern recognition—see C. M. Bishop, PRML, Springer, 2006.

  $\Rightarrow$ we have a bunch of data, $\mathbf{x}$ and corresponding observations, $\mathbf{y}$

  $\Rightarrow$ we seek the relation (recognize the pattern), f, that relates $\mathbf{x}$ to $\mathbf{y}$

  $$\mathbf{y} = f(\mathbf{x}),$$

  or,

  $$f : \mathbf{x} \longmapsto \mathbf{y}$$

  $\Rightarrow$ our goal: make predictions for new (unseen) data (inputs)

Polynomial Curve Fitting

- suppose that

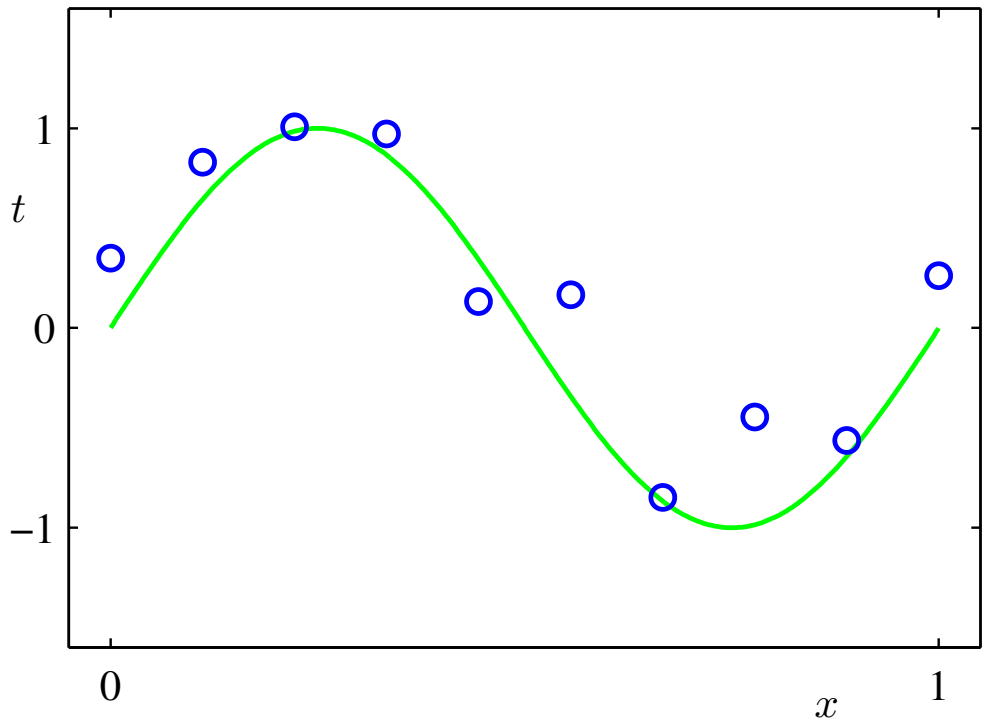
$$y = \sin(2\pi x) + \text{noise}$$

- given: a training set of N observations of x,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

and the corresponding (noisy) observations,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$



- data generation from

$$y_i = \sin(2\pi x_i) + \epsilon,$$

where the noise term,

$$\epsilon \sim \mathcal{N}(0, \sigma)$$

is due to intrinsic/natural variability and other uncertainties from the data collection methods.

- **Objective:** from the training set, learn/discover an approximation of the relation f that will enable us to make predictions based on

  $\Rightarrow$ probability theory
  $\Rightarrow$ (polynomial) curve fitting

- PCF: suppose that the data can be represented by the polynomial/"linear model"

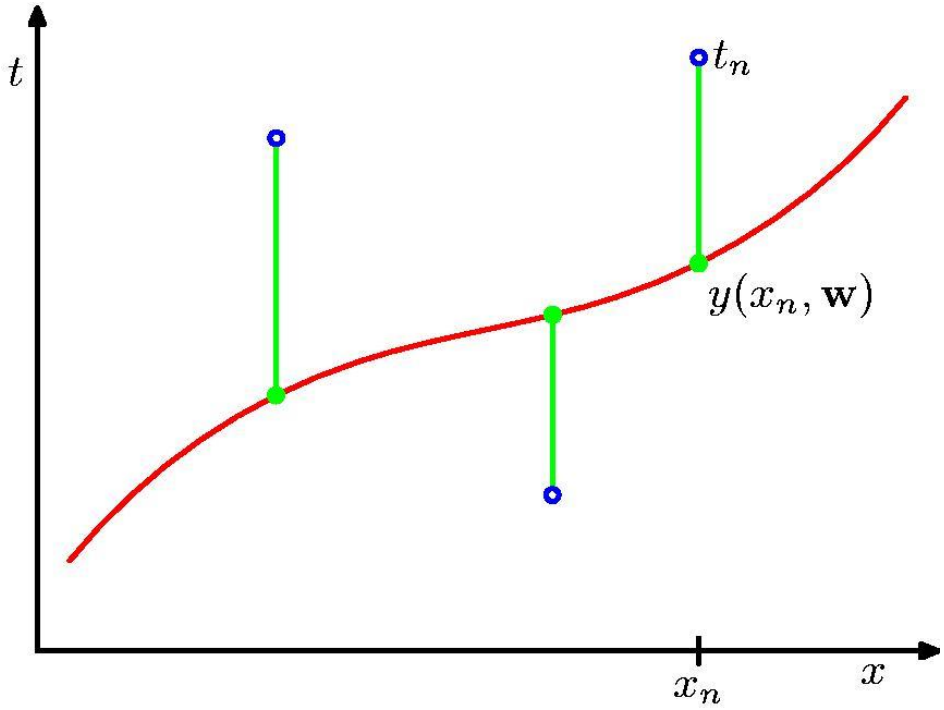
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{k=0}^{M} w_k x^k$$

  with weight vector (coefficients) $\mathbf{w}$.

