# Curse of Dimensionality and k-NN

Mohammad Emtiyaz Khan
EPFL

Oct 15, 2015

# Classification example

In many cases, a linear model may not be optimal. There are three confounding factors: perhaps our model is too rigid (bias) ~~orsecond~~, or perhaps it is too flexible (variance), or perhaps some errors are just unavoidable (the noise).


**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*.*

*All figures taken from Chapter 2 HTF

# $k$-Nearest Neighbor (k-NN)

The $k$-NN prediction for an $\mathbf{x}_*$ is,

$$f_k(\mathbf{x}_*) = \frac{1}{k} \sum_{\mathbf{x}_n \in nbh_k(\mathbf{x}_*)} y_n \ ,$$

where $nbh_k(\mathbf{x})$ is the neighborhood of $\mathbf{x}$ defined by the $k$ closest points $\mathbf{x}_n$ in the training data.

We show results for $k = 1$ and $k = 15$ respectively.

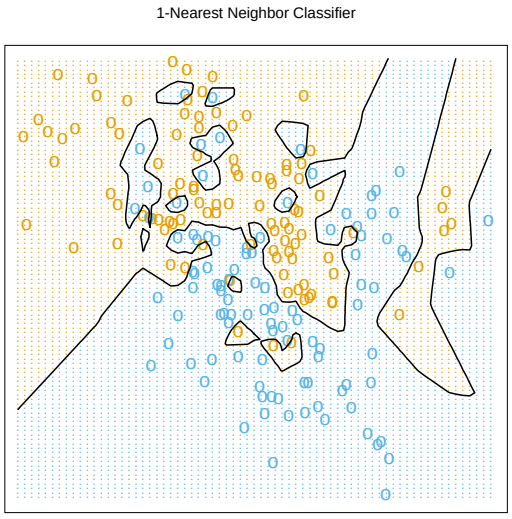
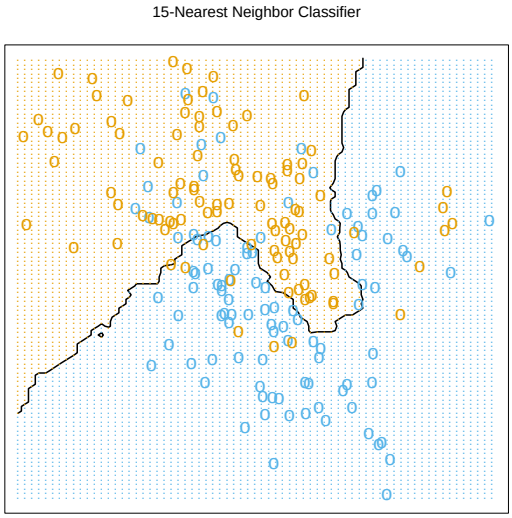



**FIGURE 2.3.** *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by* 1*-nearest-neighbor classification.*




A note on implementation:

Training involves choosing k.
During testing, we simply find the nearest k training pts and classify.

# Bias-variance revisited

How should train and test error vary with $k$?

