High Dimensional Data and Statistical Learning

Lecture 2: Statistical Decision Theory

Weixing Song

Department of Statistics
Kansas State University

Outline

Outline

Statistical Learning

Learning is viewed as generalization/inference problem from usually small sets of high dimensional, noisy data.

Where shall we start?

(1). Statistical models are essentially to deal with noise sampling and other sources of uncertainty.

(2). Supervised learning is by far the most understood class of problems.

Regularization provides a fundamental framework to solve learning problems and design learning algorithms.

In this lecture, we present some ideas and tools which are at the core of several supervised learning method and beyond.

Outline

Set-up in Decision Theory

Let

X: input or feature variables;

Y: output or outcome variables (continuous, categorical, ordinal, etc.)

The goal is seeking a function $f(X)$ to make a good prediction of the output Y.

Assume that (X, Y) follows a fixed but unknown joint distribution $P(X, Y)$. Under general conditions, $P(X, Y) = P(Y|X)P(X)$.

A training set of input-output pairs

$$\mathcal{T}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

consists of i.i.d. samples from (X, Y).

Generally speaking, we are looking for a deterministic estimator in a stochastic environment. Therefore, errors cannot be avoided. How to quantify the error?

Loss Function: A loss function L determines the price $L(Y, f(X))$ we have to pay for using $f(X)$ to predict Y.

Examples:

- Square loss (L_2-loss): $L(y, f(x)) = (y - f(x))^2$;
- Absolute deviation loss (L_1-loss): $L(y, f(x)) = |y - f(x)|$;
- L_q-loss: $L(y, f(x)) = |y - f(x)|^q$, $q > 0$;
- Huber loss:

  $$L(y, f(x)) = (y - f(x))^2 I(|y - f(x)| < \delta) + (2\delta|y - f(x)| - \delta^2) I(|y - f(x)| \geq \delta);$$

- Zero-one loss: $L(y, f(x)) = I(y \neq f(x))$.

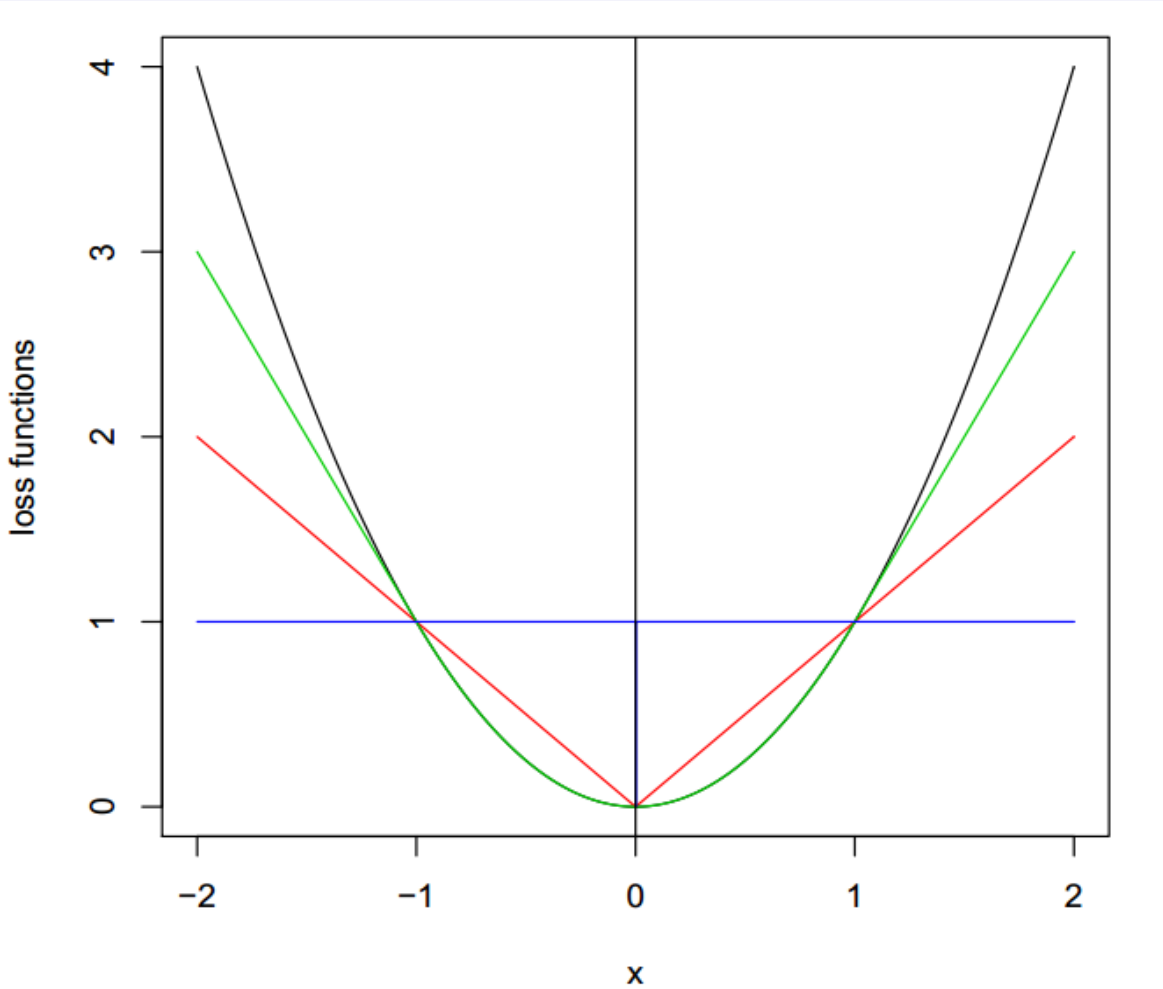


Figure : $0-1$, L_1, L_2, and Huber loss functions

A good estimator f should give small "error" or "loss".