  $\Rightarrow$ compute the coefficents by minimizing an error/loss function—this is (often) a least-squares approach, where the loss function is defined as

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \left[\hat{y}(x_i, \mathbf{w}) - y_i \right],$$

  where $\hat{y}_i$ is the approximation obtained from our polynomial model.



- in the case of a quadratic loss function, the gradient is linear and the solution to the optimization problem exists and can be found in closed form using the normal form, or psuedo-inverse,

$$\mathbf{w}_* = \left(X^{\mathrm{T}} X\right)^{-1} X^{\mathrm{T}} \mathbf{y},$$

where X is the (rectangular) coefficient matrix of dimension $(N \times M)$ with $N > M$ and $\mathbf{y}$ is the data/observation vector.

- the choice of M is a model selection problem

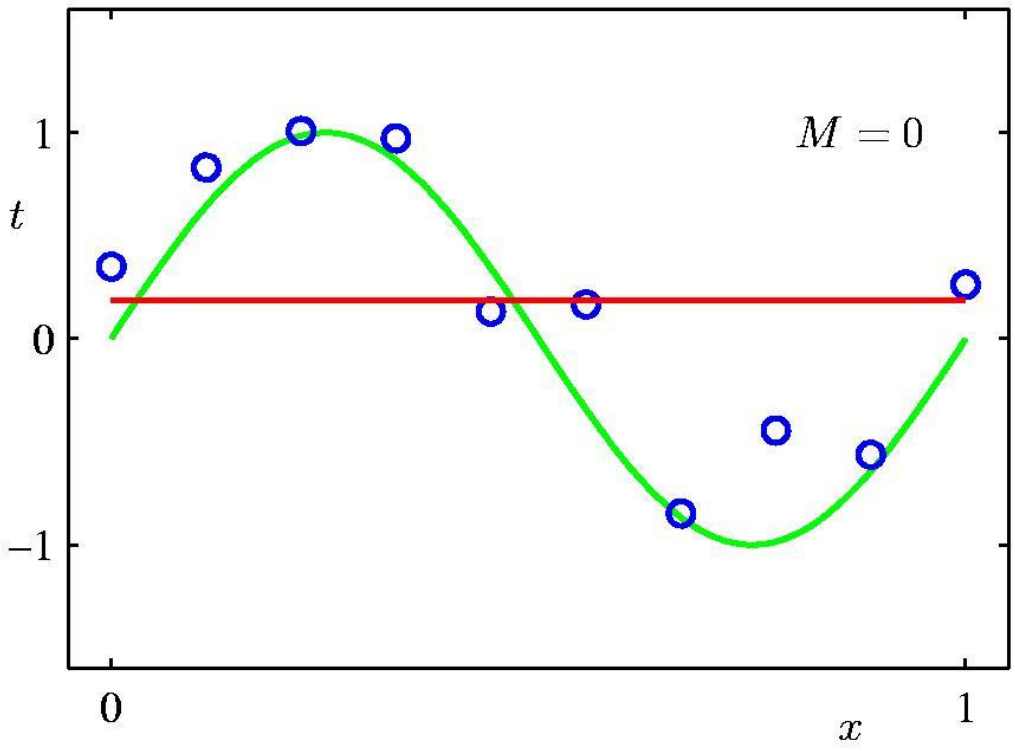
  $\Rightarrow$ we compute the solution for 4 values: $M = 0, 1, 3, 9$—see below

  $\Rightarrow$ Model $M = 0$ produces underfitting—it is too simple and does not find the trend

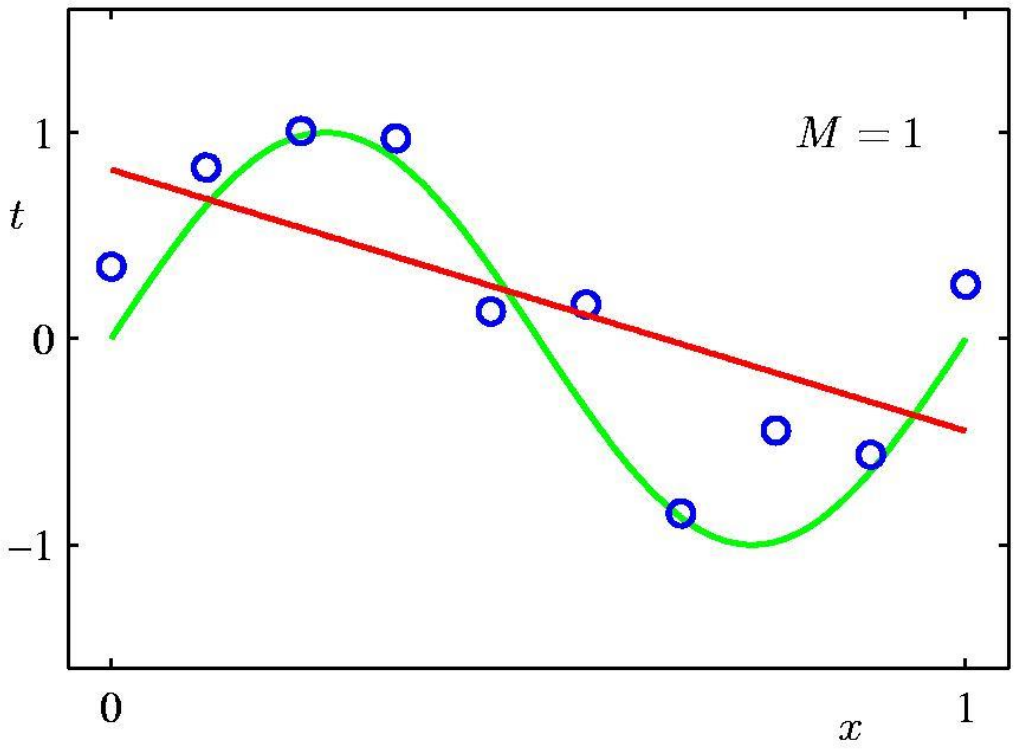
  $\Rightarrow$ Model $M = P$ produces overfitting—it fits the data points perfectly, but is "brittle" and does not generalize

- **Conclusion:** we must seek a compromise between the two.

