

Lecture 4: K-means and K-nearest neighbors

Reading: Sections 13.3, 14.3.6

GU4241/GR5241 Statistical Machine Learning

Linxi Liu

January 26, 2017

Clustering

We assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering

We assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering is an **unsupervised** procedure, whose goal is to find homogeneous subgroups among the observations. It has wide applications in practice. **Image segmentation, handwritten digit identification, vector quantization**

Clustering

We assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering is an **unsupervised** procedure, whose goal is to find homogeneous subgroups among the observations. It has wide applications in practice. **Image segmentation, handwritten digit identification, vector quantization**

We will discuss 4 algorithms in this semester:

- ▶ EM algorithm
- ▶ K -means clustering

Clustering

We assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering is an **unsupervised** procedure, whose goal is to find homogeneous subgroups among the observations. It has wide applications in practice. **Image segmentation, handwritten digit identification, vector quantization**

We will discuss 4 algorithms in this semester:

- ▶ EM algorithm
 - ▶ ***K*-means clustering**
 - ▶ ***K*-medoids clustering**
- EM is more general, K means and K medoids are very similar

Clustering

We assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering is an **unsupervised** procedure, whose goal is to find homogeneous subgroups among the observations. It has wide applications in practice. **Image segmentation, handwritten digit identification, vector quantization**

We will discuss 4 algorithms in this semester:

- ▶ EM algorithm
- ▶ K -means clustering
- ▶ K -medoids clustering
- ▶ Hierarchical clustering

Handwritten digit identification



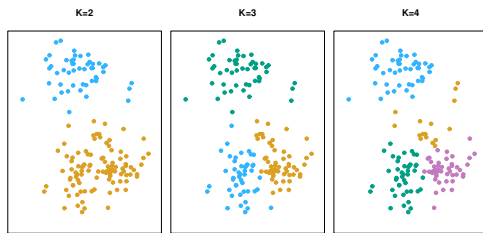
FIGURE 11.9. *Examples of training cases from ZIP code data. Each image is a 16×16 8-bit grayscale representation of a handwritten digit.*

Image segmentation



K -means clustering

- K is the number of clusters and must be fixed in advance.

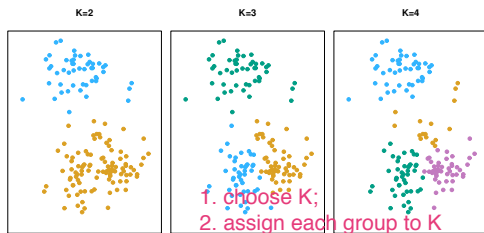


ISL Figure 10.5

we want to minimize the within parawised distance

K -means clustering

- K is the number of clusters and must be fixed in advance.



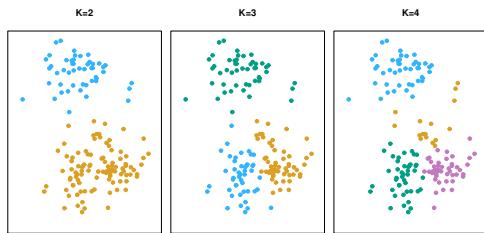
ISL Figure 10.5

- The goal of this method is to maximize the similarity of samples within each cluster:

$$\min_C W(C)$$

K -means clustering

- K is the number of clusters and must be fixed in advance.



ISL Figure 10.5

- The goal of this method is to maximize the similarity of samples within each cluster:

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

K -means clustering algorithm

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.

K -means clustering algorithm

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:

K -means clustering algorithm

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:
 - Find the *centroid* of each cluster ℓ ; i.e. the average $\bar{x}_{\ell,:}$ of all the samples in the cluster:

$$\bar{x}_{\ell,j} = \frac{1}{|\{i : C(i) = \ell\}|} \sum_{i:C(i)=\ell} x_{i,j} \quad \text{for } j = 1, \dots, p.$$

K -means clustering algorithm

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:
 - ▶ Find the *centroid* of each cluster ℓ ; i.e. the average $\bar{x}_{\ell,:}$ of all the samples in the cluster:

$$\bar{x}_{\ell,j} = \frac{1}{|\{i : C(i) = \ell\}|} \sum_{i:C(i)=\ell} x_{i,j} \quad \text{for } j = 1, \dots, p.$$

- ▶ Reassign each sample to the nearest centroid.

