Overview of Supervised Learning

Yuan Yao

Department of Mathematics
Hong Kong University of Science and Technology

Most of the materials here are from Chapter 2 of Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.

Spring, 2024

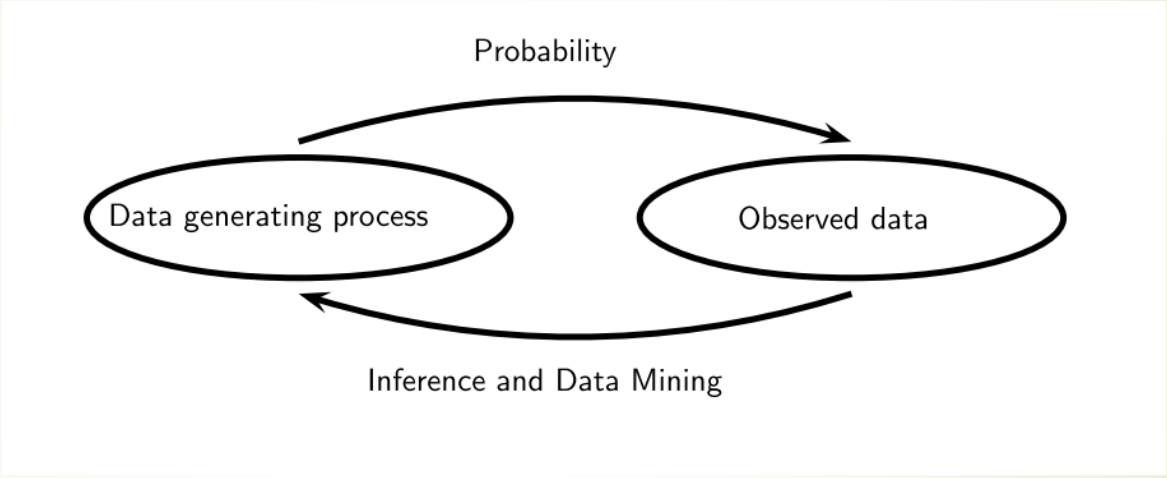
Supervised Learning from Statistical Perspective

The Bias-Variance Trade-Off of Prediction Error

Reference and Textbooks

- Textbook: An introduction to Statistical Learning (ISLR)
- Reference: Elements of Statistical Learning (ESL)
- Will stick with ISL and may cite ESL accassionally.
- Programming languages: R or Python
- Acknowledge the use of the graphics in the textbook/reference for only the purpose of presentation.

Probability vs. Statistical Machine Learning



Forward problem: Probability is a language to quantify uncertainty.

Inverse Problem: Statistics or Machine Learning

Statistics/Data Mining Dictionary

Statisticians and computer scientists often use different language for the same thing. Here is a dictionary that the reader may want to return to throughout the course.

Statistics	Computer Science	Meaning
estimation	learning	using data to estimate an unknown quantity
classification	supervised learning	predicting a discrete Y from X
clustering	unsupervised learning	putting data into groups
data	training sample	$(X_1, Y_1), \ldots, (X_n, Y_n)$
covariates	features	the X_i's
classifier	hypothesis	a map from covariates to outcomes
hypothesis	—	subset of a parameter space Θ
confidence interval	—	interval that contains an unknown quantity with given frequency
directed acyclic graph	Bayes net	multivariate distribution with given conditional independence relations
Bayesian inference	Bayesian inference	statistical methods for using data to update beliefs
frequentist inference	—	statistical methods with guaranteed frequency behavior
large deviation bounds	PAC learning	uniform bounds on probability of errors

Figure: Larry Wasserman's classification of statistical learning vs. machine learning in Computer Science

Supervised vs. Unsupervised Learning

- Supervised Learning
  - **Data**: (x, y), where x is data and y is label
  - **Goal**: learn a function to map $f : x \rightarrow y$
  - **Examples**: classification (object detection, segmentation, image captioning), regression, etc.
  - **Golden standard**: *prediction*!
- Unsupervised Learning
  - **Data**: x, just data and no labels!
  - **Goal**: learn some hidden structure of data x
  - **Examples**: clustering (topology), dimensionality reduction (geometry), density estimation (GAN), etc.
  - **Golden standard**: Non!
- "Self-supervised Learning": cloze task in language models