This key idea leads us to a criterion for choosing f: Minimizing the *expected risk*.

Expected Risk (Expected Prediction Error):

$$R(f) = E[L(Y, f(X))] = \int_{X \times Y} L(y, f(x)) p(x, y)\, dxdy$$

is called the expected risk or expected prediction error of using f to predict Y, which measures the loss averaged over the unknown distribution, or the average performance of predicting Y using f under loss L.

Usually, the estimator f is constrained to a function space $\mathcal{F}$, and we are interested in estimating $f_0 \in \mathcal{F}$ such that

$$R(f_0) = \inf_{f \in \mathcal{F}} R(f).$$

The minimizer f_0 is called the target function.

In parametric models, the function f_0 are characterized by some unknown parameters, such as the regression coefficients in parametric regression models. In nonparametric models, f_0 might be only assumed to belong to some function spaces $\mathcal{F}$ with certain smooth conditions.

Risk Minimizer for Squared Error Loss: Regression Problem

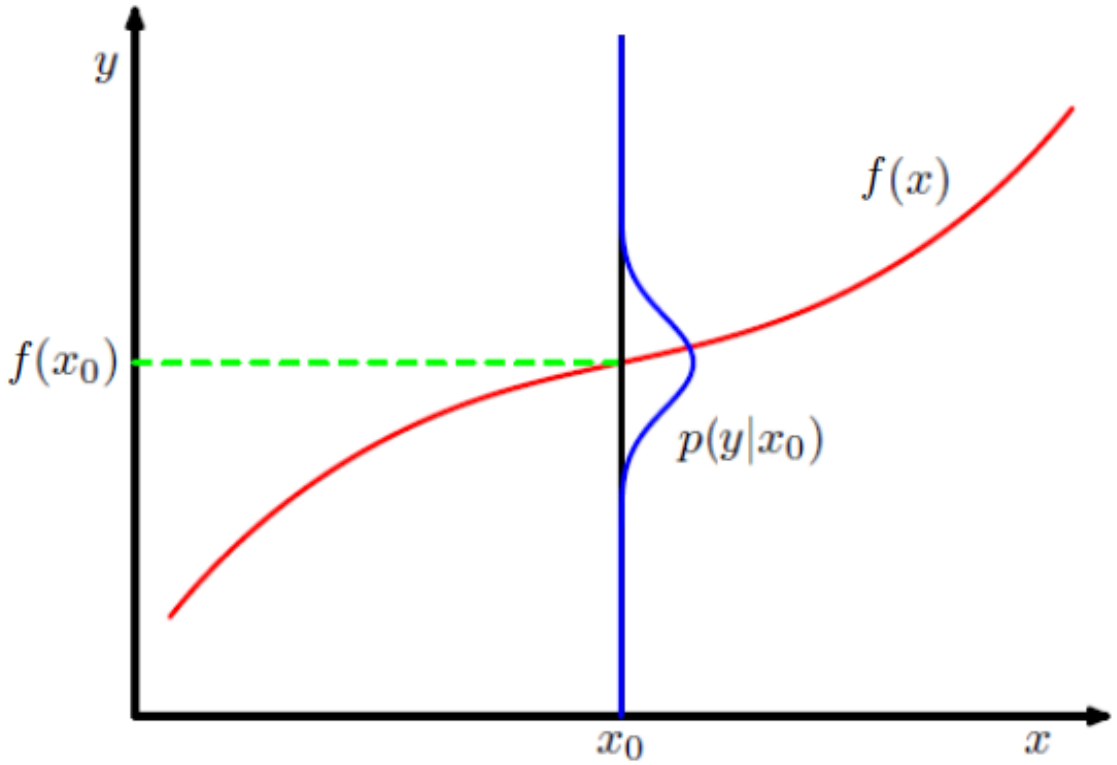
For squared error loss

$$R(f) = E[(Y - f(X))^2] = \int [y - f(x)]^2 p(x, y)\, dxdy = E_X E_{Y|X}\{[Y - f(X)]^2 | X\}.$$

To minimize $R(f)$, it suffices to find

$$f_0(x) = \text{argmin}_f E_{Y|X}\{[Y - f(X)]^2 | X = x\} \Longrightarrow f_0(x) = E(Y|X = x).$$

That is, under the square error loss, the best prediction of Y at any point $X = x$ is the condition expectation $E(Y|X = x)$, also known as the regression function.



The regression function $f(x) = E(Y|X = x)$, which minimizes the expected risk, is given by the mean of the conditional distribution $p(y|x_0)$

How to estimate the target function $E(Y|X=x)$ with $\mathcal{T}_n$?