K -means clustering algorithm

Elements of Statistical Learning (2nd Ed.) ©Hastie, Tibshirani & Friedman 2009 Chap 14

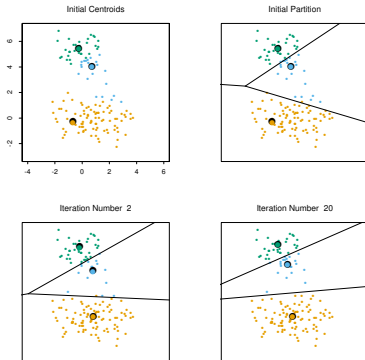


FIGURE 14.6. Successive iterations of the K -means clustering algorithm for the simulated data of Figure 14.4.

Properties of K -means clustering

- The algorithm always converges to a local minimum of

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

this is the minimized within the clustering pairwise distance

Properties of K -means clustering

- The algorithm always converges to a local minimum of

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

Why? When d is the Euclidean distance

we use the mean to calculate the ED

$$\frac{1}{2} \sum_{C(i)=\ell} \sum_{C(j)=\ell} d(x_i, x_j) = |N_\ell| \sum_{C(i)=\ell} d(x_i, \bar{x}_\ell)$$

Properties of K -means clustering

- The algorithm always converges to a local minimum of

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

Why? When d is the Euclidean distance

$$\frac{1}{2} \sum_{C(i)=\ell} \sum_{C(j)=\ell} d(x_i, x_j) = |N_\ell| \sum_{C(i)=\ell} d(x_i, \bar{x}_\ell)$$

This side can only be reduced in each iteration.

Properties of K -means clustering

- The algorithm always converges to a local minimum of

$$\min_C W(C) \quad ; \quad W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j).$$

Why? When d is the Euclidean distance

$$\frac{1}{2} \sum_{C(i)=\ell} \sum_{C(j)=\ell} d(x_i, x_j) = |N_\ell| \sum_{C(i)=\ell} d(x_i, \bar{x}_\ell)$$

This side can only be reduced in each iteration.

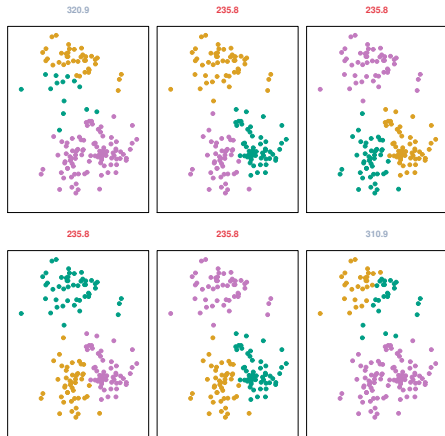
- Each initialization could yield a different minimum.

Example: K -means output with different initializations



ISL Figure 10.7

Example: K -means output with different initializations



In practice, we start from many random initializations and choose the output which **minimizes** the objective function.

ISL Figure 10.7

Practical Issues

- Categorical features are usually coded as dummy variables:

$$X = 1, 2, \text{ or } 3 \quad \rightarrow \quad \begin{matrix} (1 \ 0 \ 0) \\ (0 \ 1 \ 0) \\ \text{or } (0 \ 0 \ 1) \end{matrix}$$

Practical Issues

- ▶ Categorical features are usually coded as dummy variables:

$$X = 1, 2, \text{ or } 3 \quad \rightarrow \quad \begin{matrix} (1 \ 0 \ 0) \\ (0 \ 1 \ 0) \\ \text{or } (0 \ 0 \ 1) \end{matrix}$$

- ▶ Weighting is also possible

Practical Issues

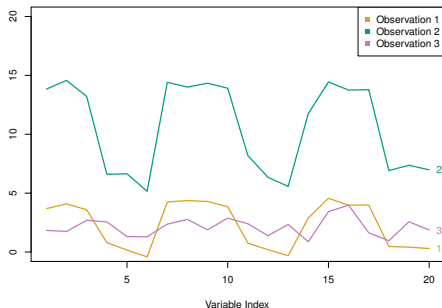
- ▶ Categorical features are usually coded as dummy variables:

$$X = 1, 2, \text{ or } 3 \quad \rightarrow \quad \begin{matrix} (1 \ 0 \ 0) \\ (0 \ 1 \ 0) \\ \text{or } (0 \ 0 \ 1) \end{matrix}$$

- ▶ Weighting is also possible
- ▶ How to choose the number of clusters K ?

