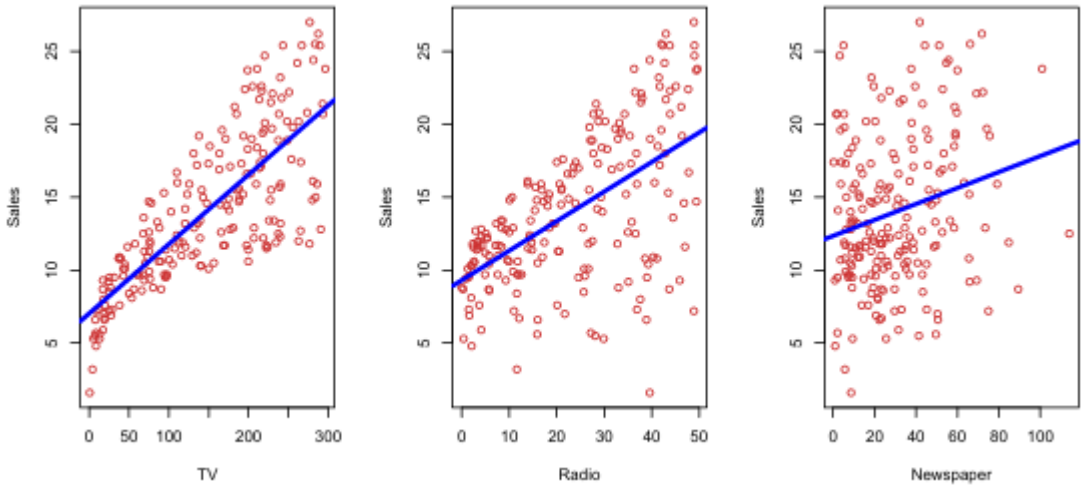
Chapter 2 Statistical Learning

What is Staistical Learning?

Given below is the *Advertising dataset* .The **plot** displays *sales*,in thousands of units, as a function of *TV*, *radio*, and *newspaper* budgets, in thousands of dollars, for 200 different markets. In each plot we show the simple least squares fit of sales to that variable, as described in Chapter 3. In other words, each blue line represents a simple model that can be used to predict sales using TV, radio, and newspaper, respectively.



Can we predict sales using these three?

We would typically like to know the joint relationship between the response: sales and all these three predictors together.

$$sales \approx f(TV, radio, newspaper)$$

Notation

Here *Sales* is a response or target that we wish to predict. We generically refer to the response as Y .

TV is a feature , or input ,or predictor, we name it X_1. Likewise we name *Radio* as X_2 and so on. We can refer to the *input vector* collectively as

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

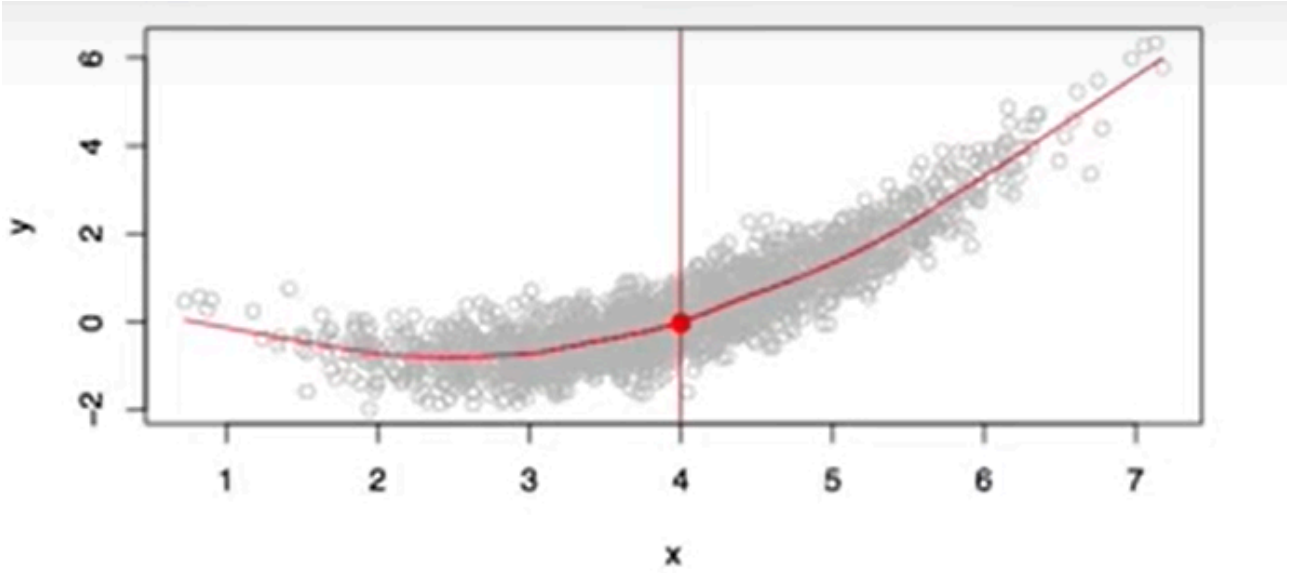
Now we write our model as

$$Y = f(X) + \epsilon$$

where ϵ captures measurement errors and other discrepancies.

$What\ is\ f(X)\ good\ for?$

- With a good $f(X)$ we can make predictions Y at new points $X = x$.
- We can understand which components of $X = (X_1, X_2, \ldots, X_p)$ are important in explaining Y , and which are irrelevant e.g. $Seniority$ and $Years\ of\ Education$ have big impact on $Income$, but $Marital\ Status$ typically doesn't.
- Depending on the complexity of f , we may be able to understand how each component X_j of X effects Y.