- Linear Regression (Parametric Method)

  Assume the function class $\mathcal{F} = \{X^T\beta : \beta \in \mathbb{R}^p\}$.

$$R(f) = \int [(y - x^T\beta)^2] p(x,y)\, dxdy$$

$$\frac{dR(f)}{d\beta} = \int 2(xy - xx^T\beta) p(x,y)\, dxdy = 2[E(XY) - E(XX^T)\beta]$$

$$\Longrightarrow \quad \beta = [E(XX^T)]^{-1} E(XY).$$

The least squares method (LS) minimizes the sum of squared error and its solution amounts to replacing the expectations in the above formula by averages over data samples.

It is worth distinguishing between the squared error loss from decision theory and the sum of squares error from LS estimation. We may use more sophisticated techniques than LS such as regularized LS, but they can all be combined with the squared error loss for the purpose of making predictions.

How to estimate the target function $E(Y|X=x)$ with $\mathcal{T}_n$?

- Nearest-neighbor (Nonparametric Method)

  The nearest-neighbor methods attempt to directly implement this recipe using the training data.

  At each point x, we might ask for the average of all those y_i's with input $x_i = x$. Since there is typically at most one observation at any point x, we settle for

  $$\hat{f}(x) = \text{Average}(y_i | x_i \in N_k(x)).$$

  Two approximations:

  (1). Expectation is approximated by averaging over data samples;
  (2). Conditioning at one point is relaxed to conditioning on some region "close" to the target point.

  Under mild conditions, $\hat{f}(x) \to E(Y|X=x)$ as $n, k \to \infty$ and $k/n \to 0$.

  However, we will soon see that the method may fail miserably in high dimensional settings.

Risk Minimizer for 0-1 Loss: Classification Problem

Suppose $Y \in \mathcal{K} = \{1, 2, \ldots, K\}$. Then

$$R(f) = E[L(Y, f(X))] = E_X \sum_{k=1}^{K} L(k, f(X)) P(Y = k|X)$$

$$\Longrightarrow \qquad f_0(x) = \text{argmin}_{g \in \mathcal{K}} \sum_{k=1}^{K} L(k, g) P(Y = k|X = x)$$

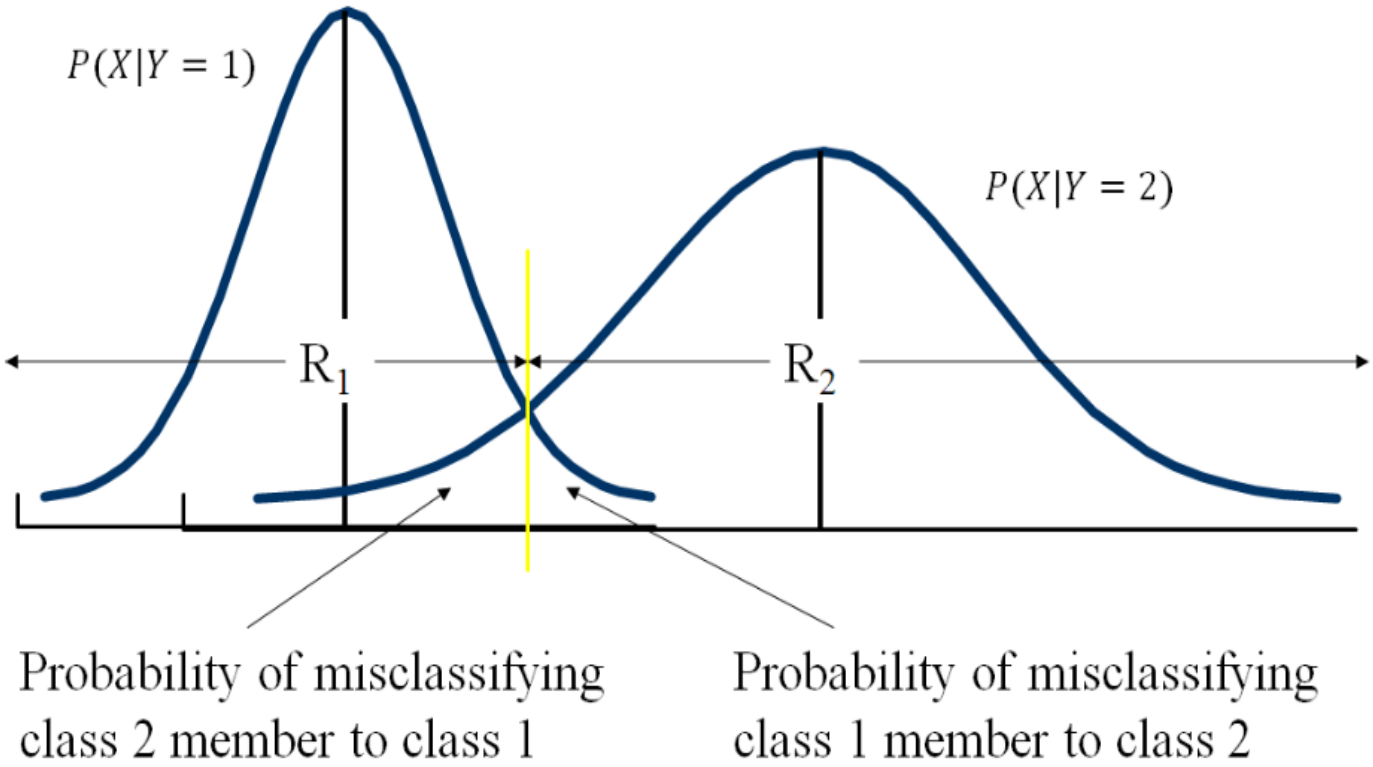
$$\Longrightarrow \qquad f_0(x) = \text{argmin}_{g \in \mathcal{K}} [1 - P(Y = g|X = x)] = \text{argmax}_{g \in \mathcal{K}} P(Y = g|X = x).$$

This reasonable solution is known as the Bayes classfier which classifies to the most probable class, using the conditional distribution $P(Y|X)$ (discrete).