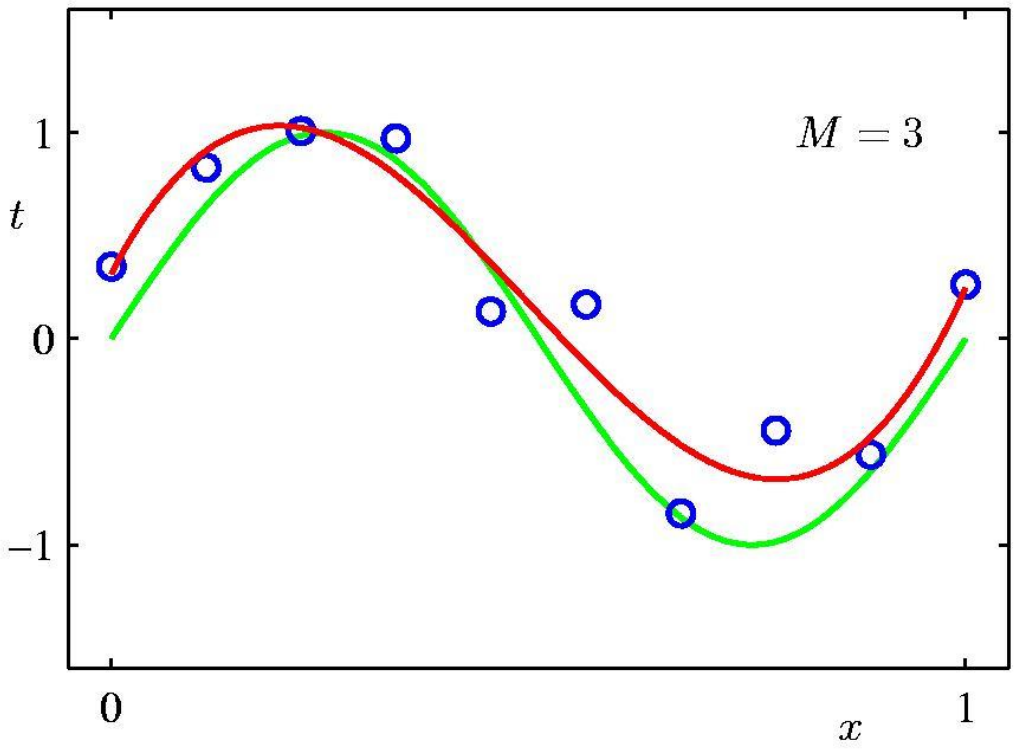
0^{th} Order Polynomial



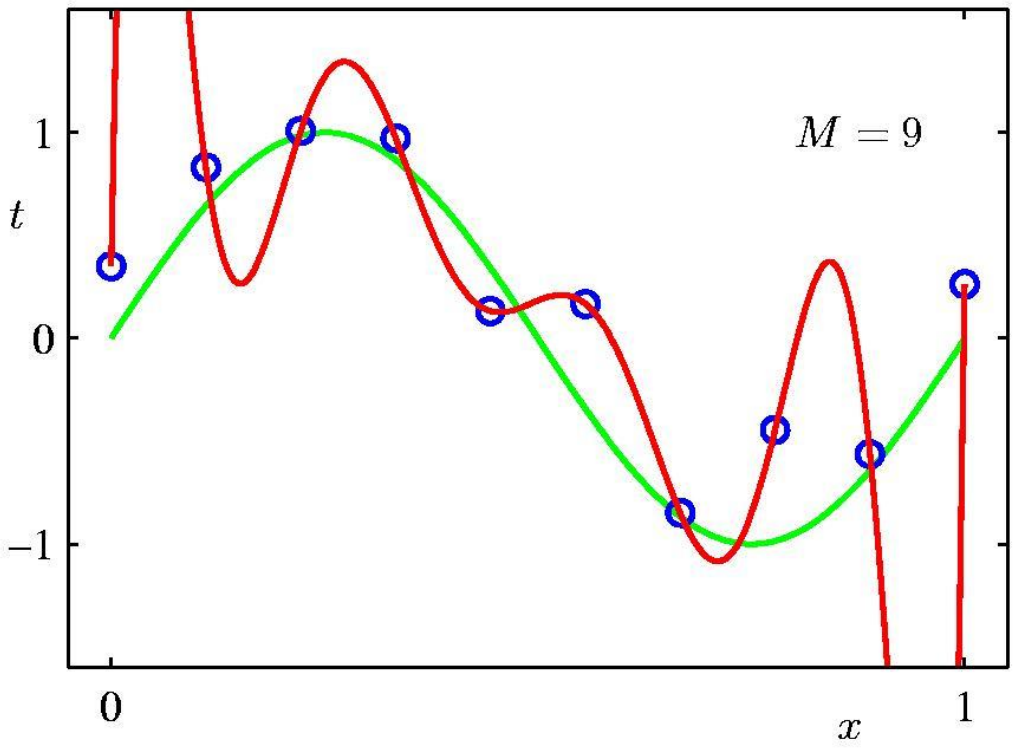
1st Order Polynomial



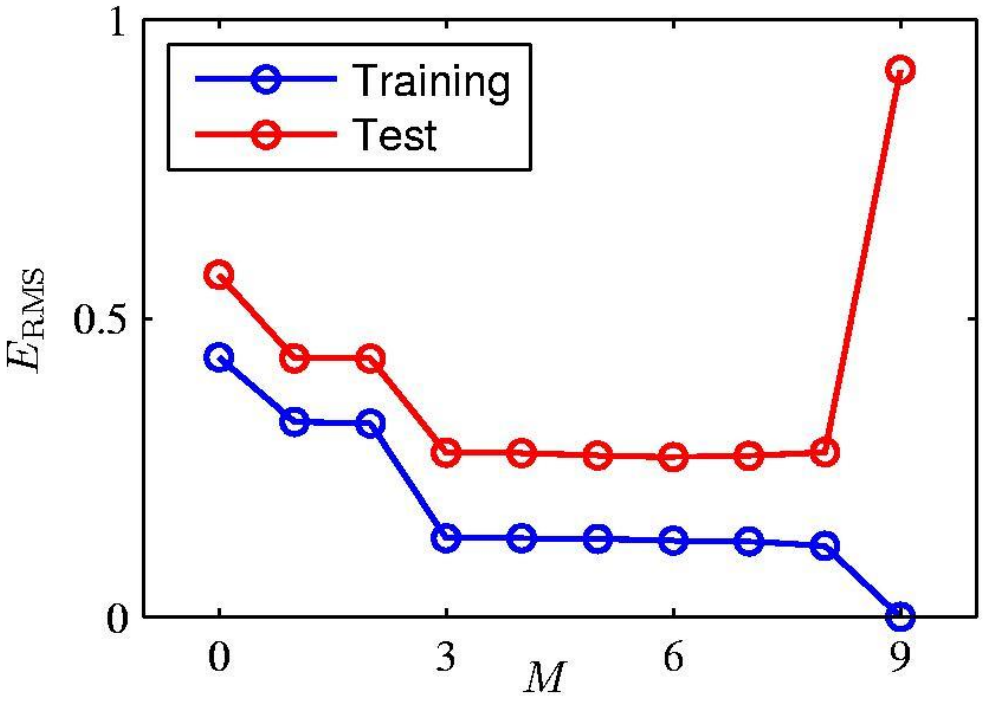
3rd Order Polynomial



9th Order Polynomial



Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

Polynomial Coefficients

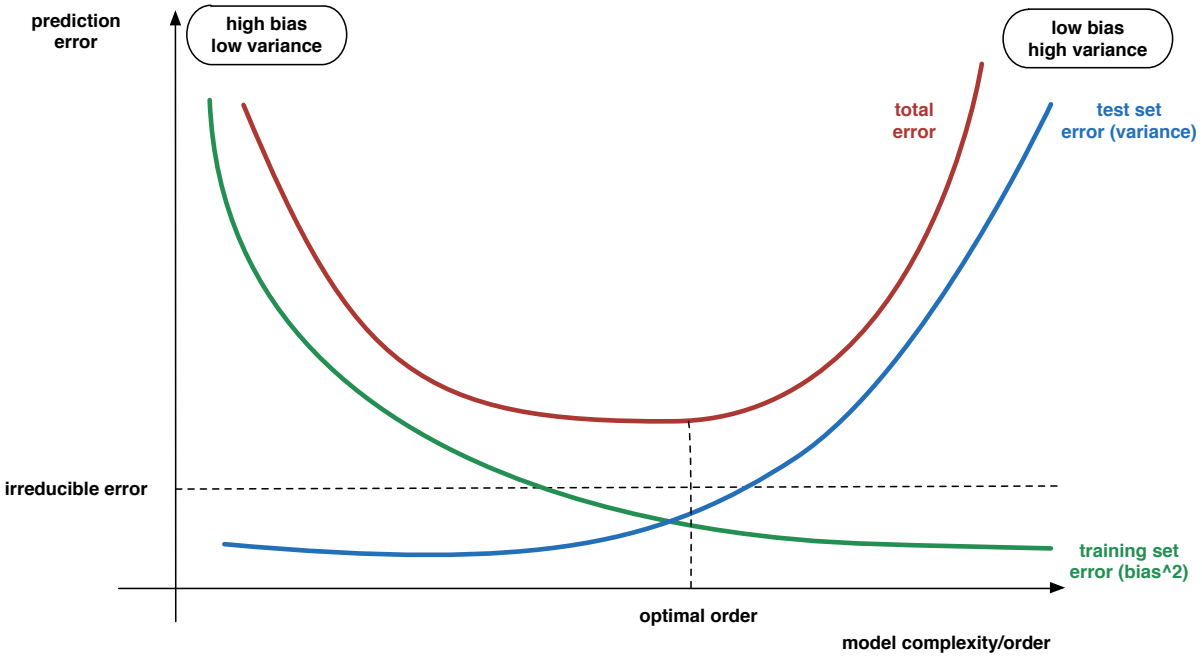
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Bias-Variance Compromise

- We saw the dangers of overfitting in the very simple polynomial fitting example.

- To treat this, we need to make a compromise between bias and variance, i.e. forgo some precision in order to attain better predictive performance.

  $\Rightarrow$ this is a fundamental point underlying **ALL** machine learning methods



Recall: Mean-Squared Error

- Since we are estimating an unknown parameter, or more often, unknown function, we need to evaluate

  $\Rightarrow$ how good our estimation is?
  $\Rightarrow$ the quality of the estimation?

- This is traditionally done by using a discrete L_2 norm, though other norms are possible and are used in specific circumstances—see below.

- Before defining the MSE, we need to recall the definitions of

  $\Rightarrow$ expectation (or expected value): $\mathrm{E}[X] = \sum_i x_i f(x_i)$, where f is the (discrete) probability function.
  $\Rightarrow$ bias: $\mathrm{Bias}(T, \theta) = \mathrm{Bias}(T) = \mathrm{E}[T] - \theta$, where T is a statistic used to estimate a parameter, and θ is the parameter to be estimated

$\Rightarrow$ variance: $\mathrm{Var}\left[X\right] = \mathrm{E}\left[(X-\mu)^2\right] = \mathrm{E}\left[X^2\right] - \mu^2$, where μ is the mean of X.