Related Courses

- Supervised Learning
  - Math 4432: Statistical Machine Learning
    https://yuany-pku.github.io/2018_math4432/
  - MAFS 6010S: Machine Learning and its Applications
  - Math 6380o, Deep learning
    (https://deeplearning-math.github.io/)
  - Best machine learning algorithms: neural networks, random forests, and support vector machines
- Unsupervised Learning
  - Math 4432: Statistical Machine Learning (PCA/clustering)
    https://yuany-pku.github.io/2018_math4432/
  - CSIC 5011, Topological and Geometric Data Reduction
    (https://yao-lab.github.io/2019_csic5011/)
  - Math 6380o, Deep learning (Generative models and GANs)
    (https://deeplearning-math.github.io/)

Outline

Statistical (supervised) learning

- Suppose that we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$. We assume that there is some mapping $f : X = (X_1, X_2, \ldots, X_p) \to Y$, written as

$$Y = f(X) + \epsilon, \tag{1}$$

  where
  - f is some fixed but unknown function to be estimated;
  - ϵ is a random *error* term, which is independent of X and has mean zero;
  - There are two main reasons that we may wish to estimate f: *prediction* and *inference*.

Prediction vs. Inference

- *Prediction* aims to minimize the gap between true value Y and predicted value $\hat{Y} = \hat{f}(X)$, usually measured by loss

$$\mathbb{E}_{(X,Y)}\mathcal{L}(Y, \hat{f}(X))$$

  - The estimation $\hat{f}$, as a mapping from X to Y with unknown parameters (e.g. linear coefficients, support vector machines, neural networks), is a random variable depending on the random data for training.

Prediction vs. Inference

- *Inference* aims to estimate f and its properties, but the goal is not necessarily to make predictions for Y, e.g.
  - **Variable selection**: which predictors are associated with the response?
  - **Model selection**: can the relationship between Y and each predictor be adequately summarized using a linear equation, more complicated ones?
  - **Uncertainty**: how much is the uncertainty of your prediction or estimation given finite information?

Expected Prediction Error in Regression

- Given an estimate $\hat{f}$ and a set of predictors X, we can predict Y using

$$\hat{Y} = \hat{f}(X),$$

- Assume for a moment that both $\hat{f}$ and X are fixed. In regression setting,

$$\begin{aligned} \mathbb{E}(Y - \hat{Y})^2 &= \mathbb{E}[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\textbf{Reducible}} + \underbrace{Var(\epsilon)}_{\textbf{Irreducible}}, \end{aligned} \tag{2}$$

where $\mathbb{E}(Y - \hat{Y})^2$ represents the expected squared error between the predicted and actual value of Y, and $Var(\epsilon)$ represents the variance associated with the error term ϵ. An optimal estimate is to minimize the reducible error.

Reducible vs. Irreducible Error

$$\underbrace{[f(X) - \hat{f}(X)]^2}_{\textbf{Reducible}}$$

- **Reducible error**: $\hat{f}$ will not be a perfect estimate for f, and this inaccuracy will introduce some error. This error is reducible because we can potentially improve the accuracy of $\hat{f}$ by using the most appropriate statistical learning technique to estimate f.
- For example, one may choose different model families for $\hat{f}$:
  - linear models
  - nonlinear models: splines, trees, support vector machines, neural networks

Reducible vs. Irreducible Error

$$\underbrace{Var(\epsilon)}_{\textbf{Irreducible}}$$

- **Irreducible error**: Even if it were possible to form a perfect estimate for f, so that our estimated response took the form $\hat{Y} = f(X)$, our prediction would still have some error in it! This is because Y is also a function of ϵ, which, by definition, cannot be predicted using X. The quantity ϵ may contain unmeasured variables that are useful in predicting Y: since we don't measure them, f cannot use them for its prediction. Therefore, variability associated with ϵ also affects the accuracy of our predictions. This is known as the irreducible error, because no matter how well we estimate f, we cannot reduce the error introduced by uncertainty of ϵ.

How to estimate f?

Depending on whether our ultimate goal is prediction, inference, or a combination of the two, different methods for estimating f may be appropriate.

- Assume that we have observed a set of n different data points. These observations are called the *training data* because we will use these observations to train, or teach, our method how to estimate f.
- Our goal is to apply a statistical learning method to the training data in order to estimate the unknown function f. In other words, we want to find a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y).
- Most statistical learning methods for this task can be characterized as either *parametric* or *non-parametric*.

Parametric Methods

Parametric methods are model-based approach, as it reduces the problem of estimating f down to one of estimating a set of parameters.

- First, we make an assumption about the functional form, or shape, of f.
  - For example, one very simple assumption is that f is linear in X:
$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p. \tag{3}$$
  - Once we have assumed that f is linear, the problem of estimating f is greatly simplified. Instead of having to estimate an entirely arbitrary p-dimensional function $f(X)$, one only needs to estimate the $p + 1$ coefficients $\beta_0, \beta_1, \ldots, \beta_p$.
- After a model has been selected, we need a procedure that uses the training data to fit or train the model, e.g. the most common approach as the least square method.