The error rate of the Bayes classifier is called the Bayes rate.

Both the k-nearest neighbor method (majority voting) and linear regression method (dummy variable approach) can be used to estimate this target solution with training data.

When $K = 2$, the probability of misclassifying can be illustrated by the following figure:



We already know that

- For square error loss, the optimal $f(X)$ is the conditional expectation $E(Y|X)$;
- For zero-one error loss, in the classification problem, the optimal $f(X)$ is the Bayes classifier, or the mode of the conditional probability $P(Y = k|X = x)$.

How about other loss function?

- absolute deviation error loss: $f(X)$ =median$(Y|X)$, the median of the conditional distribution;
- Huber loss?
- L_q-loss?

Not all loss functions admit explicit solutions.

Outline

Local Method in High Dimensions

To estimate the regression function $f(x) = E(Y|X = x)$,

- linear regression assumes $f(x)$ is well approximated by a globally linear function, the resulting estimate (LS) is stable but maybe biased;
- the k-nearest neighbor method assumes that $f(x)$ is well approximated by a locally constant function, the resulting estimate is less stable but less biased.

It seems that with a reasonably large training data set, we could always approximate the regression function by k-nearest neighbor averaging, since we should be able to find observations in the close neighborhood of any x and average them to estimate $f(x)$.

Can we?

Recall the "Curse of Dimensionality".

Example 1: Suppose n observations are uniformly distributed in a p-dimensional unit hypercube centered at x_0. Consider the construction of a hypercubical neighborhood of x_0 with edge length $e_p(r)$ which captures a fraction of r observations. We know $e_p(r) = r^{1/p}$. For example, $e_{10}(0.1) = 0.8$.

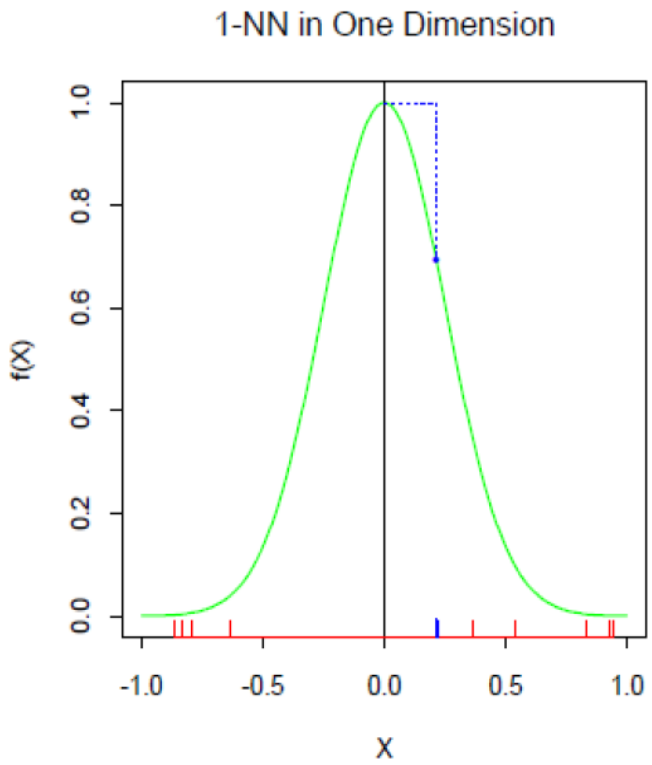
Example 2: Suppose n observations are uniformly distributed in a p-dimensional unit ball centered at x_0. Then the median distance from x_0 to the closest data point is $d(p, n) = (1 - 2^{-1/n})^{1/p}$. For example, $d(10, 500) \approx 0.52$. (**Hint:** The volume of a p dimensional ball of radius r is $V = \pi^{p/2} r^p / \Gamma(1 + p/2)$.)