Definition 1 (Mean Squared Error). The mean squared error measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(e_i\right)^2,$$

or for an estimator

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{E}_\theta\left[\left(\hat{\theta} - \theta\right)^2\right].$$

The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator (how widely spread the estimates are from one data sample to another) and its bias (how far off the average estimated value is from the true value).

The MSE either assesses the quality of a predictor (i.e., a function mapping arbitrary inputs to a sample of values of some random variable), or of an estimator (i.e., a mathematical function mapping a sample of data to an estimate of a parameter of the population from which the data is sampled). The definition of an MSE differs according to whether one is describing a predictor or an estimator.

Remark 1. The mean squared error has the disadvantage of heavily weighting outliers. This is a result of the squaring of each term, which effectively weights large errors more heavily than small ones. This property, undesirable in many applications, has led to the use alternatives such as the mean absolute error, which is a discrete L_1 norm, or those based on the median.

Definition 2 (Mean Absolute Error)**.** The mean absolute error is the average of the sum of absolute errors,

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i| \,,$$

or for an estimator

- A python snippet:

```
from sklearn.metrics import mean_absolute_error
import numpy as np
# Generate some sample data
y_true = np.array([1, 2, 3, 4, 5])
y_pred = np.array([1.5, 2.5, 2.8, 4.2, 4.9])
# Calculate the MAE
mae = mean_absolute_error(y_true, y_pred)
print("Mean Absolute Error", mae)
```

Remark. The MAE has two important properties. It is robust to outliers, and it has the same units as the response variable.

Bias-Variance Decomposition of MSE

Theorem 1. *The bias-variance decomposition of the MSE is given by*

$$\begin{aligned}\mathrm{MSE} &\doteq \mathrm{E}\left[\left(y - \hat{f}\right)^2\right] \\ &= \mathrm{Bias}\left[\hat{f}\right]^2 + \mathrm{Var}\left[\hat{f}\right] + \sigma^2,\end{aligned}$$

where $y = f(x) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ *and* $\hat{f}(x)$ *is an approximation/model of the true (unknown) function* $f(x)$.

Proof. First expand the definition of MSE,

$$\begin{aligned}\mathrm{MSE} &\doteq \mathrm{E}\left[\left(y - \hat{f}\right)^2\right] \\ &= \mathrm{E}\left[y^2 - 2y\hat{f} + \hat{f}^2\right] \\ &= \mathrm{E}\left[y^2\right] - 2\mathrm{E}\left[y\hat{f}\right] + \mathrm{E}\left[\hat{f}^2\right].\end{aligned}$$

Now look at each of the three terms.

$$\mathrm{E}\left[\hat{f}^2\right] = \mathrm{Var}\left[\hat{f}\right] + \mathrm{E}\left[\hat{f}\right]^2$$

from variance decomposition.

$$\begin{aligned}
\mathrm{E}\left[y^2\right] &= \mathrm{E}\left[(f+\epsilon)^2\right] \\
&= \mathrm{E}\left[f^2\right] + 2\mathrm{E}\left[f\epsilon\right] + \mathrm{E}\left[\epsilon^2\right] \\
&= f^2 + 2f \cdot 0 + \sigma^2
\end{aligned}$$

from definition of the noise distribution ϵ.

$$\begin{aligned}
\mathrm{E}\left[y\hat{f}\right] &= \mathrm{E}\left[(f+\epsilon)\,\hat{f}\right] \\
&= \mathrm{E}\left[f\hat{f}\right] + \mathrm{E}\left[\epsilon\right]\mathrm{E}\left[\hat{f}\right] \\
&= f\mathrm{E}\left[\hat{f}\right].
\end{aligned}$$

Putting it all together, we find

$$\begin{aligned}\mathrm{MSE} &= \left(f^2 + \sigma^2\right) - 2f\mathrm{E}\left[\hat{f}\right] + \left(\mathrm{Var}\left[\hat{f}\right] + \mathrm{E}\left[\hat{f}\right]^2\right) \\ &= \left(f - \mathrm{E}\left[\hat{f}\right]\right)^2 + \sigma^2 + \mathrm{Var}\left[\hat{f}\right] \\ &= \mathrm{Bias}\left[\hat{f}\right]^2 + \mathrm{Var}\left[\hat{f}\right] + \sigma^2.\end{aligned}$$

