Announcement: Assignment 1 is out and due January the 26th.

Nearest Neighbor method:

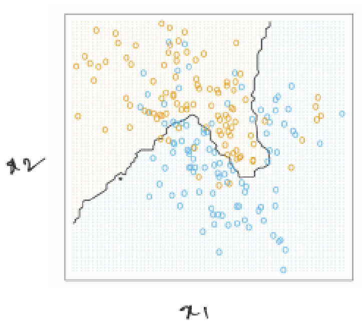
Use those predictions in the training set closest in input space to x to from $\hat{y}$

$$\hat{y} = \frac{1}{k} \Sigma_{x^{(i)} \in N_k(x)} y^{(i)}$$

$N_k(x)$-neighberhood of x, closet k points $x^{(i)}$. What metric ? Euclidean distance.

$$\hat{y}(x) = \begin{cases} 0 & if\ \hat{y} \leq 0.5 \\ 1 & if\ \hat{y} > 0.5 \end{cases}$$

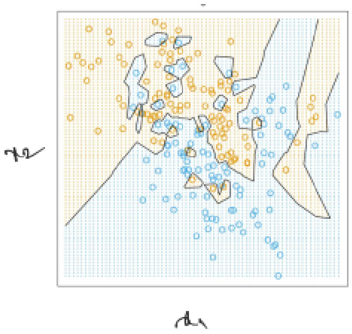
Assumption: This model assumes that the class distribution is locally smooth.



k = 15

decision boundary is for more irregular and responds to local clusters where one class dominates.

⇒ There are still some misclassifications.

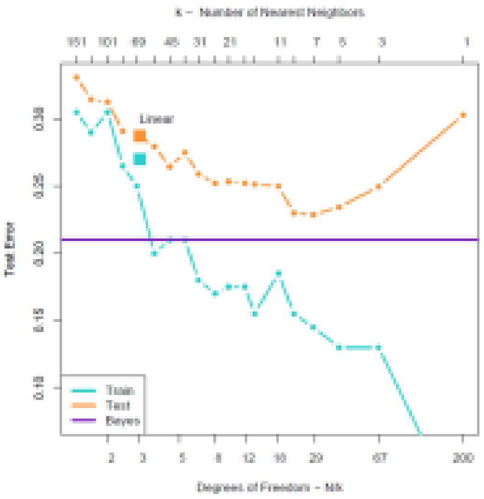


k = 1

The decision boundary is even more irregular than before!

⇒ There are no misclassification.

K is hyper parameter. For k = 1, overfits the training data. We have to choose k based on a validation test.



Q: Are there any other hyper-parameteres for K-NN ?
A: Metric to compute NN.

Least squares VS Nearest neighbor :

- Decision boundary is very smooth.	- Decision boundary is wiggly and depends on a handful of input points and their positions.
- More stable	- less stable.
- assumes that the decision boundary is linear. ↳ strong assumption.	- does not have any such strong assumptions about the decision boundary.
- high bias, low variance.	- low bias, high variance.

$x \in \mathbb{R}^p$: real valued random input vecor.
$y \in \mathbb{R}$: ral valued random input variable.
Pr(x,y): joint distribution of x,y. **Goal:** find a function f(x) for predicting y given values of x.
L(y,f(x)) = Loss function for penalizing errors in prediction.

$$L(y, f(x)) = (y - f(x))$$

$$EPE = \int\int (y - g(x))^2 Pr(x, y) dxdy$$

using Bayes: $f(x, y) = f(y \mid x)\, f(x)$

$$EPE = \int\int (y - g(x))^2 f(y \mid x)\, f(x) dxdy$$

$$EPE = \int f(x) \left(\int (y - g(x))^2 f(y \mid x) dy \right) dx$$

$$EPE = E_x(\int (y - g(x))^2 f(y \mid x) dy\ )$$

$$EPE = E_x E_{Y \mid X}(\, [Y - g(X)]^2 \mid X\,)$$

it sufficies to minimize EPE pointwise :