- Is there an ideal $f(X)$? In particular, what is good value for $f(X)$ at any selected value of X , say $X = 4$? There can be many Y values at $X = 4$. A good value is

$$f(4) = E(Y|X = 4)$$

$E(Y|X = 4)$ means $expected\ value$ (average) of Y given $X = 4$.
This ideal f is called the **regression function.**

$The\ regression\ function\ f(x)$

- Is also defined for vector X ; e.g.
$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3$
- Is the ideal or optimal predictor of Y with regard to mean-squared prediction error :
$f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions g at all points $X = x$.
- $\epsilon = Y - f(x)$ is the irreducible error - i.e. even if we knew $f(x)$, we would still make errors in prediction , since at each $X = x$ there is typically a distribution of possible Y values.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + Var(\epsilon)$$

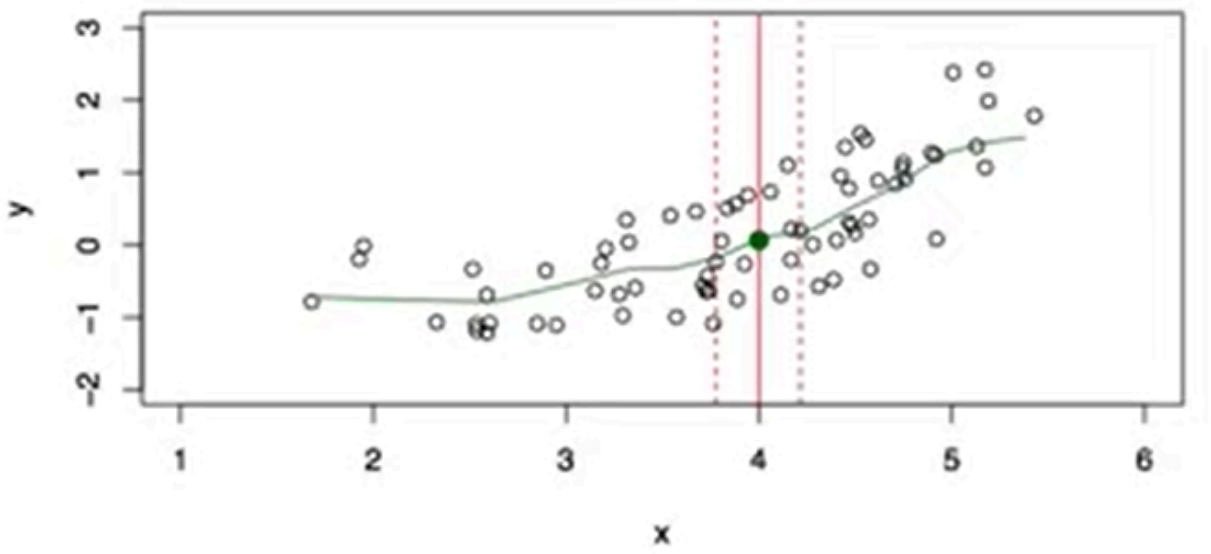
the first term on RHS is the $reducible$ part and second term is the $irreducible$ part. So if we want to improve our model, we need to reduce the reducible error.

$How\ to\ estimate\ f\ ?$

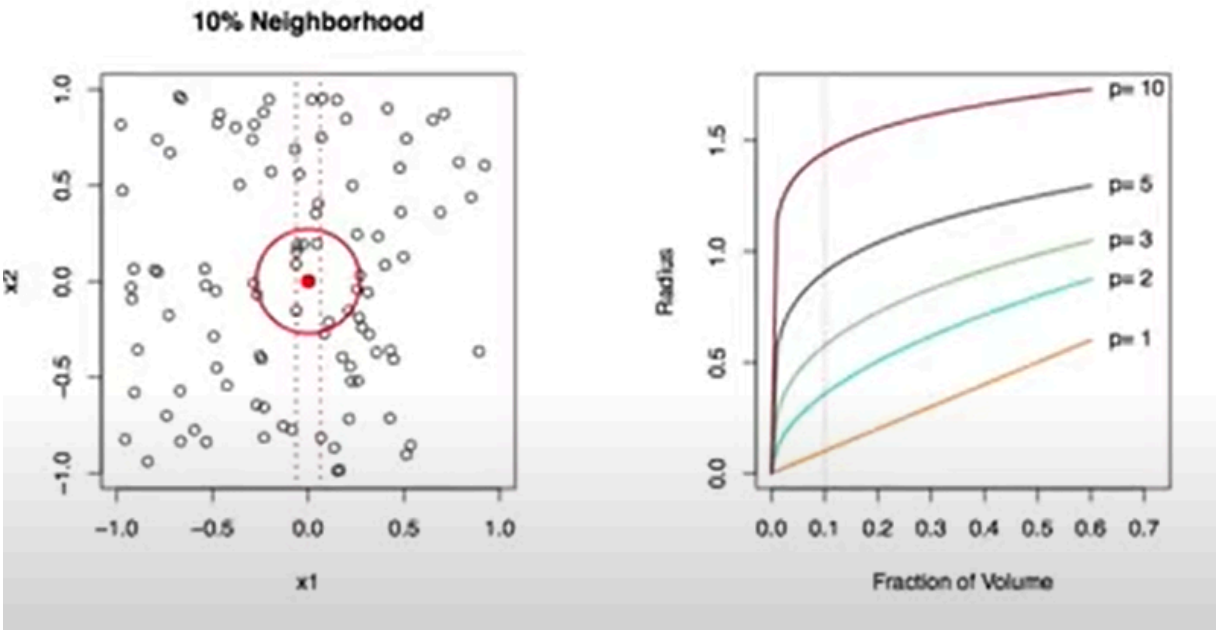
- Typically we have few if any data points with $X = 4$ exactly.
- So we can't compute $E(Y|X = x)$!
- Relax the definition and let

$$\hat{f}(x) = Ave(Y|X \in \mathcal{N}(x))$$

where $\mathcal{N}(x)$ is some neighborhood of x.



- Nearest neighbor averaging can be pretty good for a small p - i.e. $p \leq 4$ and large-ish N.
- Nearest neighbor methods can be really lousy when p is large. Reason: the **curse of dimensionality** . Nearest neighbors tend to be far away in high dimensions.
  - We need to get a reasonable fraction of N values of y_i to average to bring the variance down- e.g 10%
  - A 10% neighborhood in high dimensions need no longer be local, so we lose the spirit of estimating $E(Y|X = x)$ by local averaging.