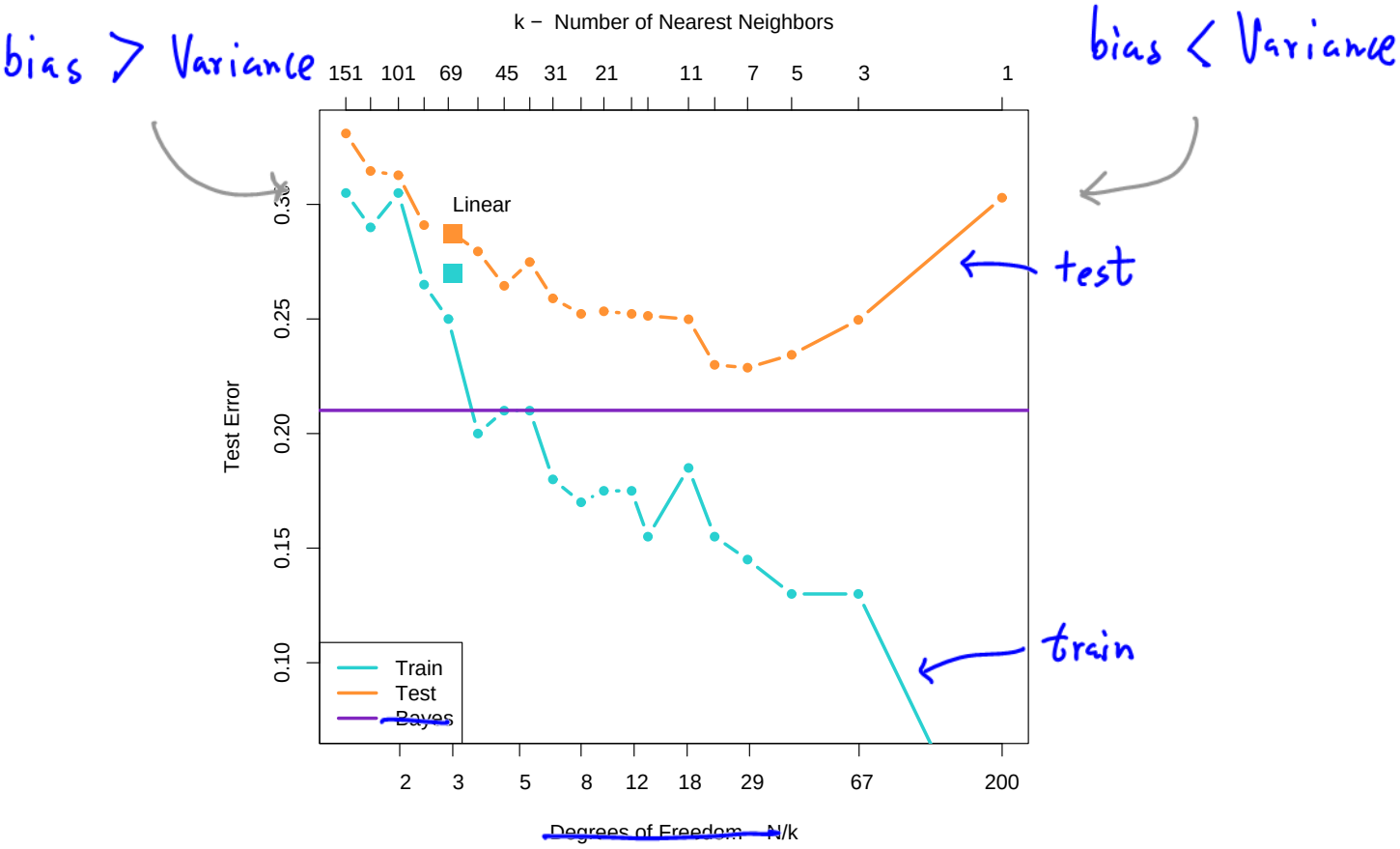
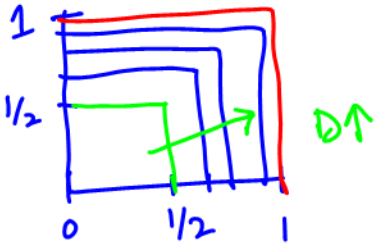



**FIGURE 2.4.** *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* $10,000$*. The orange curves are test and the blue are training error for k-nearest-neighbor classification. The results for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

# Curse of dimensionality

You might want to read "A few useful things to know about ML"

According to Pedro Domingos: "Intuitions fail in high dimensions". This is also known as the curse of dimensionality (Bellman, 1961).




$r = 1/2$

$r^2 = 1/2 \Rightarrow r = 1/\sqrt{2} \approx 0.7$

$r^D = 1/2 \Rightarrow r = 1/2^{1/D}$

**Claim 1:** "Generalizing correctly becomes exponentially harder as the dimensionality grows because fixed-size training sets cover a dwindling fraction of the input space."

The expected edge length is $e_D(r) = r^{1/D}$, e.g.

$$e_{10}(0.01) = 0.63\,,\; e_{10}(0.1) = 0.80$$

i.e. to capture 1% or 10% of the data, we must cover 63% or 80% of the range of each input variable.

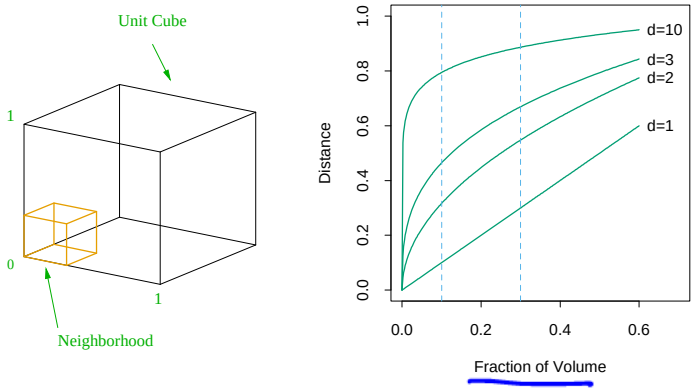



**FIGURE 2.6.** *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction $r$ of the volume of the data, for different dimensions $p$. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

As a result, the sampling density is proportional to $N^{1/D}$, i.e. if $N_1 = 100$ is the sample size for a 1-input problem, then $N_{10} = 100^{10}$ is required for the same sampling density with 10 inputs. (Worst case)

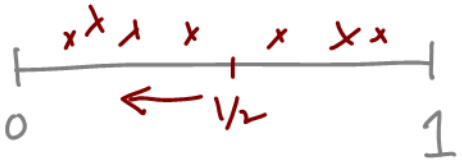
$r^{1/D} \Rightarrow N^{1/D}$

**Claim 2:** In high-dimension, data-points are far from each other. Consequently, "as the dimensionality increases, the choice of nearest neighbor becomes effectively random."