- **Pro**: Assuming a parametric form for f simplifies the problem of estimating f because it is generally much easier to estimate a set of parameters, such as $\beta_0, \beta_1, \ldots, \beta_p$ in the linear model (3), than it is to fit an entirely arbitrary function f.
- **Con**: The potential disadvantage of a parametric approach is that the model we choose will usually not match the true unknown form of f. If the chosen model is too far from the true f,then our estimate will be poor.
- We can try to address this problem by choosing *flexible* models that can fit many different possible functional forms flexible for f. But in general, fitting a more flexible model requires estimating a greater number of parameters. These more complex models can lead to a phenomenon known as **overfitting** the data, which essentially means they follow the errors, or noise, too closely.

Non-parametric Methods

- Non-parametric methods do not make explicit assumptions about the functional form of f. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.
- Such approaches can have a major advantage over parametric approaches: by avoiding the assumption of a particular functional form for f, they have the potential to accurately fit a wider range of possible shapes for f.
  - Any parametric approach brings with it the possibility that the functional form used to estimate f is very different from the true f, in which case the resulting model will not fit the data well.
  - In contrast, non-parametric approaches completely avoid this danger, since essentially no assumption about the form of f is made.

Non-parametric Methods

- But non-parametric approaches do suffer from a major **disadvantage**: since they do not reduce the problem of estimating f to a small number of parameters, a very *large number of observations* (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for f.

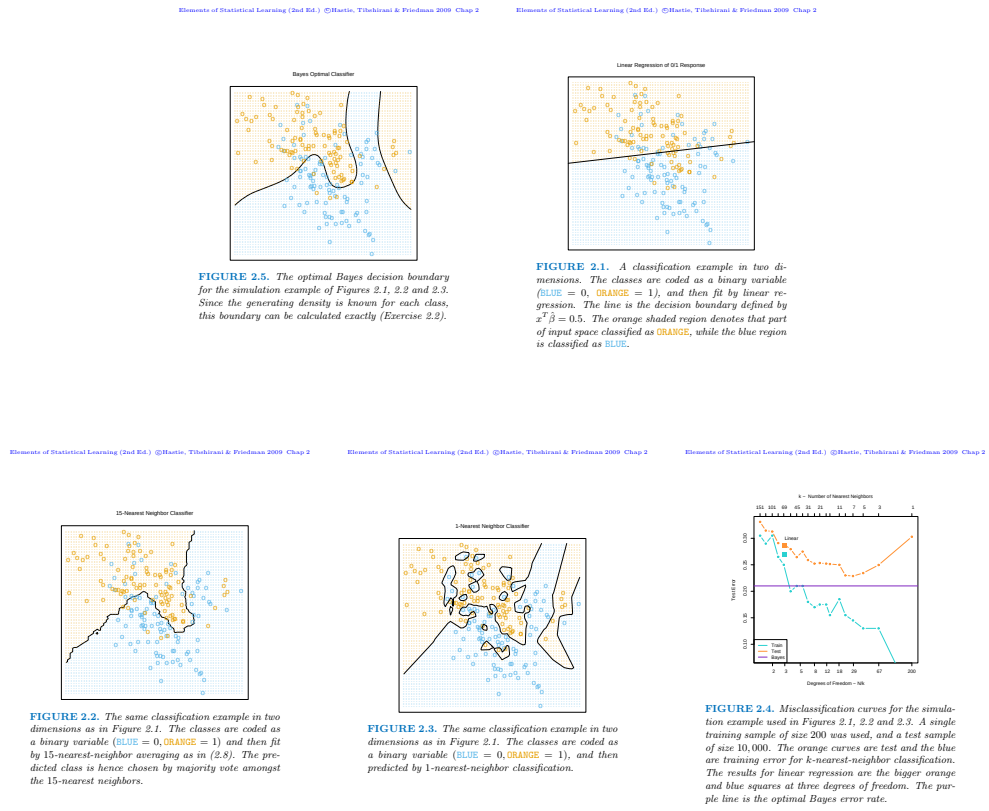


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then fit by linear regression. The line is the decision boundary defined by $x^T\hat{\beta} = 0.5$. The orange shaded region denotes that part of input space classified as ORANGE, while the blue region is classified as BLUE.*

FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.*

FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then predicted by 1-nearest-neighbor classification.*

FIGURE 2.4. *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size 200 was used, and a test sample of size 10,000. The orange curves are test and the blue are training error for k-nearest-neighbor classification. The results for linear regression are the bigger orange and blue squares at three degrees of freedom. The purple line is the optimal Bayes error rate.*

Figure: An illustrative example. Upper Left: A simulated binary classification data with its true generative model (Bayes decision boundary). Upper Right: A fitted Linear model (parametric). Lower Left: Two fitted k-nn models (non-parametric). Lower Right: Test errors.