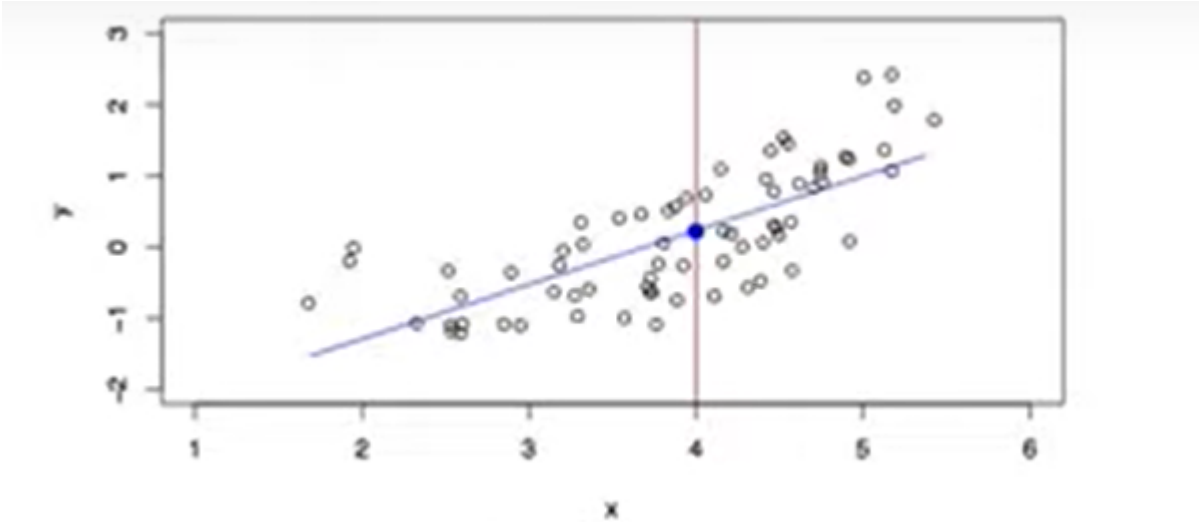
Notice how far you have to go out to get same 10% neighborhood when the dimensions p increases.

Parametric and Structured models

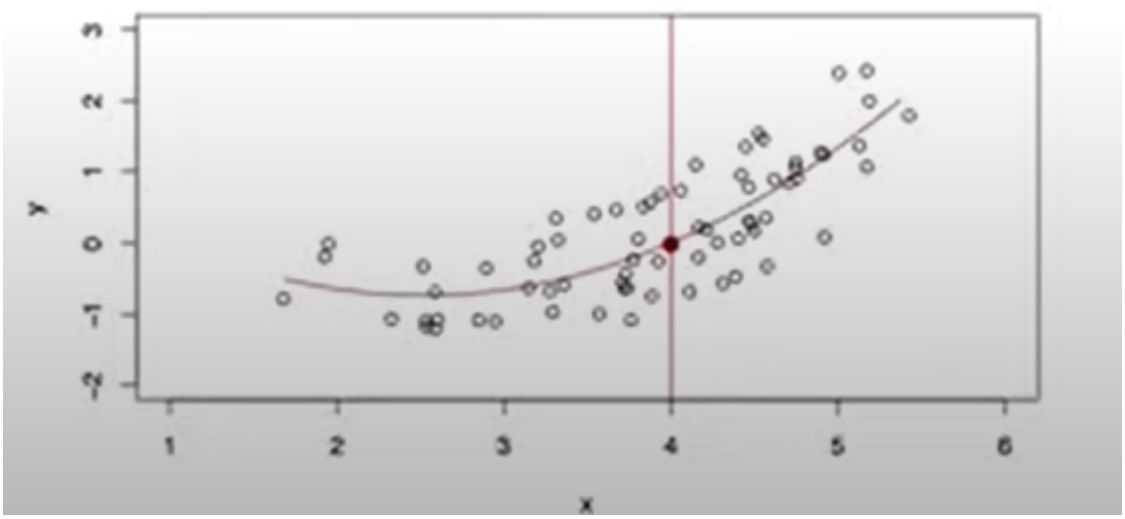
The linear model is an important example of a parametric model:

$$f_L(x) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

- A linear model is specified in terms is specified in terms of $p + 1$ parameters $\beta_0, \beta_1, \ldots, \beta_p$.
- We estimate the parameters by fitting the model to training data.
- Although it is almost never correct, a linear model often serves as a good and interpretable approximation to the unknown true function $f(x)$
- A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit here