Consider $N$ data points uniformly distributed in a $D$-dimensional unit ball centered at the origin. We consider a nearest-neighbor estimate at the origin. The median distance from the origin to the closest data point is,

$$\left(1 - \tfrac{1}{2}^{1/N}\right)^{1/D}$$ ✓

For $N = 500, D = 10$, this number is 0.52, more than halfway to the boundary.




$D=1$, $\left(1-\frac{1}{2}\right)$

| D \ N | 1 | 10 | 100 |
|---|---|---|---|
| 1 | 1/2 | 0.07 | 0.01 |
| 10 | $(1/2)^{1/10}$ | $(0.07)^{1/10}$ | $(0.01)^{1/10}$ |
| | | ≃0.76 | ≃0.63 |

(see HTF for details)
Chapter 2

# k-NN vs linear revisited

In high-dimension, both bias and variance increase.

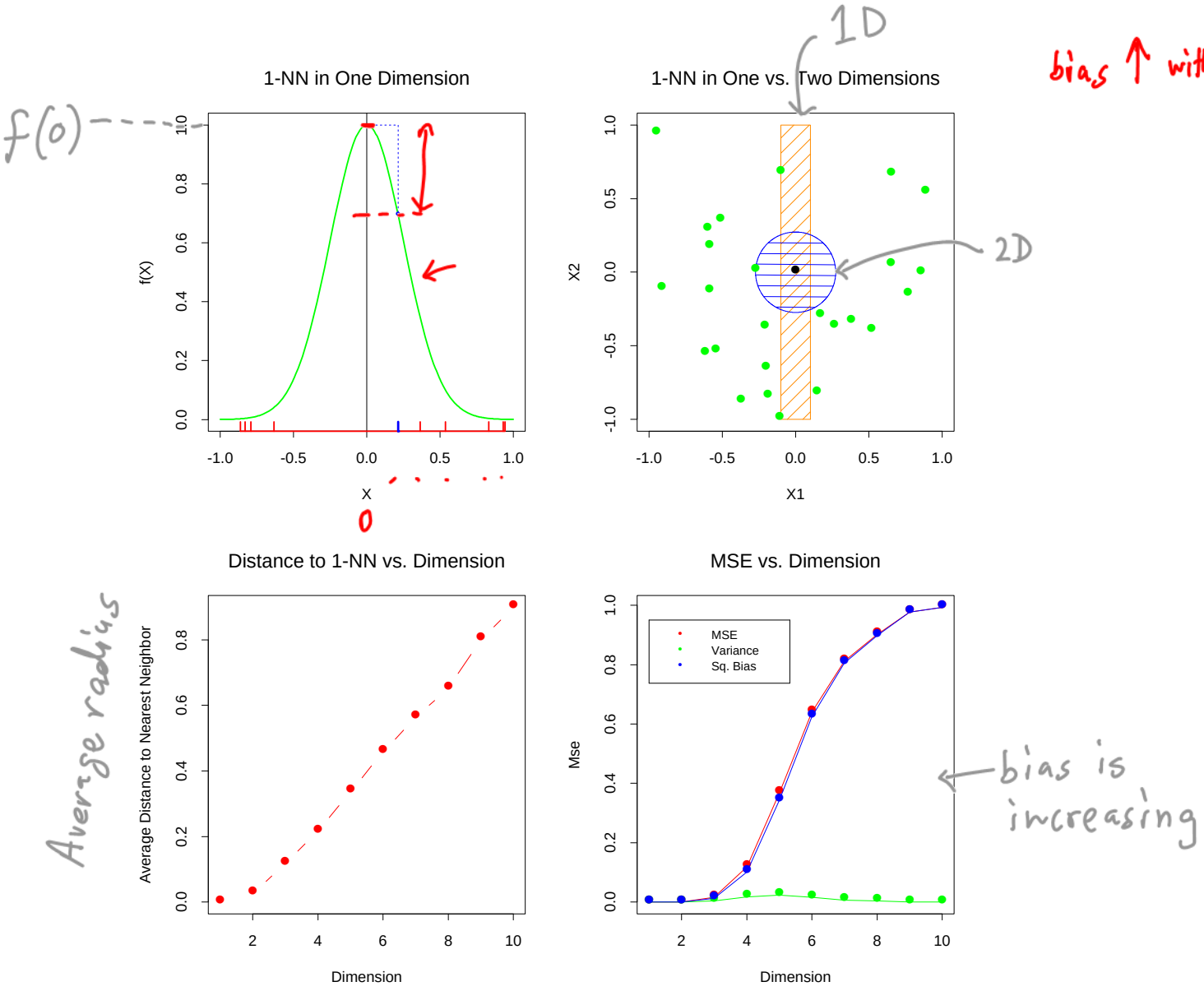
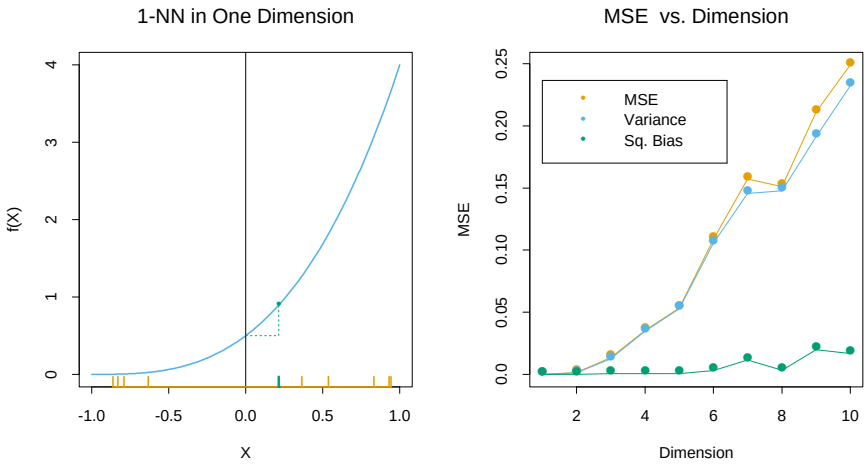