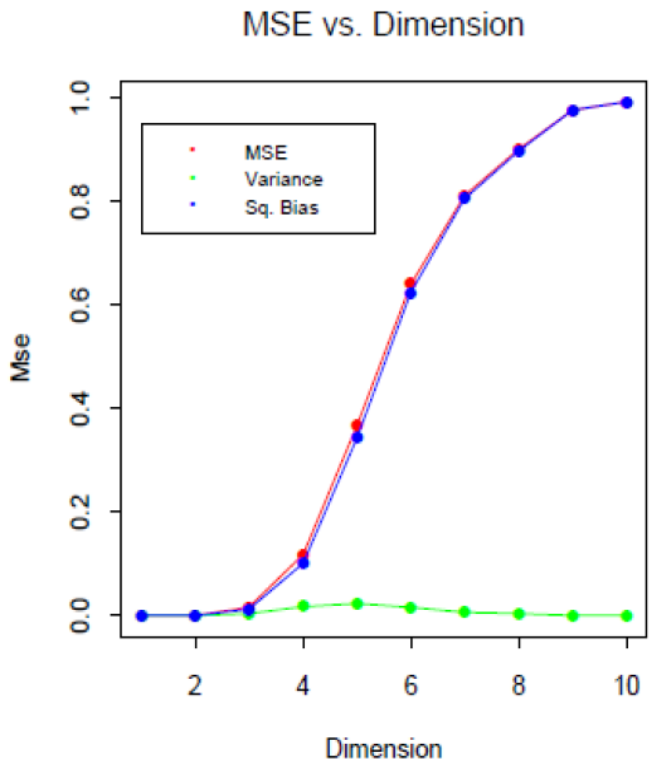
The sampling density is proportional to $n^{1/p}$. If $n = 100$ represents a dense sample for $p = 1$, then $n = 100^{10}$ is the sample size required for the sample sampling density with $p = 10$.

Example 3: Assume there is a deterministic relationship between Y and X: $Y = f(X) = \exp(-8\|X\|^2)$, X is uniformly distributed on $[-1, 1]^p$. Suppose we have $n = 1000$ training samples $\mathcal{T} = \{(x_i, y_i) : i = 1, 2, \ldots, 1000\}$. If we use the 1-nearest neighbor rule to predict y_0 at $x_0 = 0$. Then

$$\begin{aligned} \text{MSE}(x_0) &= E_{\mathcal{T}}(y_0 - \hat{y}_0)^2 = E_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\ &= E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}\hat{y}_0]^2 + [E_{\mathcal{T}}\hat{y}_0 - f(x_0)]^2 \\ &= \text{Var}_{\mathcal{T}}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0). \end{aligned}$$

For small p, both bias and variance are small. As p increases, the nearest neighbor is further and further away from the target, inducing both bias and variance. For large p, the bias tends to -1 and the variance decreases.


1-NN in One Dimension
f(X)
0.0
0.2
0.4
0.6
0.8
1.0
-1.0
-0.5
0.0
0.5
1.0
X


MSE vs. Dimension
Mse
0.0
0.2
0.4
0.6
0.8
1.0
MSE
Variance
Sq. Bias
2
4
6
8
10
Dimension