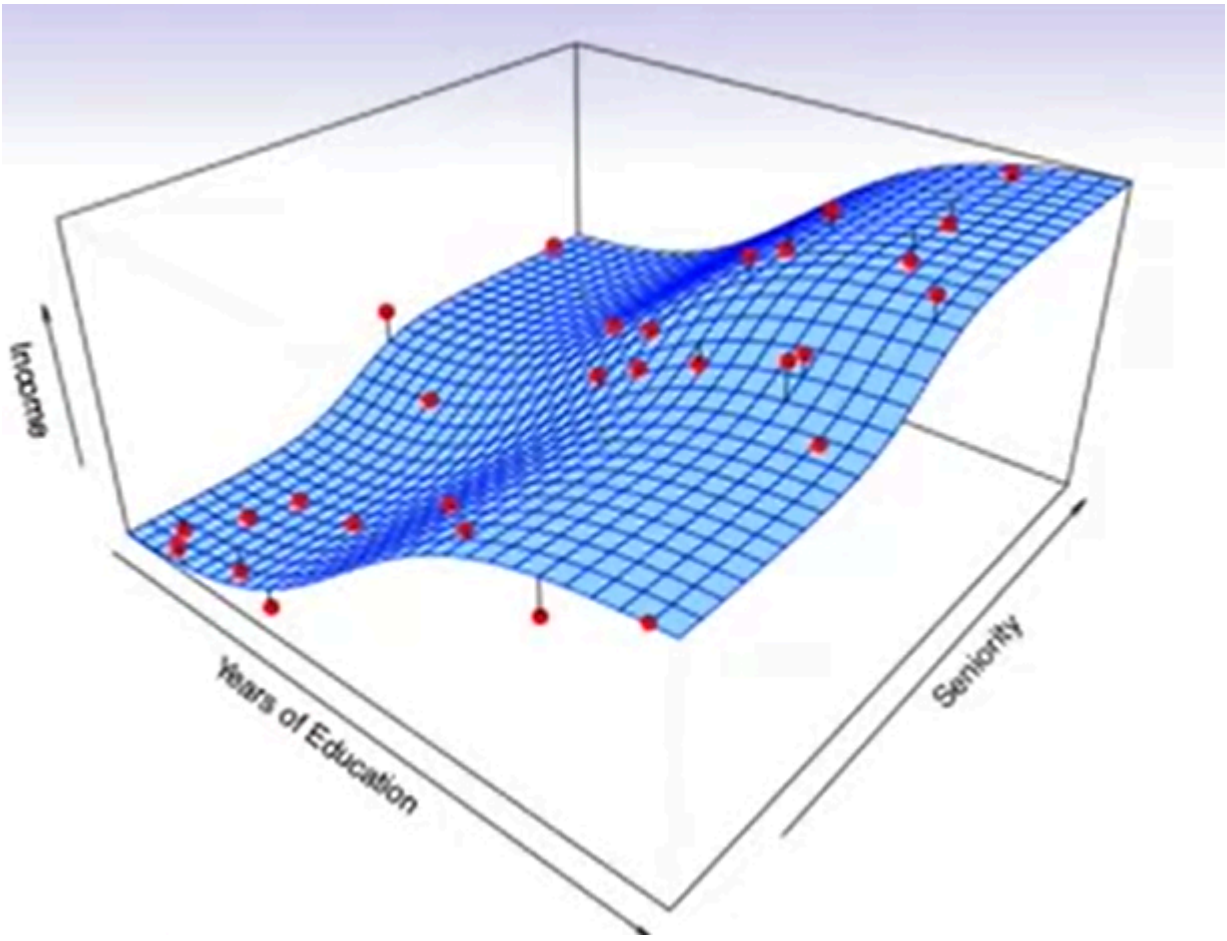
- A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.



- Here's another simulated example below. Red points are simulated values for $income$ from the model

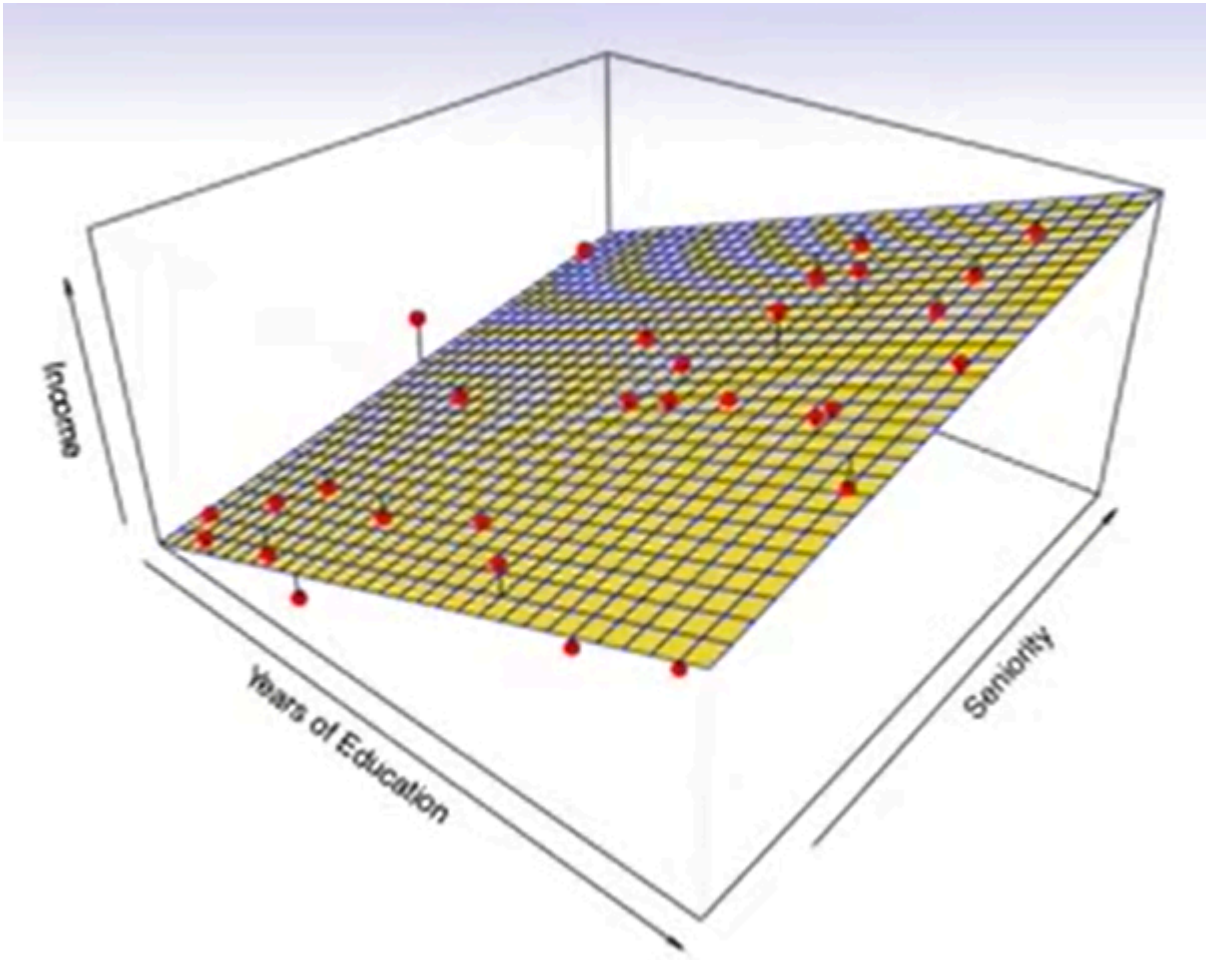
$$income = f(education, seniority) + \epsilon$$

f is the blue surface.