"No free lunch in statistics"

- Why is it necessary to introduce so many different statistical learning approaches, rather than just a single best method?
- **There is no free lunch in statistics**: no one method dominates all others over all possible data sets. On a particular data set, one specific method may work best, but some other method may work better on a similar but different data set.
- Hence it is an important task to decide for any given set of data which method produces the best results. Selecting the best approach can be one of the most challenging parts of performing statistical learning in practice.
- In this section, we discuss some of the most important concepts that arise in selecting a statistical learning procedure for a specific data set.

Outline

The Goodness of Fit Measure

- Let $f(X)$ be the true function which we aim at estimating from a training data set $\mathcal{D} = \{(x_i, y_i) : i = 1, \ldots, n\}$.
- Let $\hat{f}(X; \mathcal{D})$ be the estimated function from the training data set $\mathcal{D}$.
- In the regression setting, the most commonly-used measure is the *mean squared error* (MSE), given by

$$MSE(\mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2, \tag{4}$$

  where the MSE in (4) is computed using the *training* data and so should more accurately be referred to as the *training MSE*.
- But in general, **we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data**.

- We are really not interested in whether $\hat{f}(x_i) \approx y_i$ (training MSE given by (4) is small); instead, we want to know whether $\hat{f}(x_0)$ is approximately equal to y_0, where (x_0, y_0) is a **previously unseen test observation not used to train the statistical learning method**.
- We want to choose the method that gives the lowest **test MSE**, as opposed to the lowest training MSE. In other words, if we had a large number of test observations, we could compute

$$Ave(\hat{f}(x_0) - y_0)^2, \tag{5}$$

the average squared prediction error for these test observations (x_0, y_0). We'd like to select the model for which the average of this quantity-the test MSE-is as small as possible.