$$f^* = \underset{f}{\text{argmin}}\ E_{Y|X}\left[(Y-f)^2 | X=x\right]$$

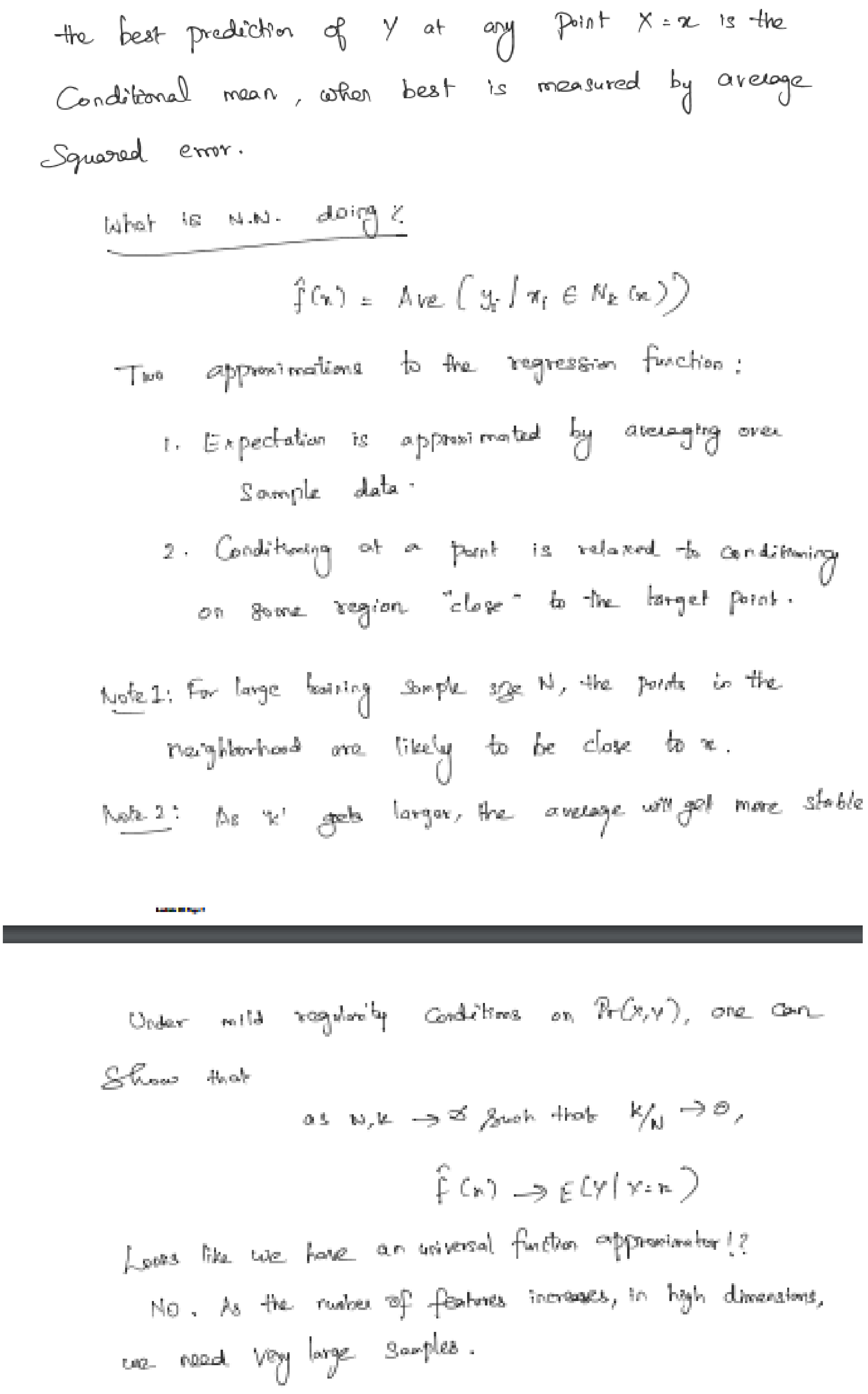
$$\frac{\partial}{\partial f} E_{Y|X}\left[(y-f)^2 \,|\, X=x\right] = 0$$

$$\frac{\partial}{\partial f}\ E_{Y|X}(y^2|X=x) + f^2 - 2\, E_{Y|X}(y|X=x) f = 0$$

$$2f - 2\, E_{Y|X}(y|X=x) = 0$$

$$\boxed{f^* = E_{Y|X}(Y|X=x)}$$

↑
regression function.



the best prediction of Y at any point $X = x$ is the Conditional mean, when best is measured by average Squared error.

What is N.N. doing?

$$\hat{f}(x) = \text{Ave}\left(y_i \mid x_i \in N_k(x)\right)$$

Two approximations to the regression function:

1. Expectation is approximated by averaging over Sample data.
2. Conditioning at a point is relaxed to conditioning on some region "close" to the target point.