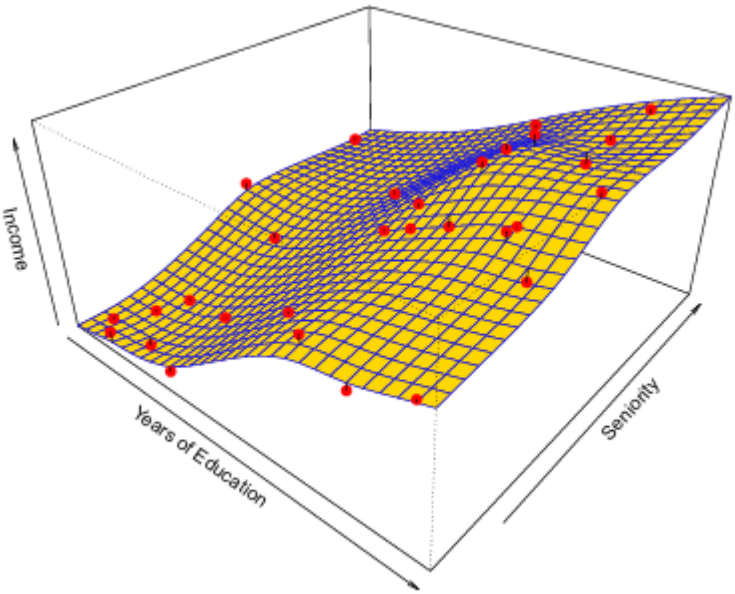


- Here's a linear regression model fit to the simulated data.

$$\hat{f}_L(education, seniority) = \hat{\beta}_0 + \hat{\beta}_1 \cdot education + \hat{\beta}_2 \cdot seniority$$



- Here's a more flexible regression model $\hat{f}_S(education, seniority)$ fit to the simulated data. Here we use a technique called a $thin-plate\ spline$ to fit a flexible surface.



- Even more flexible regression model fit to the simulated data that makes zero errors on the training data. This is also known as **overfitting.**

$Some\ Trade-offs$

- Prediction accuracy versus interpretability :- Linear models are easy to interpret ; thin-plate splines are not.
- Good fit versus over-fit or under-fit :- How do we know when the fit is just right?
- Parsimony versus black-box :- We often prefer a simple model involving fewer variables over a black-box predictor involving them all.

The above image is a representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.
What does flexibility even signifies? take example of a flexible method like spline, it means it can generate a much wider range of possible shapes to estimate f , meanwhile inflexible methods like linear regression can generate lines, planes or hyperplanes only.

Assessing Model Accuracy

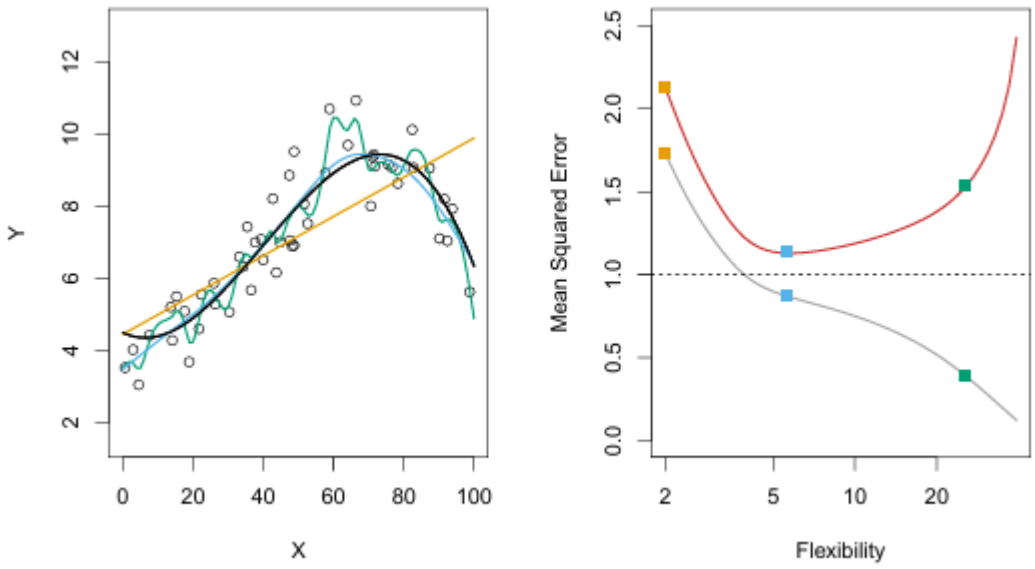
Suppose we fit a model $\hat{f}(x)$ to some training data $Tr = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

- We could compute the average squared prediction error over Tr:

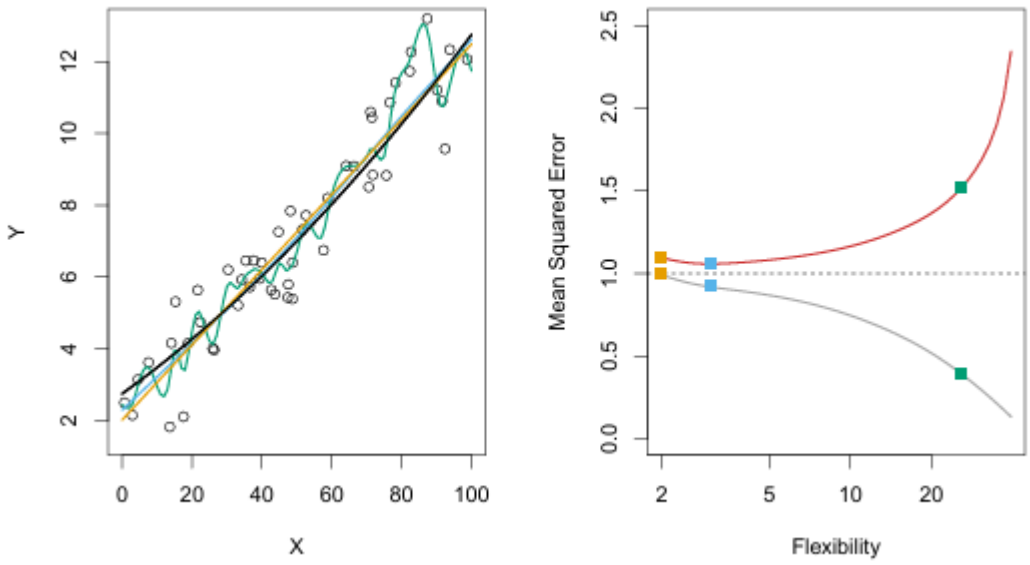
$$MSE_{Tr} = Ave_{i \in Tr}[y_i - \hat{f}(x_i)]^2$$