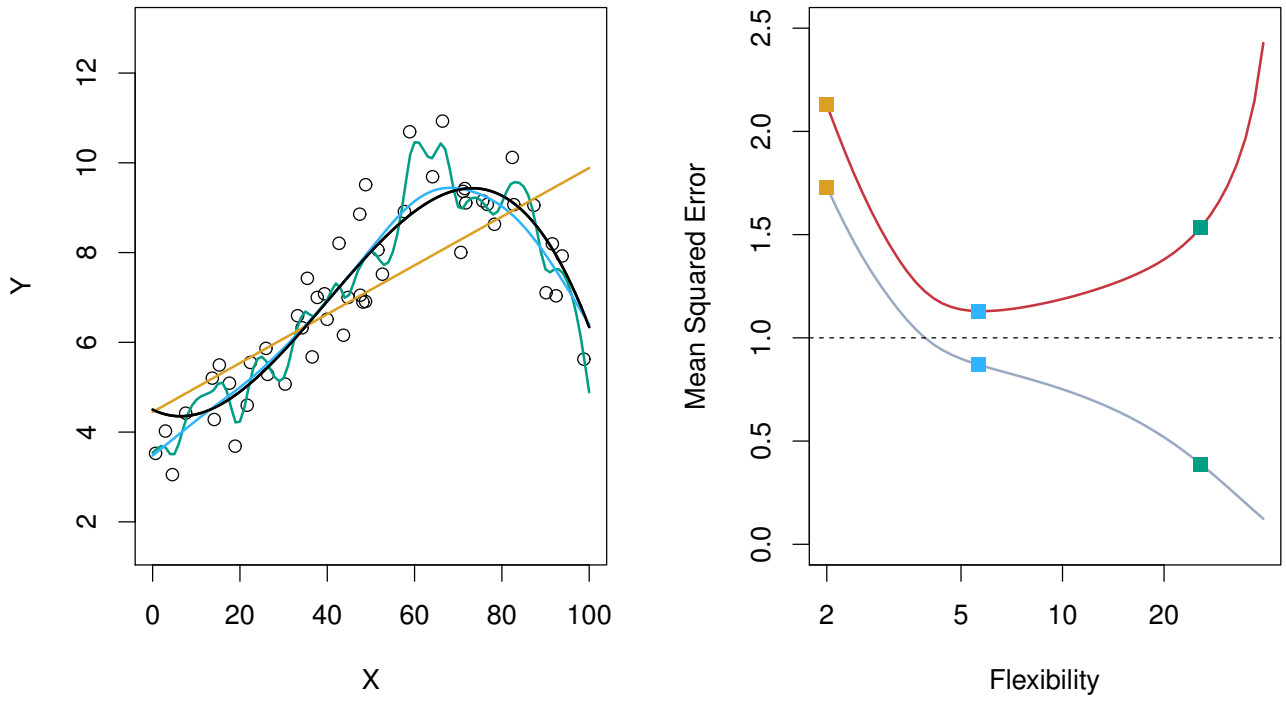


Figure: Illustration. Left: Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Comments on Left panel of Figure 3

- The orange, blue and green curves illustrate three possible estimates for f obtained using methods with increasing levels of flexibility. The orange line is the linear regression fit, which is relatively inflexible. The blue and green curves were produced using smoothing splines with different levels of smoothness.

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \qquad (6)$$

  where λ is a nonnegative tuning parameter. The function g that minimizes (6) is known as a smoothing spline.
- As λ tends to ∞, the function g tends to linear because $\int g''(t)^2 dt$ has to tend to 0.

Comments on Right panel of Figure 3: Training MSE

- It is clear that as the level of flexibility increases, the curves fit the observed data more closely. The green curve is the most flexible and matches the data very well; however, we observe that it fits the true f (shown in black) poorly because it is too wiggly. By adjusting the level of flexibility of the smoothing spline fit, we can produce many different fits to this data.
- The grey curve displays the average training MSE as a function of flexibility, or more formally, the **degrees of freedom** which is a quantity that summarizes the flexibility of a model. The orange, blue and green squares indicate the MSEs associated with the corresponding curves in the left-hand panel.
- The more flexibility, or degree of freedom, the smaller is the training MSE.

Comments on Right panel of Figure 3: Test MSE

- In this example, we know the true function f, and so we can also compute the test MSE over a very large test set, as a function of flexibility. (Of course, in general f is unknown, so this will not be possible.)
- As with the training MSE, the test MSE initially declines as the level of flexibility increases. However, at some point the test MSE levels off and then starts to increase again. Consequently, the orange and green curves both have higher test MSE. The blue curve minimizes the test MSE, which should not be surprising given that visually it appears to estimate f the best.
- The horizontal dashed line indicates $Var(\epsilon)$, the irreducible error in (2), which corresponds to the lowest achievable test MSE among all possible methods.

The Bias-Variance Trade-Off of Prediction Error

- **Fisher's view**: data set $\mathcal{D}$ is a **random selection** from the set of all possible measurements which form the true distribution!
- Expected prediction error

$$\min_{\hat{f}} \mathbb{E}_{\mathcal{D}} \left[f(X) - \hat{f}(X; \mathcal{D}) \right]^2, \tag{7}$$

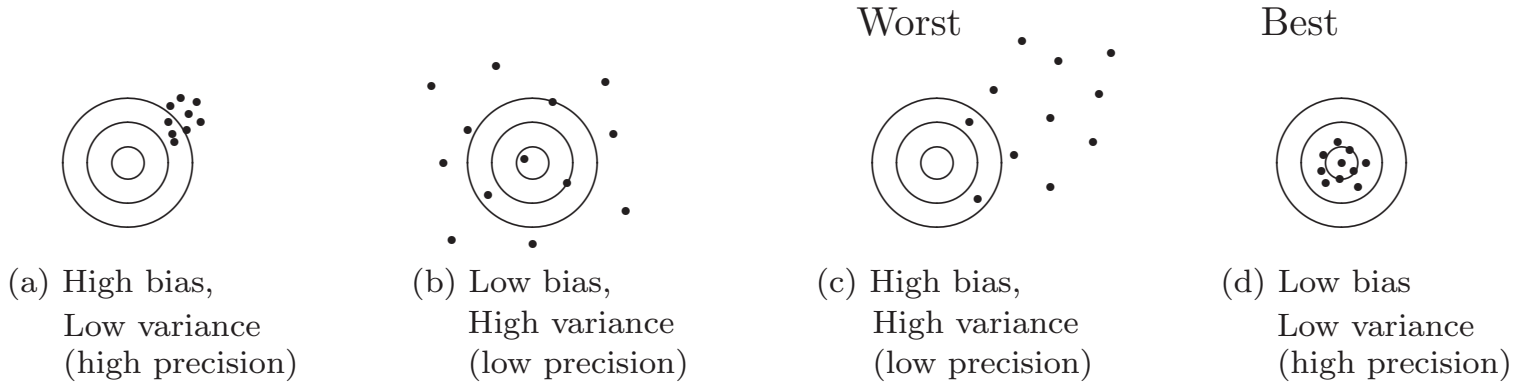
where randomness caused by **random selection** has been taken into account.