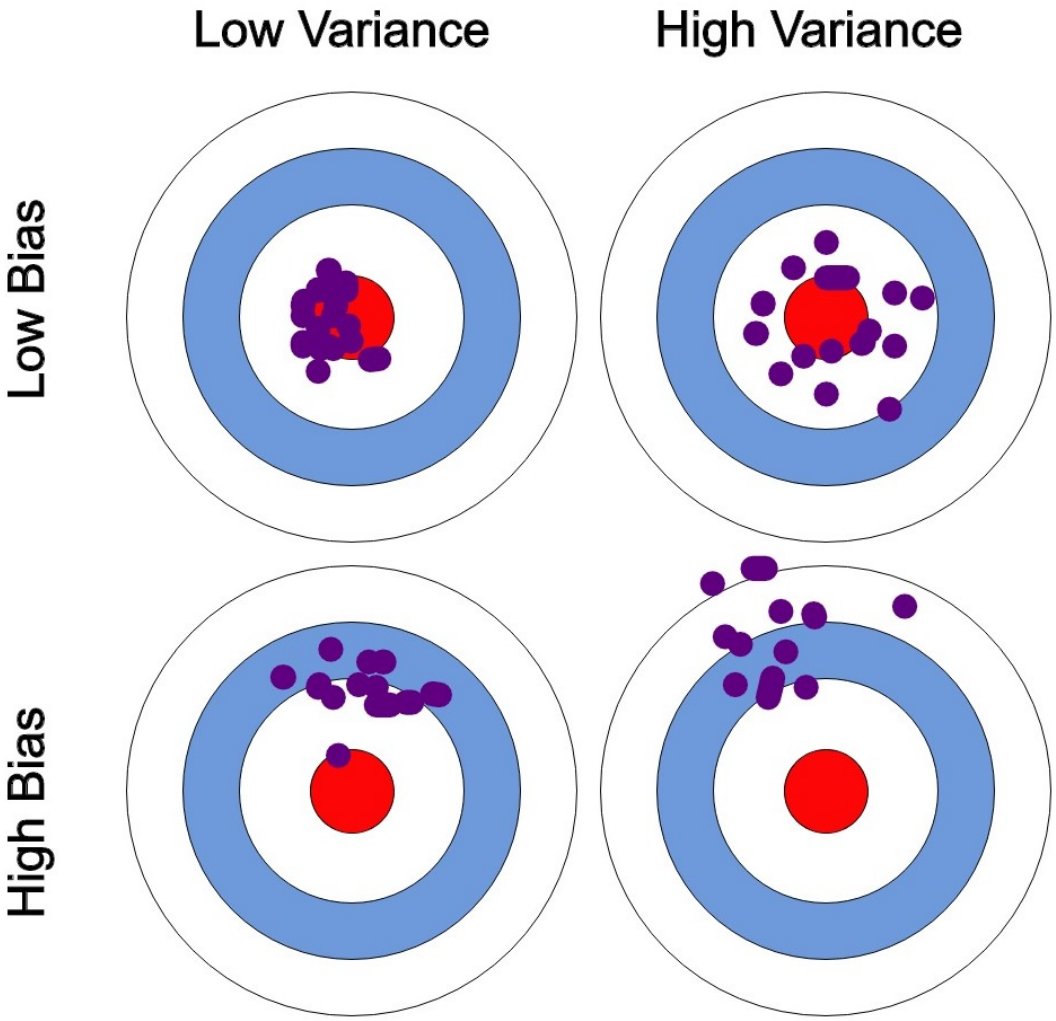
□

Bias-Variance Compromise II

- Questions:

  ⇒ What is bias? (see also Ethics lecture)
  - → Low Bias: Predicted data points are close to the target. The model has fewer assumptions about the form of the target function.
  - → High-Bias: Predicted data points are far from the target. The model has more assumptions about the form of the target function.

  ⇒ What is variance?
  - → Low Variance: Data points are close to each other, and as a result are close to the function. The model suggests small changes to the estimate of the target function with changes in the training dataset.
  - → High Variance: Data points are spread out and as a result are far from the function. The model suggests large changes to the estimate of the target function with changes in the training dataset.


Low Variance
High Variance
Low Bias
High Bias

Bias-Variance Compromise III

- We can now return to the question of the tradeoff between bias and variance.

- Looking at the bias-variance decomposition of the MSE,

$$\mathrm{MSE} = \mathrm{Bias}\left[\hat{f}\right]^2 + \mathrm{Var}\left[\hat{f}\right] + \sigma^2,$$

  we observe that we need to simultaneously minimize both bias and variance. This is the reducible part of the error, whereas σ^2 is the intrinsic, irreducible part and cannot be reduced by the ML model.

- Theoretical result (see also the "compromise curve" above):

  $\Rightarrow$ Decreasing the bias will mechanically increase the variance.

  $\rightarrow$ Low bias $\Longrightarrow$ high variance.

$\Rightarrow$ Decreasing the variance will mechanically increase the bias.

$\rightarrow$ Low variance $\Longrightarrow$ high bias.

- **Conclusion:**

$\Rightarrow$ To build a good model, we need to find a good balance/compromise/tradeoff between bias and variance that minimizes the total/generalization error.

$\Rightarrow$ An optimal balance of bias and variance should neither overfit nor underfit the model. This is achieved by hyperparameter tuning on the basis of systematic grid-search (trial and error, or experience, for the ranges) and cross-validation. See below, and examples.

Over- and Underfitting

- ML Algorithm properties:
  - ⇒ Examples of low-bias machine learning algorithms: Decision Trees, k-Nearest Neighbors and Support Vector Machines.
  - ⇒ Examples of high-bias machine learning algorithms: Linear Regression, Linear Discriminant Analysis and Logistic Regression.
  - ⇒ Examples of low-variance machine learning algorithms: Linear Regression, Linear Discriminant Analysis and Logistic Regression.
  - ⇒ Examples of high-variance machine learning algorithms: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

- Dealing with the overfitting problem:
  - ⇒ always be prepared to "sacrifice" some bias, to get a lower variance, since it is the variance that is the most "dangerous" in terms of predictive power/performance.