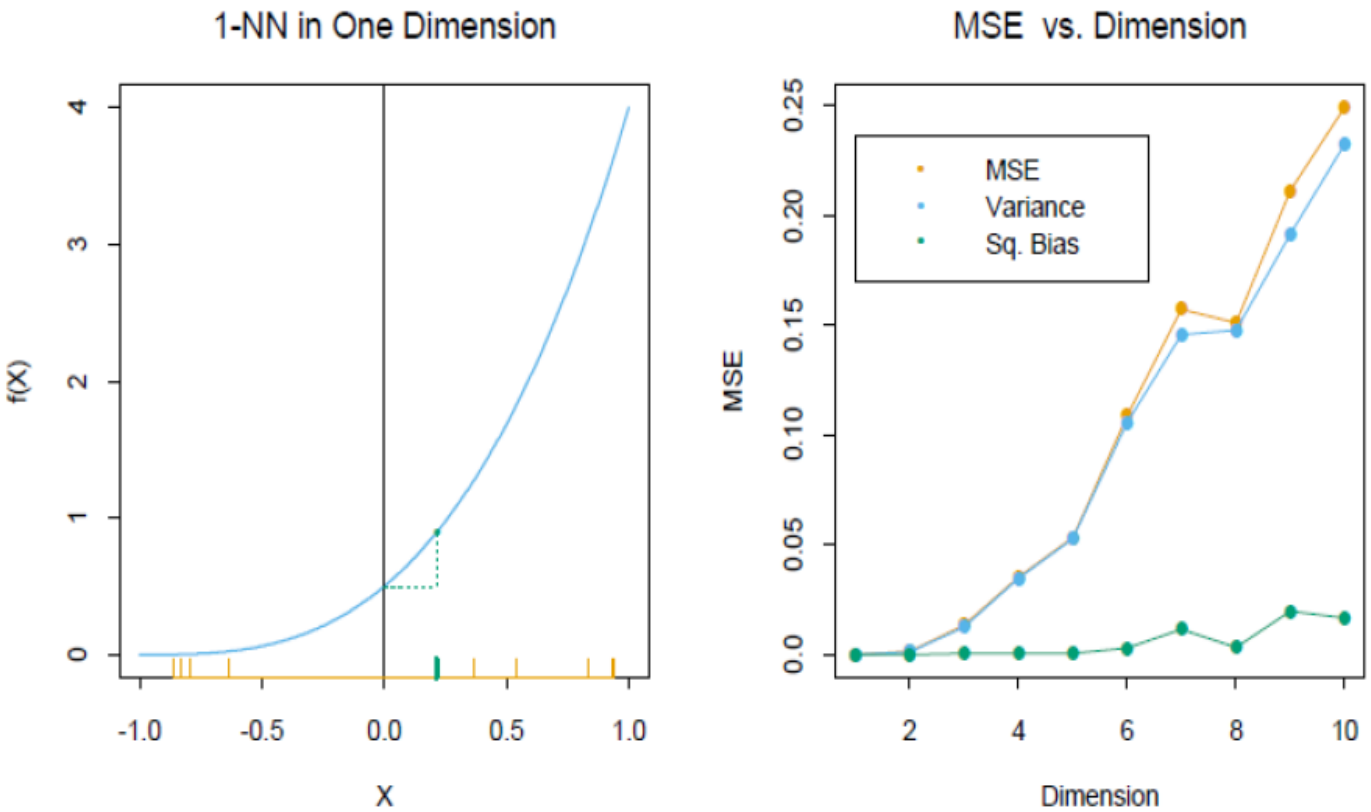


FIGURE 2.8. *A simulation example with the same setup as in Figure 2.7. Here the function is constant in all but one dimension: $F(X) = \frac{1}{2}(X_1 + 1)^3$. The variance dominates.*

The implications of the previous examples for local methods in high dimensions:

- The complexity of the target can grow exponentially with the dimension, so does the "difficulty level" of estimation.
- If we wish to be able to estimate such functions with the same accuracy or precision as functions in low dimensions, then we need the size of our training set to grow exponentially as well.
- In reality, we have very sparse samples in high dimensional space. The nearest neighbor methods may fail us miserably.

Now, assume a linear relationship between Y and X: $Y = X^T\beta + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$. WLOG, assume that $EX = 0$.

Suppose we have n training examples $\mathcal{T} = \{(x_i, y_i) : i = 1, 2, \ldots, n\}$.

Consider a test point x_0 and the prediction value $\hat{y}_0 = x_0^T\hat{\beta}$. Note that

$$\begin{aligned}
\text{EPE}(x_0) &= E_{y_0|x_0} E_{\mathcal{T}}(y_0 - \hat{y}_0)^2 \\
&= \text{Var}(y_0|x_0) + E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)]^2 + [E_{\mathcal{T}}\hat{y}_0 - f(x_0)] \\
&= \sigma^2 + \text{Var}_{\mathcal{T}}(\hat{y}_0) + 0.
\end{aligned}$$

For large n,

$$E_{x_0}\text{EPE}(x_0) = \sigma^2\left[1 + O\left(\frac{p}{n}\right)\right].$$

The above result implies: the expected EPE increases linearly in p, with slope σ^2/n; the growth is slow if σ^2/n is small, therefore, the curse of dimensionality is alleviated.

Implications on Mitigating the Curses of Dimensionality

- By relying on rigid assumptions, the linear model has no bias at all and negligible variance, which the error in k-nearest neighbor methods can be substantially larger.
- By imposing some heavy restrictions on the class of models being fitted, the curse of dimensionality may be mitigated.
- Even in low dimensions, there are many cases where more structured approaches can make more efficient use of the data.
- However, if the assumptions are wrong, all bets are off and local method may dominate.
- A whole spectrum of methods between the rigid linear models and the extremely flexible nearest neighbor models, each with their own assumptions, biases and variances, have been proposes specifically to avoid the exponential growth in complexity of function in high dimensions.

Outline

Empirical Risk Minimization

Notations:

- $P(X, Y)$: joint probability distribution of (X, Y);
- $\mathcal{T}_n$: a training data set of size n;
- $L(Y, f(X))$: loss function;
- $R(f) = \int_{X \times Y} L(y, f(x)) p(x, y) dxdy$: the expected risk;
- $f_0 = \inf_{f \in \mathcal{F}} R(f)$: the target function.

Usually, f_0 can not be obtained explicitly.

We want to find "good" learning algorithm $A(\mathcal{T}_n) = f_n \in \mathcal{F}$, a map from the training set to a set of candidate functions, to minimize the "empirical risk".

The **Empirical Risk Function** is defined as

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$$

which is a natural estimate (empirical version) of the risk function $R(f)$.

A common framework for estimating the best function f_0 is using the empirical risk minimizer

$$\hat{f} = \text{argmin}_{f\in\mathcal{F}}\hat{R}(f).$$

This framework covers the commonly used methods:

- Least squares estimation (LSE) using squared error loss;
- Maximum likelihood estimation (MLE) using negative log-likelihood loss;
- Support vector machine (SVM) using the hinge loss $L(Y, f(X)) = \max(0, 1 - Yf(X))$ for classification in machine learning.

Is minimizing empirical risk on the data always a good idea? The answer is NO.

Solving the empirical risk minimization (ERM) typically requires that $p \leq n$. We have already seen that the ERM can be worse when the dimensionality p is large compared to the sample size n or even fails when $p > n$ (e.g., for ordinary LSE and MLE).

Then, how to improve ERM?

Outline

Penalized Empirical Risk Minimization

What is a "good" learning method/algorithm?

- Is the algorithm consistent?

  The algorithm gets better as we get more data.

- Is the algorithm generalizable?