- Add and subtract $\mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D}))$ inside the braces, then expand,

$$\begin{aligned}
&\left[f(X) - \hat{f}(X;\mathcal{D})\right]^2 \\
&= \left[f(X) - \mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D})) + \mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D})) - \hat{f}(X;\mathcal{D})\right]^2 \\
&= \left[f(X) - \mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D}))\right]^2 + \left[\mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D})) - \hat{f}(X;\mathcal{D})\right]^2 \\
&\quad + 2\left[f(X) - \mathbb{E}_{\mathcal{D}}[\hat{f}(X;\mathcal{D})]\right]\left[\mathbb{E}_{\mathcal{D}}[\hat{f}(X;\mathcal{D})] - \hat{f}(X;\mathcal{D})\right].
\end{aligned}$$

- Take the expectation with respect to $\mathcal{D}$,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[f(X) - \hat{f}(X;\mathcal{D})\right]^2 \\
&= \underbrace{\left[f(X) - \mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D}))\right]^2}_{\textit{Bias}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left[\mathbb{E}_{\mathcal{D}}(\hat{f}(X;\mathcal{D})) - \hat{f}(X;\mathcal{D})\right]^2\right]}_{\textit{Variance}}
\end{aligned}$$



(a) High bias, Low variance (high precision)

(b) Low bias, High variance (low precision)

(c) High bias, High variance (low precision)

(d) Low bias Low variance (high precision)

- **Bias** refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model.
- **Variance** refers to the amount by which $\hat{f}_{\mathcal{D}}$ would change if we estimated it using a different training data set $\mathcal{D}$.
- **Bias and variance trade-off**: The optimal predictive capability is the one that leads to balance between bias and variance.

Bias-variance tradeoff

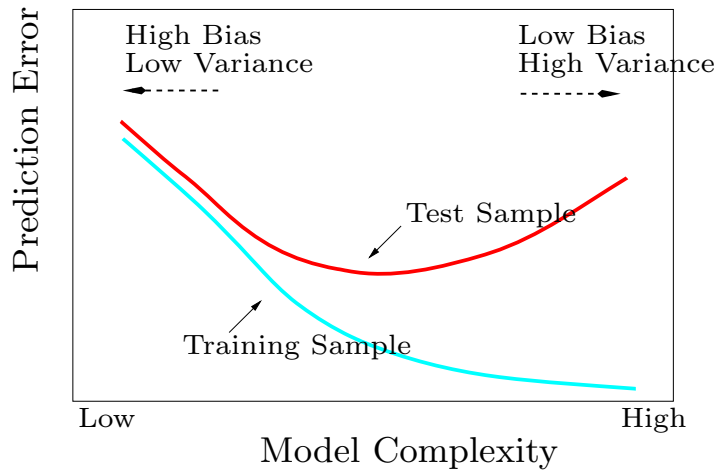
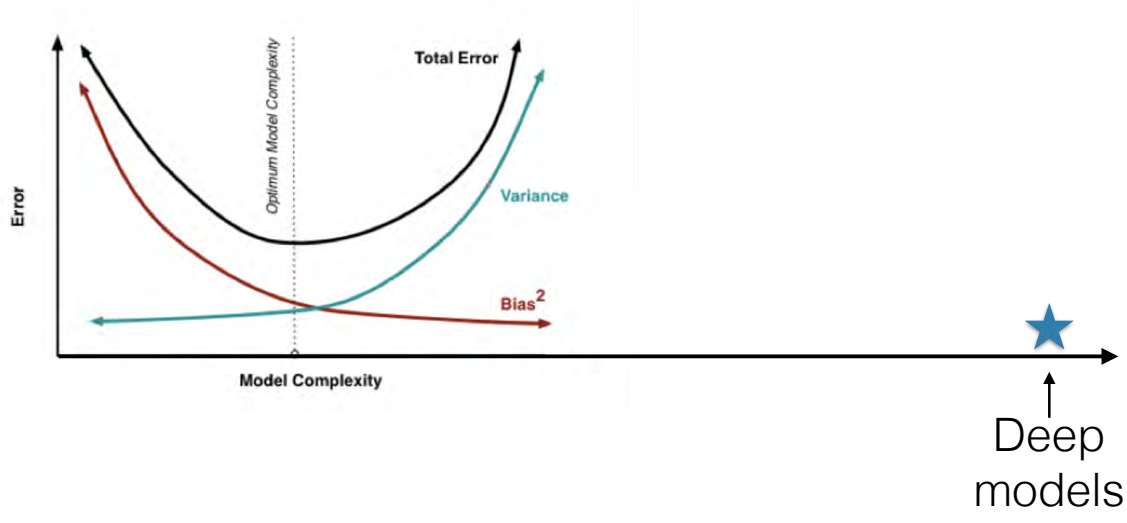