  This may be biased towards more overfit models.
- Instead we should , if possible , compute it using fresh $test$ data $Te = \{x_i, y_i\}_1^M$:

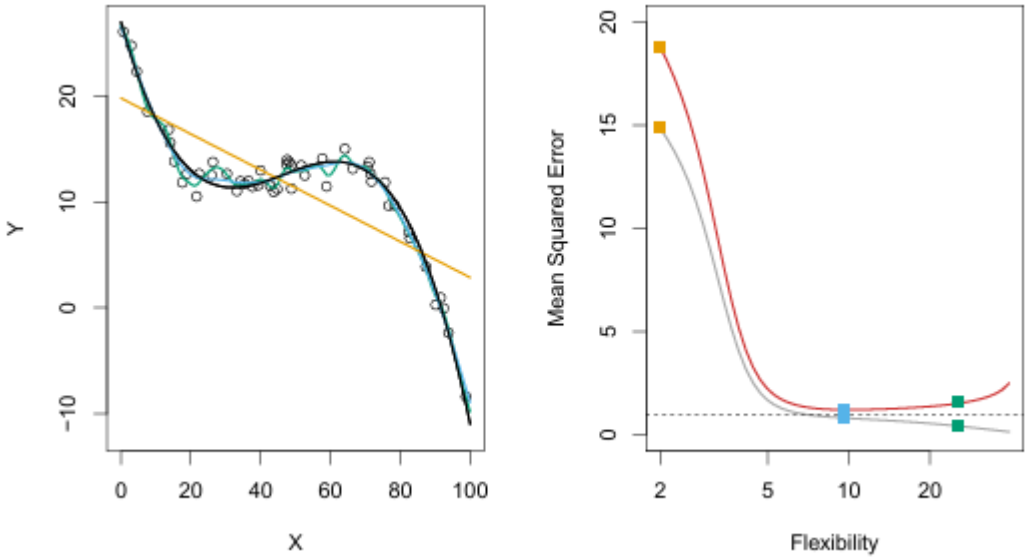
$$MSE_{Te} = Ave_{i \in Te}[y_i - \hat{f}(x_i)]^2$$



Black curve is the truth. Red curve in right is MSE_{Te} , grey curve is MSE_{Tr} . Orange, blue and green curves/squares corresponds to fits of different flexibility.



In the above figure, the truth is smoother , so the smoother fit and linear model does well.



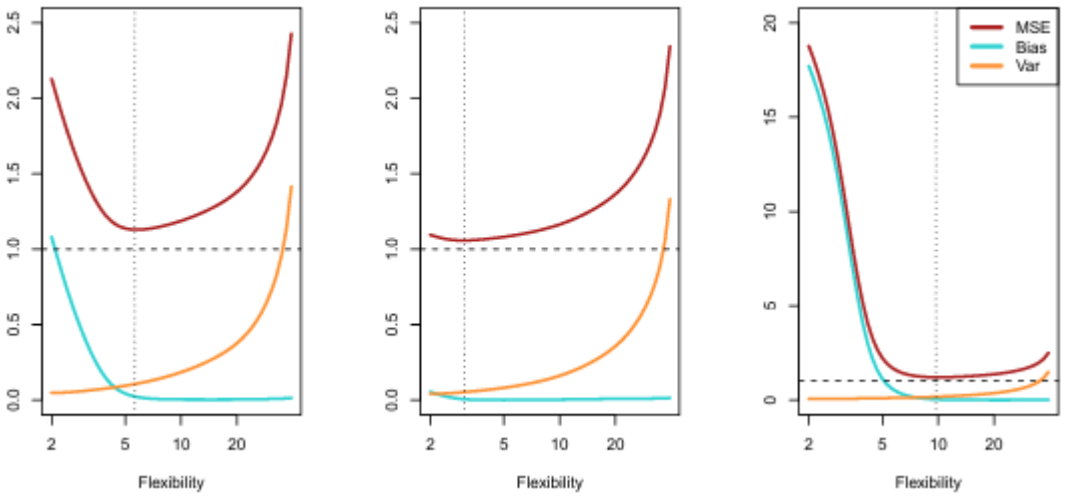
In the above figure , the truth is wiggly and the noise is low. so the more flexible fits do well.

Bias-Variance Trade off

- Suppose we have fit a model $\hat{f}(x)$ to some training data Tr, and let (x_0, y_0) be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E[Y|X = x]$), then

$$E\Big(y_0 - \hat{f}(x_0)\Big)^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\epsilon).$$

- The expectation averages over the variability of y_0 as well as the variability in Tr. Note that $\text{Bias}(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$.
- Typically as the **flexibility** of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on average test error amounts to a **bias-variance trade-off**.