**FIGURE 2.7.** *A simulation example, demonstrating the curse of dimensionality and its effect on MSE, bias and variance. The input features are uniformly distributed in* $[-1, 1]^p$ *for* $p = 1, \ldots, 10$ *The top left panel shows the target function (no noise) in* $\mathbb{R}$*:* $f(X) = e^{-8||X||^2}$*, and demonstrates the error that 1-nearest neighbor makes in estimating* $f(0)$*.*

Another example for high variance.




If the true function only involve a few dimension then the variance can dominate too.

**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.*

Recall that the variance of linear regression grows only linearly with dimensionality (~~Page 5~~ in the "bias-variance" lecture). By imposing some heavy restrictions on the class of models, we can avoid the curse of dimensionality or the curse of highly-variable functions.

the 4th key result

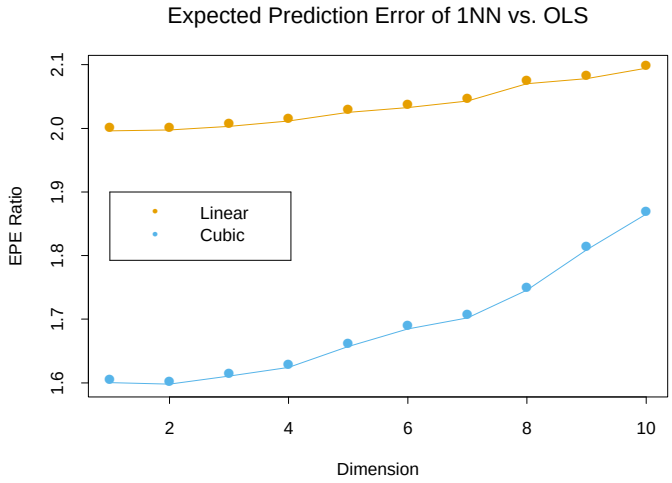



**FIGURE 2.9.** *The curves show the expected prediction error (at* $x_0 = 0$*) for* 1*-nearest neighbor relative to least squares for the model* $Y = f(X) + \varepsilon$. *For the orange curve,* $f(x) = x_1$, *while for the blue curve* $f(x) = \frac{1}{2}(x_1 + 1)^3$.

## Discussion

(Taken from HTF). We will see (in the next few lectures) that there is a whole spectrum of models between the rigid linear models and the extremely flexible 1-NN model. Each model comes with their own assumptions and biases.

(Based on Domingos). You might think that gathering more input variables never hurts, since at the worst they provide no new information about the output. But in fact their benefits may be outweighed by the curse of dimensionality.