Correlation distance

- Euclidean distance would cluster all customers who purchase few things (orange and purple).



Correlation distance

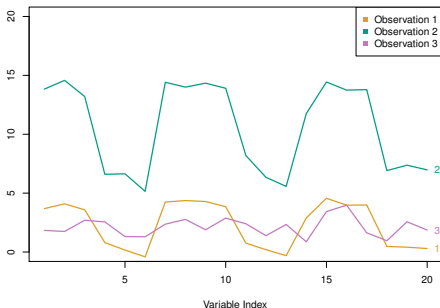
- ▶ Euclidean distance would cluster all customers who purchase few things (orange and purple).
- ▶ Perhaps we want to cluster customers who purchase *similar* things (orange and teal).

similar the scaling step, we need to normalize the data.

Otherwise, the result is informative. Thus use ED method, based on the quantity

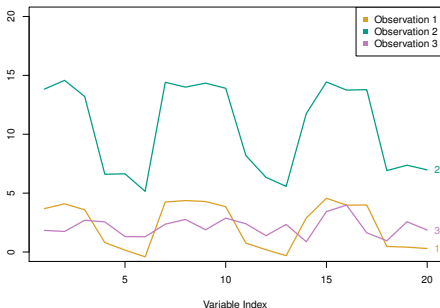
If we use correlation method, the result is based on the preference.

Thus use correlation method == scaling step.



Correlation distance

- ▶ Euclidean distance would cluster all customers who purchase few things (orange and purple).
- ▶ Perhaps we want to cluster customers who purchase *similar* things (orange and teal).
- ▶ Then, the **correlation distance** may be a more appropriate measure of dissimilarity between samples.



Fact of correlation distance

Correlation is defined by

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}},$$

where \bar{x}_i = mean of observation i .

Fact of correlation distance

Correlation is defined by

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}},$$

where \bar{x}_i = mean of observation i .

If observations are standardized:

$$x_{ij} \leftarrow \frac{x_{ij} - \bar{x}_i}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2}},$$

then $2(1 - \rho(x_i, x_{i'})) = \sum_j (x_{ij} - x_{i'j})^2$.

K -medoids clustering

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.

in K means, we need to find the center of the each cluster,

center can be just mean or other thing;

in the K medoids, we need to set the center as the particular observation

K -medoids clustering

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random. assign each cluster a label
2. Iterate these two steps until the clustering is constant:

K -medoids clustering

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:
 - For a given cluster assignment C find the **observation** in the cluster minimizing total pairwise distance with the other cluster members:

$$i_k^* = \underset{\{i: C(i)=k\}}{\operatorname{argmin}} \sum_{C(i')=k} d(x_i, x_{i'}).$$

Then $z_k = x_{i_k^*}$, $k = 1, 2, \dots, K$ are the current estimates of the cluster centers.

the centroid can give us the summary of the data set,
we want the centroid to be meaningful, and particular.
Thus the Centroids is an observation

K -medoids clustering

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.
2. Iterate these two steps until the clustering is constant:
 - ▶ For a given cluster assignment C find the **observation** in the cluster minimizing total pairwise distance with the other cluster members:

$$i_k^* = \operatorname{argmin}_{\{i: C(i)=k\}} \sum_{C(i')=k} d(x_i, x_{i'}).$$

Then $z_k = x_{i_k^*}$, $k = 1, 2, \dots, K$ are the current estimates of the cluster centers.

- ▶ Given a current set of cluster centers $\{z_1, \dots, z_K\}$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \operatorname{argmin}_{1 \leq k \leq K} d(x_i, z_k).$$

K -medoids clustering

- ▶ Same as K -means, except that centroid is required to be one of the observations.

K -medoids clustering

- ▶ Same as K -means, except that centroid is required to be one of the observations.
- ▶ Advantage: centroid is one of the observations— useful, for example when features are 0 or 1. Also, one only needs pairwise distances for K -medoids rather than the raw observations.

K-nearest neighbors regression

KNN regression: prototypical nonparametric method.

Given a training set (\mathbf{X}, \mathbf{y}) :

K -nearest neighbors regression

KNN regression: prototypical nonparametric method.