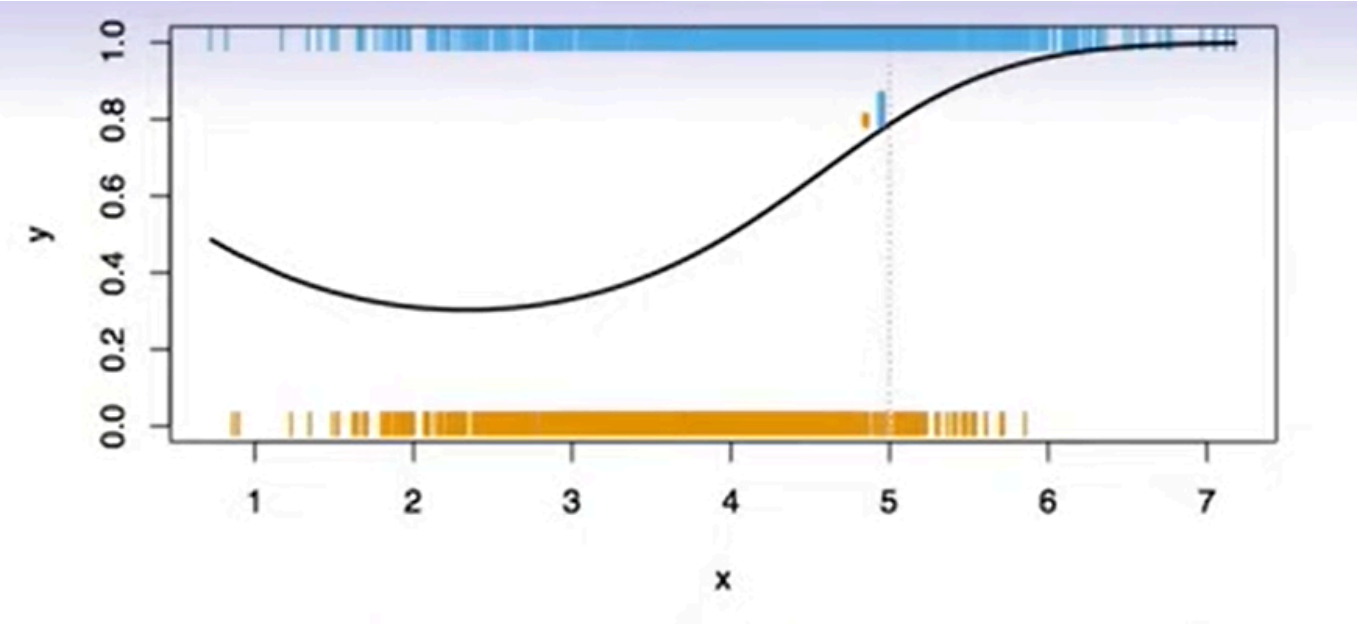
Squared bias (blue curve), variance (orange curve), Var( ) (dashed line), and test MSE

(red curve) for the three data sets . The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.

Classification Problems

Here the response variable Y is qualitative - e.g. email is one of $C = (spam, ham)$ (ham = good email) , digit class is one of $C = \{0.1, 2, \ldots, 9\}$. Our goal is to :

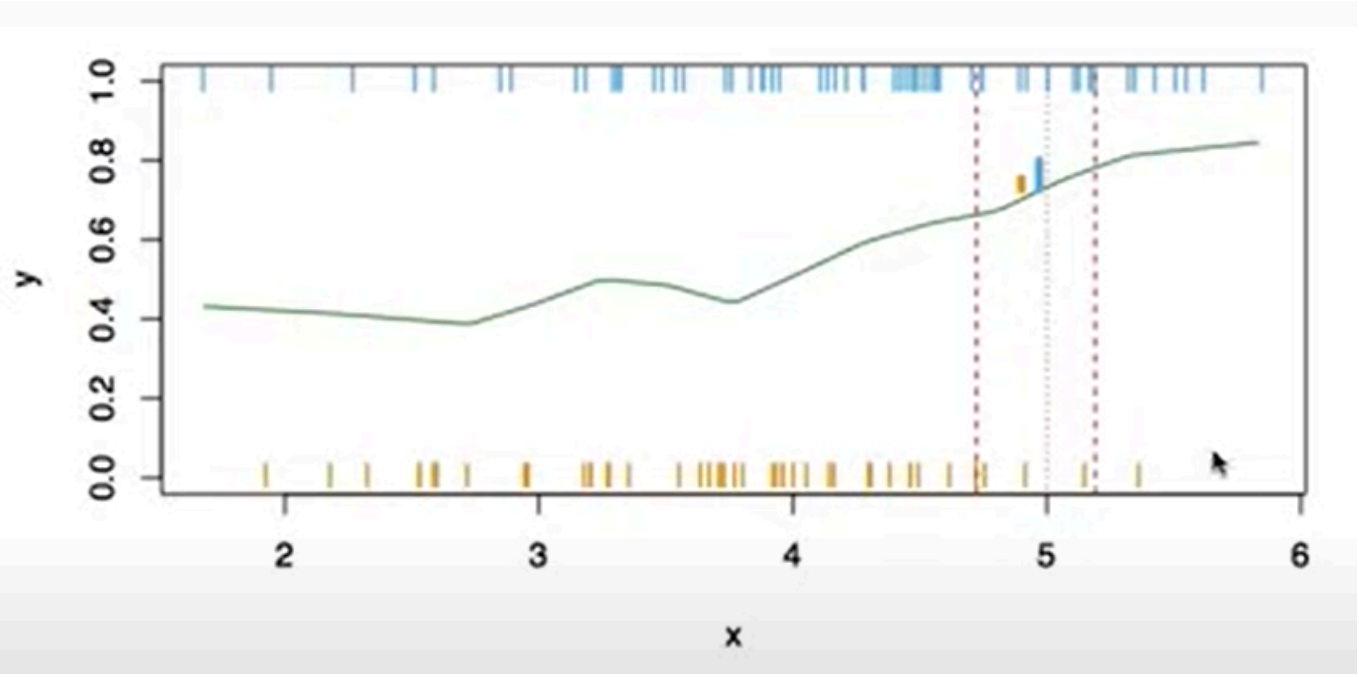
- Build a classifier $C(X)$ that assigns a class label from C to a future unlabeled observation X
- Access the uncertainty in each classification.
- Understand the roles of the different predictors among $X = X_1, X_2, \ldots, X_p$



Is there an ideal $C(X)$? Suppose the K elements in C are numbered $1, 2, \ldots, K$ .Let

$$p_k(x) = Pr(Y = k|X = x), k = 1, 2, \ldots, K$$

These are the **conditional class probabilities** at x ; e.g. see little bar plot at $x = 5$. Then the $Bayes\ optimal$ classifier at x is
$C(x) = j$ if $p_j(x) = \max\{p_1(x), p_2(x), \ldots, p_K(x)\}$