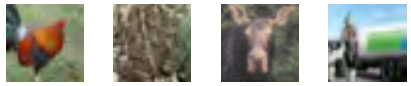
FIGURE 2.11. *Test and training error as a function of model complexity.*

Deep learning: Bias-variance tradeoff?



Models where p>20n are common

Why Big Models Generalize Well?



CIFAR10

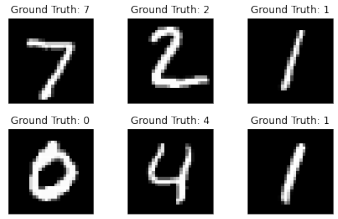
n=50,000
d=3,072
k=10

What happens when I turn off the regularizers?

Model	parameters	p/n	Train ***loss***	Test ***error***
CudaConvNet	145,578	2.9	0	23%
CudaConvNet (with regularization)	145,578	2.9	0.34	18%
MicroInception	1,649,402	33	0	14%
ResNet	2,401,440	48	0	13%

Figure: Why overparameterized deep neural networks do not overfit? Ben Recht, FoCM 2017.

MNIST experiments with Neural Networks



- MNIST: $(x_i, y_i) \in \mathbb{R}^{784 \times [10]}$, $i \in [50,000]$.
- Two-layers neural networks f_N :

$$f_N(x;\theta) = \sum_{j=1}^{N} a_j \sigma\left(\langle w_j, x\rangle\right).$$

- Square loss without regularization.
- Find a local minimizer, report training and test error.
- Perform a sequence of experiments for different N.
- Plot training and test error vs N.

Increasing # parameters

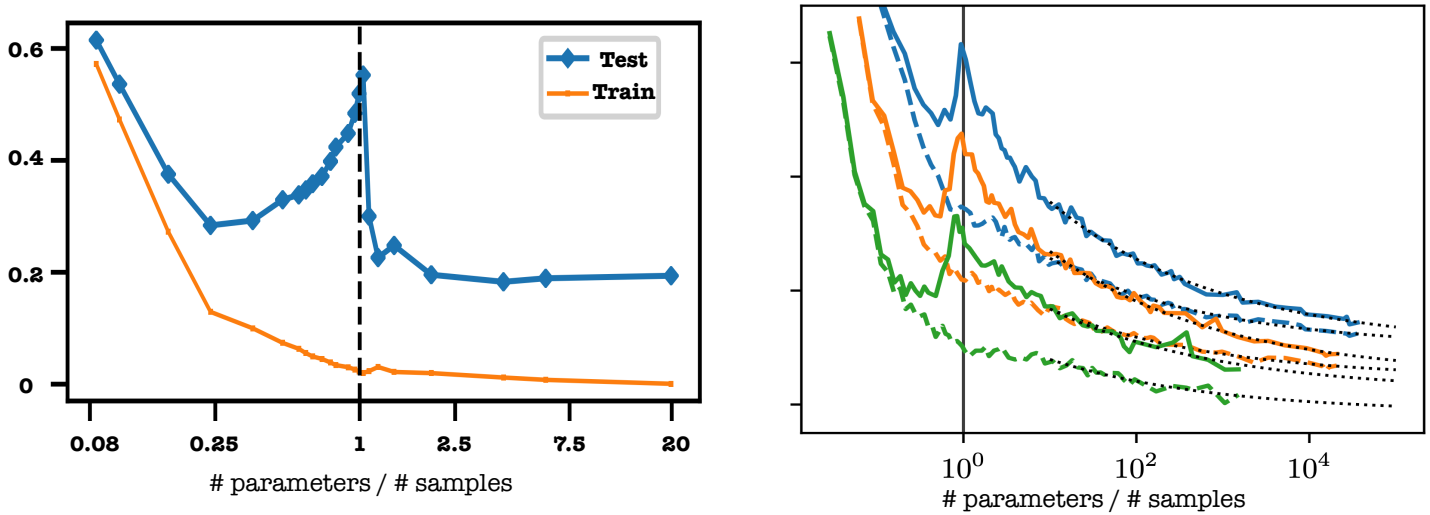


Figure: Experiments on MNIST. Left: [Belkin, Hsu, Ma, Mandal, 2018]. Right: [Spigler, Geiger, Ascoli, Sagun, Biroli, Wyart, 2018].