⇒ in general, prefer lower-order ML methods—so-called linear methods

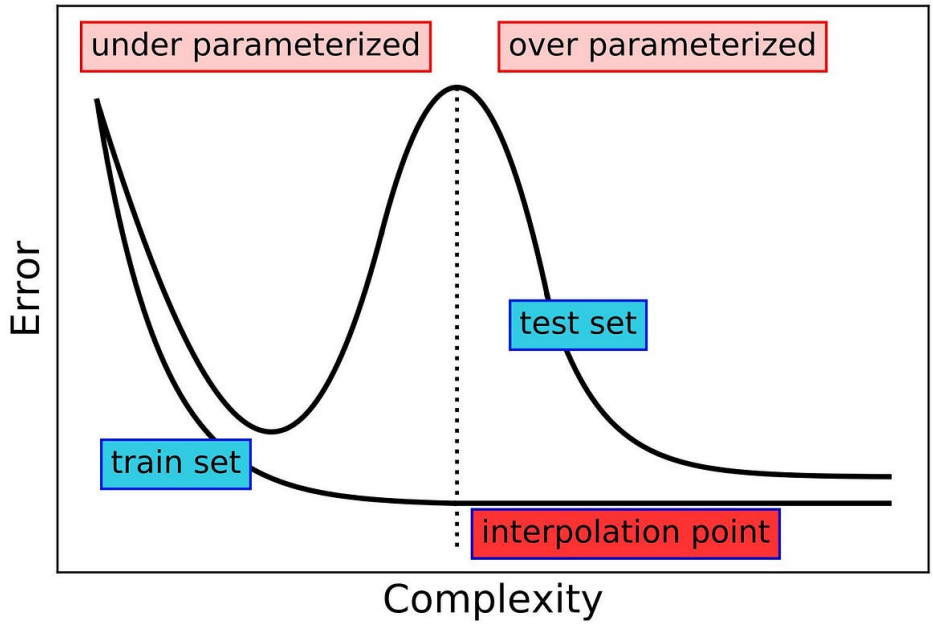
⇒ increase the number of data points—this is not always feasible in reality

⇒ use a Bayesian approach—probably the best, but more complex to implement

⇒ regularize the problem—penalize terms of large magnitude, but requires extrensive tuning of the regularization parameters that usually relies on cross-validation techniques.

BV Tradeoff: Double Descent phenomenon

- Test error can present a "double descent" phenomenon in a range of ML models.
- As the number of model parameters grows relative to the number of data points, test error drops in the highly overparameterized (data undersampled) regime



- Interpretation: [OpenAI]

  $\Rightarrow$ For models at the interpolation threshold, there is effectively only one model that fits the train data, and forcing it to fit even slightly noisy or misspecified labels will destroy its global structure.

  $\Rightarrow$ There are no "good models" which both interpolate the train set and perform well on the test set.

  $\Rightarrow$ However, in the over-parameterized regime, there are many models that fit the train set and there exist such good models.

  $\Rightarrow$ Moreover, the implicit bias of stochastic gradient descent (SGD) leads it to such good models, for reasons we do not yet understand...

Precision-Recall Compromise

- What happens with binary classification problems?

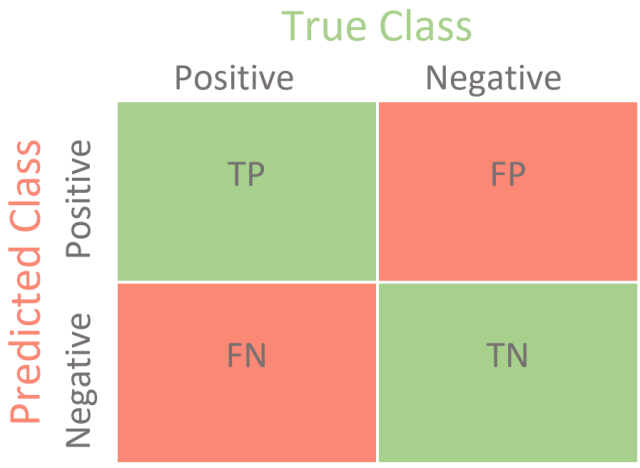
  $\Rightarrow$ the famous "truth/confusion table":

  $\rightarrow$ True positives (TP): positive COVID test, correctly diagnosed.

  $\rightarrow$ False positives (FP): positive COVID test, falsely diagnosed.

  $\rightarrow$ True negatives (TN): negative COVID test, correctly diagnosed.

  $\rightarrow$ False negatives (FN): negative COVID test, falsely diagnosed.



Definition 3. A *confusion table/matrix* , C, for a classification with n classes is an $(n \times n)$ matrix with entries

$$C_{ij} = \text{the number of observations in class } i \text{ that are predicted in class } j.$$

Then,

- C_{ii}, $i = 1, \ldots, n$, are the good classifications,
- C_{ij} with $i \neq j$ are the bad classifications.
- Now we can define precision and recall/sensitivity.

  Precision the number of true positives divided by the total number of true positives and false positives:

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 1 - \text{FDR}$$

  Appropriate when minimizing false positives is the focus—FP is more serious, think of a false postive cancer diagnosis, and the heavy, unecessary treatment that follows.

Recall the number of correct positive predictions made out of all positive predictions that could have been made:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

Appropriate when minimizing false negatives is the focus—FN is more serious, think of missing a positive COVID, that then goes on to infect many people.

Specificity is the number of true negatives divided by the total number of true negatives and false positives:

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{FPR}$$

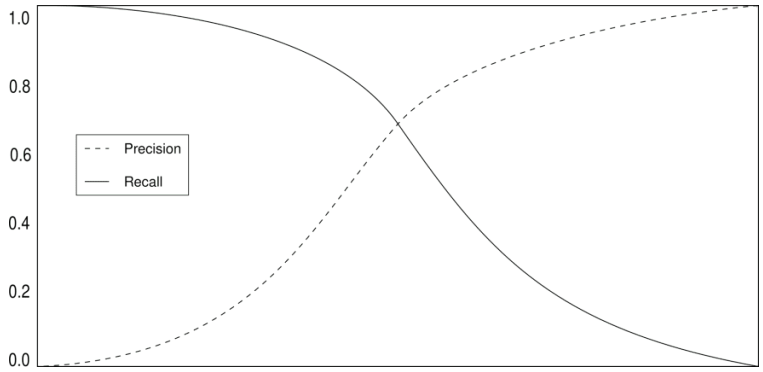
F1-Score is the harmonic mean of precision and recall:

$$\mathrm{F}_1 = 2 \times \frac{\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Alone, neither precision or recall tells the whole story. We can have excellent precision with terrible

recall, or alternately, terrible precision with excellent recall. F-measure provides a way to express both concerns with a single score.