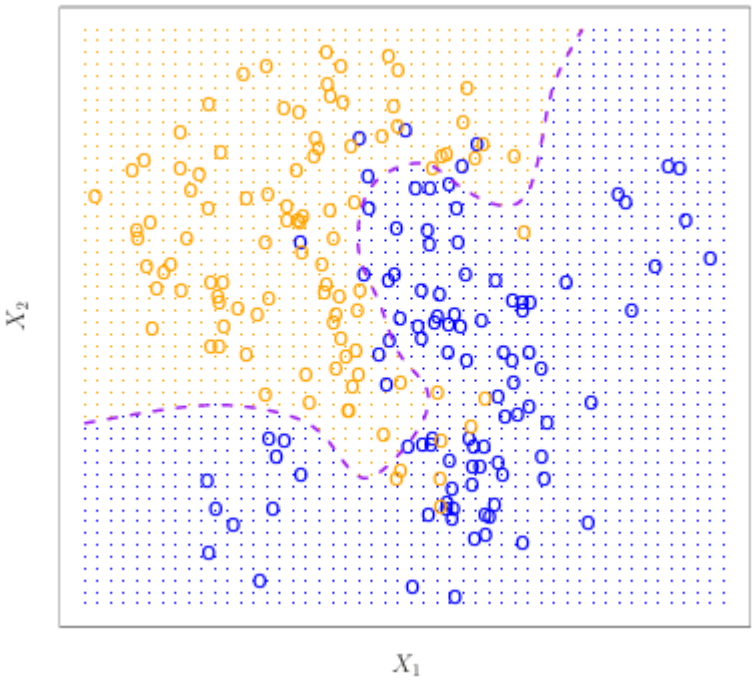


Nearest-neighbor averaging can be used as before. Also breaks down as dimension grows. However, the impact on $\hat{C}(x)$ is less than on $\hat{p}_k(x), k = 1, 2, \ldots, K$.

- Typically we measure performance of $\hat{C}(x)$ using the misclassification error rate:

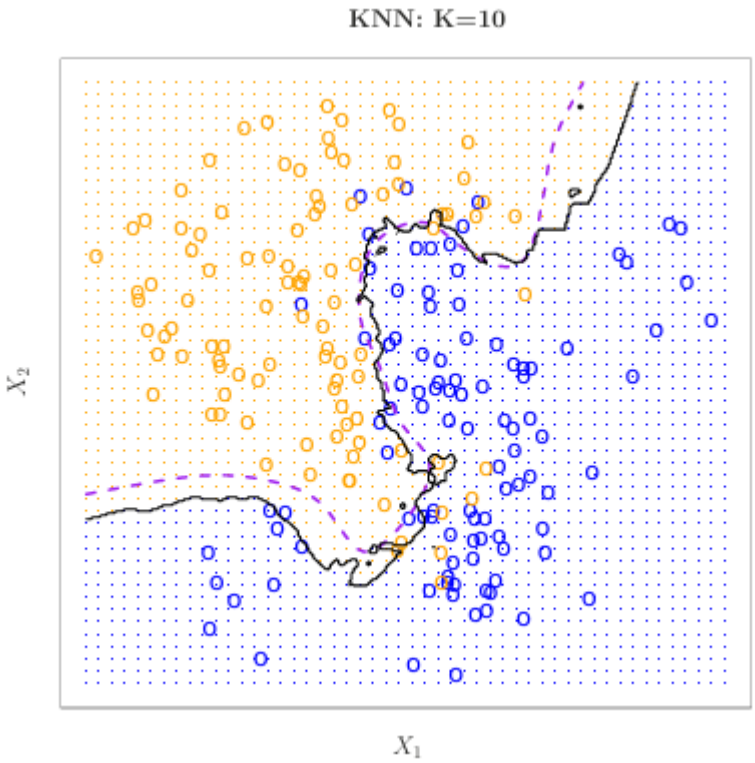
$$Err_{Te} = Ave_{i \in Te} I[y_i \neq \hat{C}(x)]$$

- Bayes classifier (using the true $p_k(x)$) has the smallest error (in the population).
- Support vector machines build structured models for $\hat{C}(x)$.
- We will also build structured models for representing the $p_k(x)$ e.g Logistic regression ,generalized additive models.

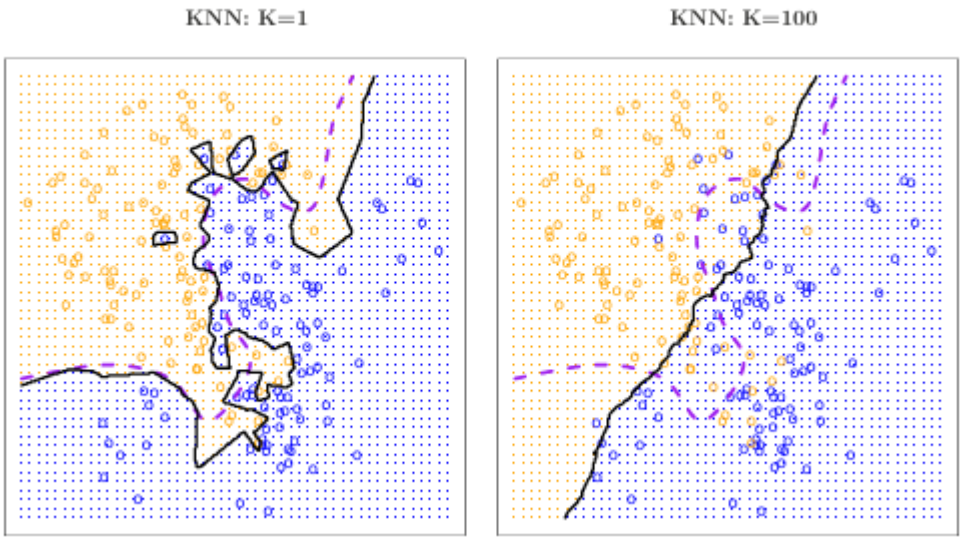


Above example has a simulated dataset, consisting of 100 observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.

We can use K- Nearest Neighbors , for the above example: on using $k = 10$, we get this KNN decision boundary represented by the black curve.



We can see that with small K , the boundary is overtly flexible but not so with large K



Shown below is the KNN training error rate(blue,200 observations) and test error rate (orange,5,000 observations) on the data, as the level of flexibility(assessed using $1/K$ on the logscale) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training dataset.