Similar phenomenon appeared in the literature [LeCun, Kanter, and Solla, 1991], [Krogh and Hertz, 1992], [Opper and Kinzel, 1995], [Neyshabur, Tomioka, Srebro, 2014], [Advani and Saxe, 2017].

Double Descent for Overparameterized Models?

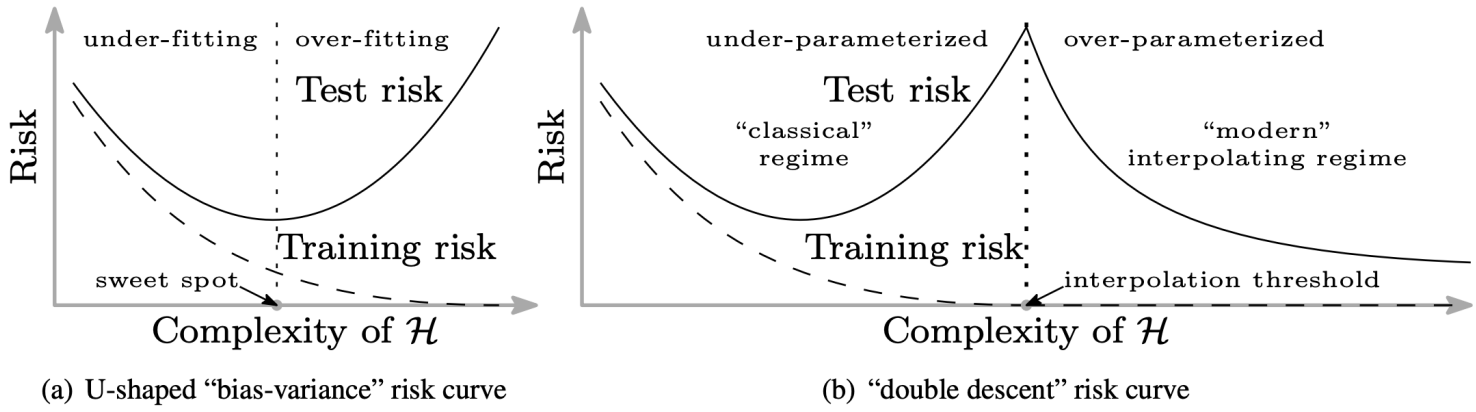


Figure: A cartoon by [Belkin, Hsu, Ma, Mandal, 2018].

Figure: Double descent: 1) Peak at the interpolation threshold; 2) monotone decreasing in the overparameterized region; 3) Global minimum when the number of parameters is infinity.

All models are wrong, but some are useful.

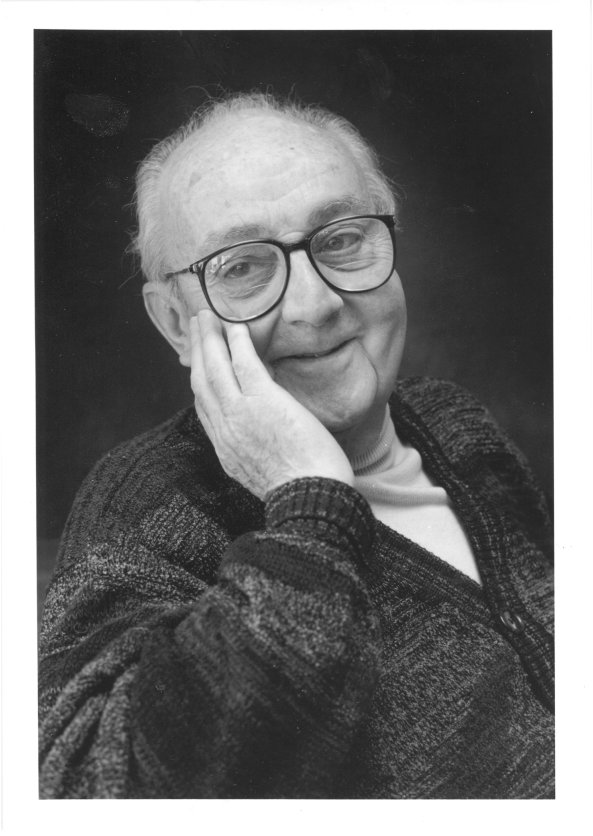


Figure: George Box: "Essentially, all models are wrong, but some are useful."