Note 1: For large training sample size N, the points in the neighborhood are likely to be close to x.

Note 2: As 'k' gets larger, the average will get more stable

Under mild regularity Conditions on $Pr(X,Y)$, one can Show that

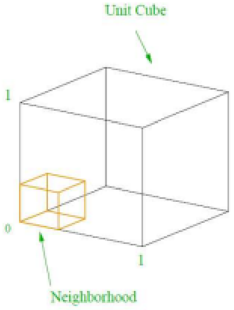
as $N, k \to \infty$ such that $k/N \to 0$,

$$\hat{f}(x) \to E(Y \mid X = x)$$

Looks like we have an universal function approximator!?

No. As the number of features increases, in high dimensions, we need very large samples.

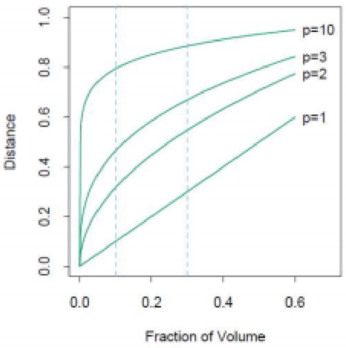
Local methods in high dimensions: Consider inputs uniformly distributed in a p-dimensional hypercube.



Consider a hypercubical neighborhood about a target point to capture a fraction 'r' of the observations.

↳ this corresponds to a fraction 'r' of the unit volume.

The expected edge length $e_p(r) = r^{\frac{1}{p}}$ $e_{10}(0.01) = 0.63, e_{0.1} = 0.83$.
To capture 1 or 10 percent of data, to from a local range, cover 63 or 83 of the range of the input variable. $\implies$ such neighberhood are no longer local. If you reduce "r", with fewer observations, variance will be high.



This is known as "Curse of Dimensionality" (Bellman, 1961).

Warning: Our intuitions breaks down in higher dimensions.

What is Linear regression doing ?

$$\hat{f}(x) = x^T w$$

This is a model based approach. We are specifying a model for the regression fn.

$$w = [E(xx^T)]^{-1} E(XY)$$

Assumptions:

1. Least squares –¿ approximated by a gloabally linear fn.
2. K-NN –¿ assumes f(n) is well approximated by a locally constant function.
   These assumptions $\implies$ inductive bias of the algorithm

Additive models:

$$f(x) = \Sigma_{j=1}^{p} f_j(x_j)$$

This retains the additivity of the linear model but each coordinate function f_j is arbitrary. Q: What happens if we replace L_2 loss with L_1 ?
A: $\hat{f}(x) = median(y|X = x)$

Classification: Output is categorical variable G.
some paradigms work here, we need a different loss function for penelazing prediction error.
G: set of possible classes.
$\hat{G}$: prediction.

loss fn: $k \times k$ matrix where $k = Card(G)$

$$\begin{array}{c|ccccc} & 1 & 2 & \cdots & & k \\ 1 & 0 & & & & - \\ 2 & & 0 & & & \\ \vdots & & & 0 & 0 & \\ k & & & & & 0 \end{array}$$

Price paid for classifying an observation belonging to class 1 to class k.

$L(k,l)$ - price for classifying an observation belonging to class k to class l.

$$EPE = E[L(G, \hat{G}(x))]$$

Expectation is wit respect to Pr(G,x).

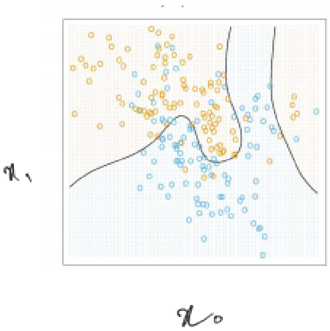
$$= E_x \Sigma_{k=1}^{K} L(G, \hat{G}(x)) Pr(G, x)$$

Again, it suffices to minimize EPE pointwise:

$$\hat{G}(x) = argmin_{g \in G} \Sigma_{k=1}^{K} L(G_k, g) Pr(G_k, |X = x)$$

For 0-1 loss:

$$\hat{G}(x) = argmin_{g \in G}(1 - Pr(g|X = x))$$
$$= argmax_{g \in G} Pr(g|X = x)$$



The optimal Bayes decision boundary.

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

K-NN directly approximate this solution
Majority voting in neighberhood.

1. conditional probability at a point is related to conditional probability within a neigherhoof of a point.
2. Probabilities are estimated by training sample proportions.