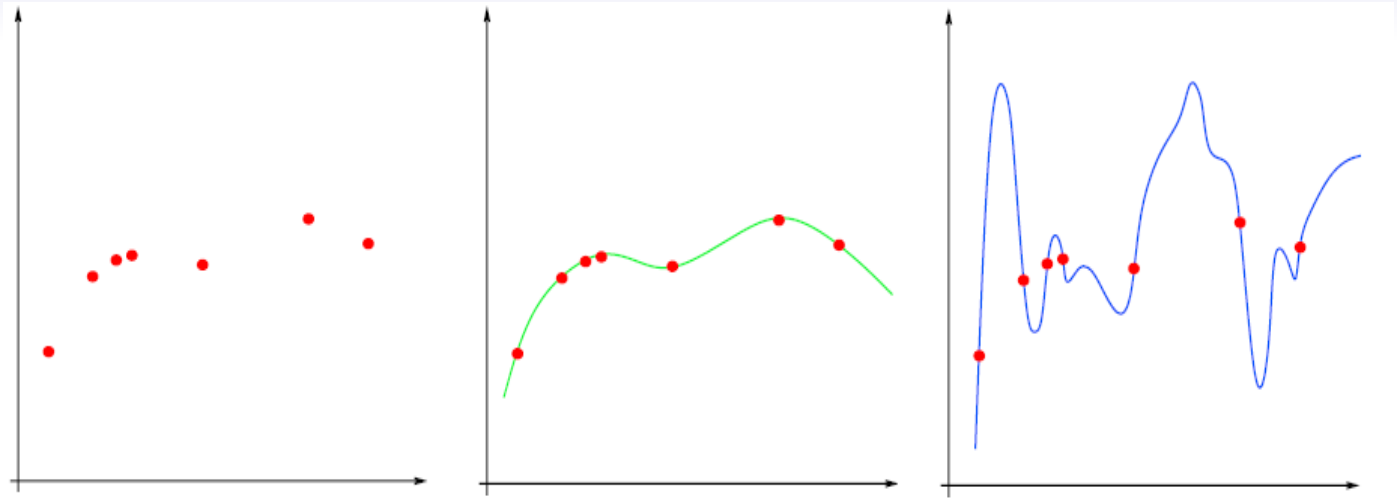
  The training error for the solution must converge to the expected error and thus be a "proxy" for it. Otherwise, the solution would not be "predictive".

- Is the algorithm stable (robust)?

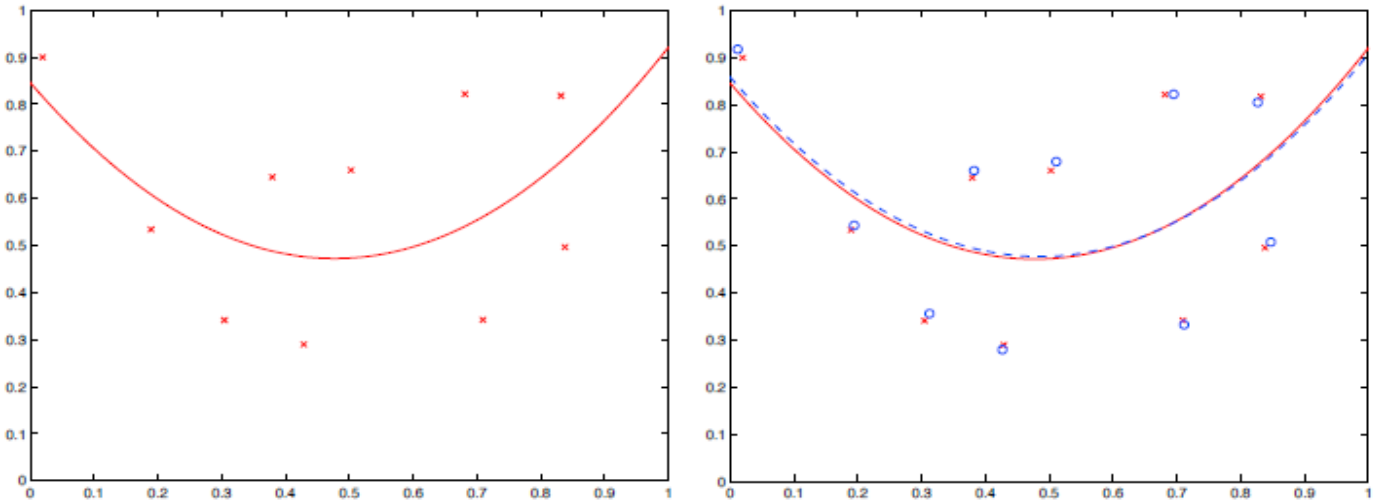
  $\hat{f}_n$ should depend continuously on the training set $\mathcal{T}_n$. In particular, changing one of the training points should affect less and less the solution as n goes to infinity.

An optimization method is well-posed if its solution exists, is unique, and stable.