Precision-Recall Compromise II



- The compromise:
  - ⇒ When precision increases, recall decreases, and vice-versa.
  - ⇒ We cannot simultaneously increase both precision and recall.
- To balance precision and recall, a number of techniques can be used, such as
  - ⇒ adjusting the decision threshold or
  - ⇒ using an ensemble of models.
- Another approach is to use a metric that combines both precision and recall,

$\Rightarrow$ such as the F1 score, which is the harmonic mean of precision and recall.

- Ultimately, the choice between precision and recall depends on the specific goals/context of the application.

  $\Rightarrow$ In some cases, such as detecting fraudulent transactions, precision may be more important than recall, as false positives can have significant consequences.

  $\Rightarrow$ In other cases, such as detecting rare medical conditions, recall may be more important, as missing a true positive can have serious consequences.

Compromises: Conclusions

1. There is always the need to compromise, or tradeoff:
   (a) Bias-Variance
   (b) Precison-Recall

2. The compromise we choose will always be subjective, and require judgement in context. There is no single, well-defined, "right" answer.

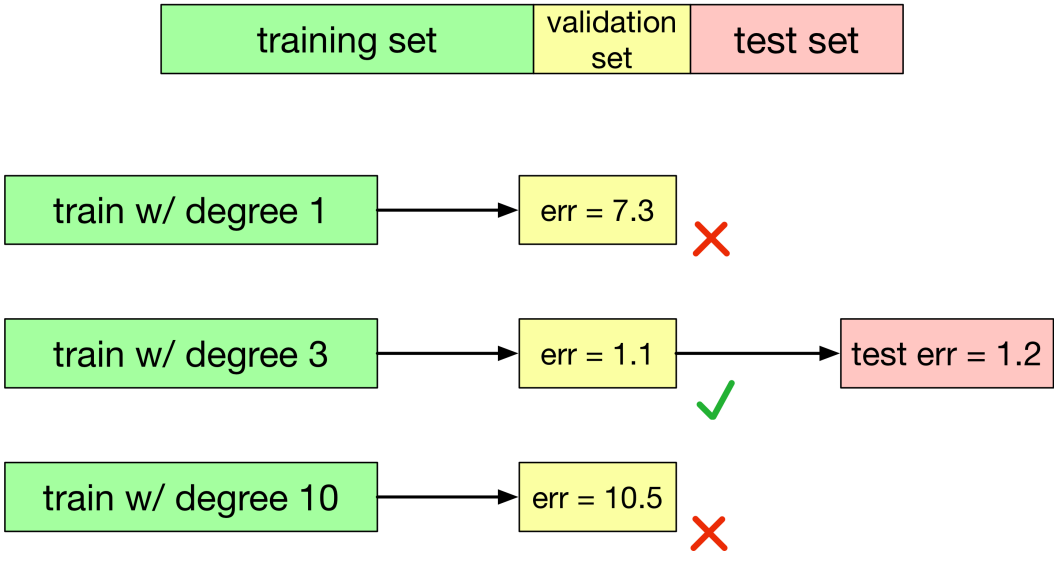
3. This will necessarily introduce an ethical bias. See Ethics lectures.

Cross-Validation and Tuning

Please REVIEW/SEE Basic Course... [00_resample.pdf]

- in the previous example of curve fitting, the degree of the polynomial is an example of a hyperparameter that cannot be included in the training/learning procedure

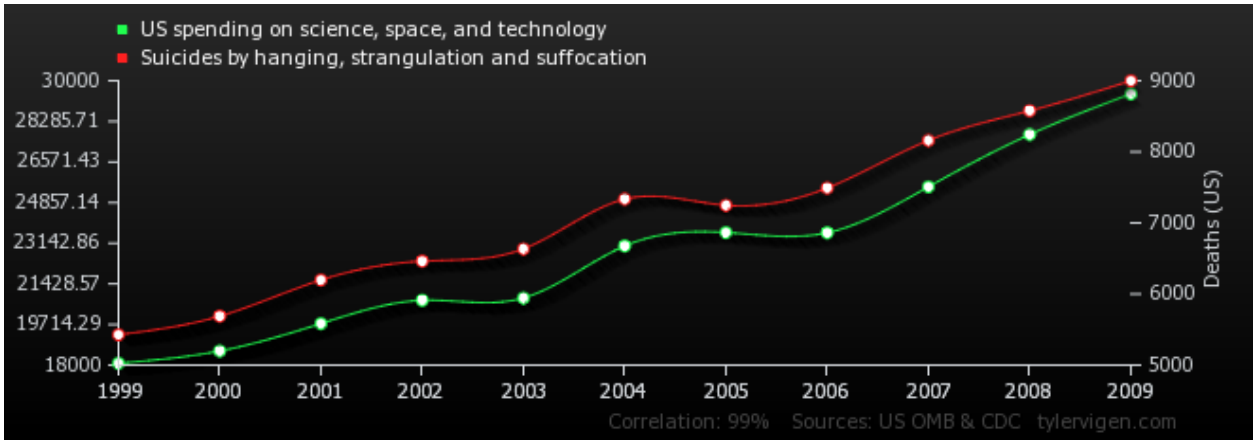
  $\Rightarrow$ we can tune/select these hyperparameters using a validation set approach, which is just one of the numerous cross-validation procedures available—see Basic Course.



Correlation vs. Causation

Warning

Correlation does not always imply causation.



References

1. C. M. Bishop, *Pattern Recognition and Machine Learning,* Springer, 2006. (downloadable)

2. G. James, D. Witten, T. Hastie, R. Tibshirani. *An Introduction to Statistical Learning with Applications in R.* Springer. 2013. (downloadable)

3. M. Asch. *Digital Twins: from Model-Based to Data-Driven.* SIAM, 2022. (extracts)