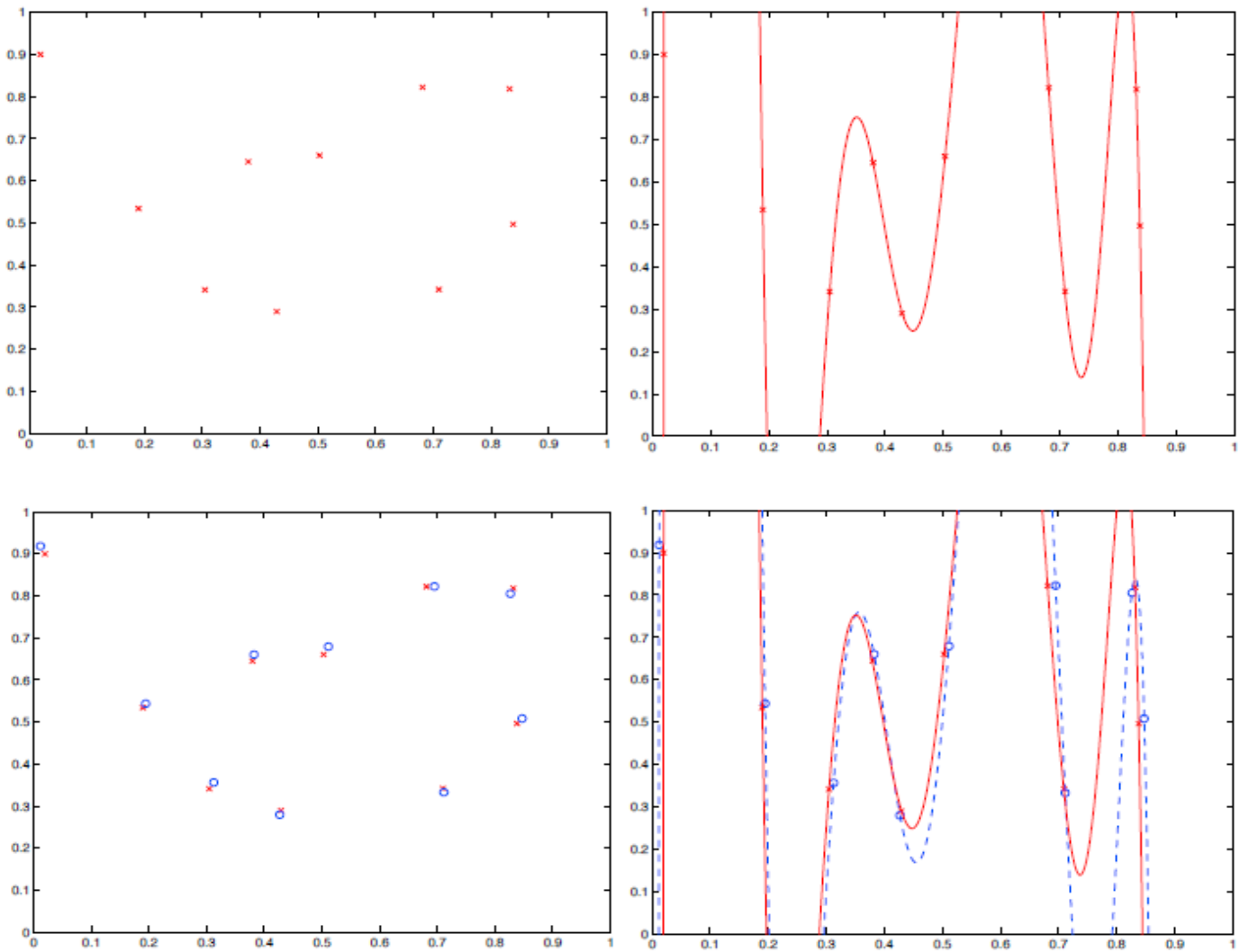
Example 1: Generalizability



Example 2: Stability



Example 3: Stability



How to Achieve Generalizability and Stability?

- In statistical learning, choosing a suitable space of $\mathcal{F}$ is a foremost task. For many algorithms, such as optimization algorithms, it is the space the algorithm is allowed to search. If is often important to choose $\mathcal{F}$ as a function of the amount of available data.
- A generally ill-posed problem such as ERM, can be guaranteed to be well-posed and therefore stable by an appropriate choice or restriction of $\mathcal{F}$. Also the same restrictions may also result in predictability.
- Regularization is the solution!

Regularization: Penalized Empirical Risk Minimization

Regularization is an extremely useful way to restore well posedness and ensure generalizability by constraining the function space $\mathcal{F}$.

Some form of regularized empirical risk minimization produces sparse estimates in high dimensions; it also helps prevent overfitting.

A common regularization framework for high-dimensional data: using the regularized/penalized empirical risk minimization:

$$\hat{f} = \text{argmin}_{f \in \mathcal{F}} \{\hat{R}(f) + p_\lambda(f)\},$$

where $\hat{R}(f) = n^{-1} \sum_{i=1}^{n} L(y_i, f(x_i))$ is the empirical risk function, and $p_\lambda(f)$ is called regularizer or penalty function on the complexity of function f with regularization parameter $\lambda \geq 0$.

We will mainly follow this framework in this course, such as in the area of high-dimensional variable selection. I hope this course will provide us some answers to the following three important questions:

(1). What is the role of penalty function in variable selection? (Intuition)

(2). What are the statistical properties of the regularized estimator? (Theory)

(3). How can we efficiently implement the regularization methods? (Implementation)

We will also discuss some other recent high-dimensional inference methods which are closely related to the regularization methods.

Outline

References

- Hastie, Tibshirani and Friedman (2009). The Element of Statistical Learning. Springer.
- Izenman, A. J. (2008). Modern Multivariate Statistical Techniques. Springer.