Given a training set (\mathbf{X}, \mathbf{y}) :

$$\hat{f}(x) = \frac{1}{K} \sum_{i \in N_K(x)} y_i$$

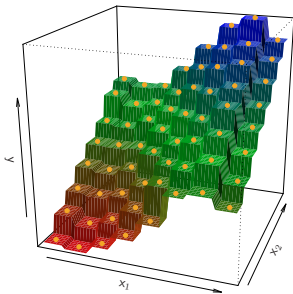
K -nearest neighbors regression

KNN regression: prototypical nonparametric method.

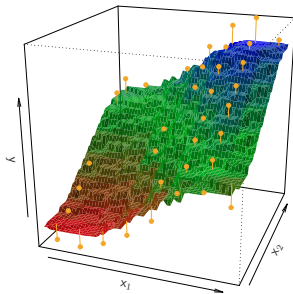
Given a training set (\mathbf{X}, \mathbf{y}) :

n/k is the number of the parameters.

$$\hat{f}(x) = \frac{1}{K} \sum_{i \in N_K(x)} y_i$$



$K = 1$



$K = 9$

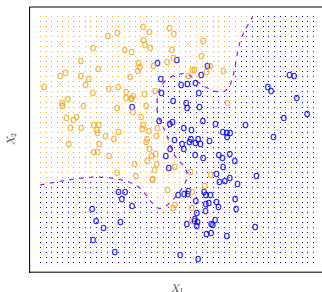
$1/k$ gives the flexibility, less $1/k$, then the model is smoother

Classification problem

Recall:

- ▶ $X = (X_1, X_2)$ are inputs.
- ▶ Color $Y \in \{\text{Yellow}, \text{Blue}\}$ is the output.

we will only use bayes classifier , for it is the optimal one under zero one laus function

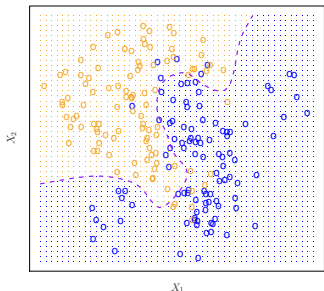


ISL Figure 2.13

Classification problem

Recall:

- ▶ $X = (X_1, X_2)$ are inputs.
- ▶ Color $Y \in \{\text{Yellow}, \text{Blue}\}$ is the output.
- ▶ (X, Y) have a joint distribution.

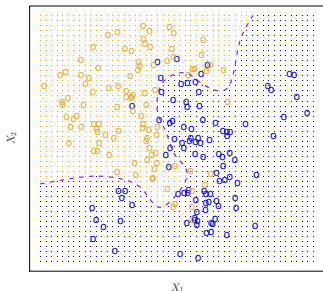


ISL Figure 2.13

Classification problem

Recall:

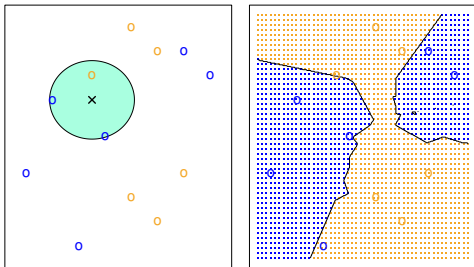
- ▶ $X = (X_1, X_2)$ are inputs.
- ▶ Color $Y \in \{\text{Yellow}, \text{Blue}\}$ is the output.
- ▶ (X, Y) have a joint distribution.
- ▶ Purple line is *Bayes boundary* — the best we could do if we knew the joint distribution of (X, Y)



ISL Figure 2.13

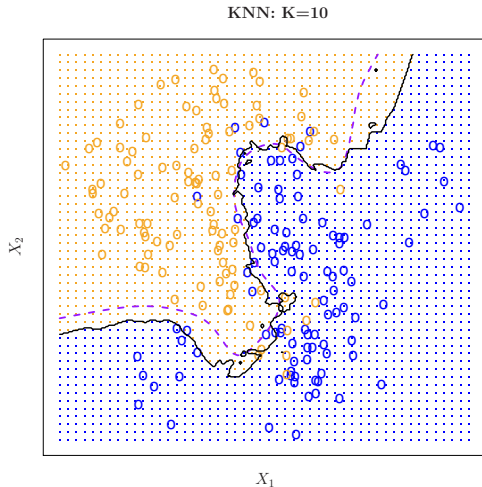
K -nearest neighbors

To assign a color to the input \times , we look at its $K = 3$ nearest neighbors. We predict the color of the majority of the neighbors.



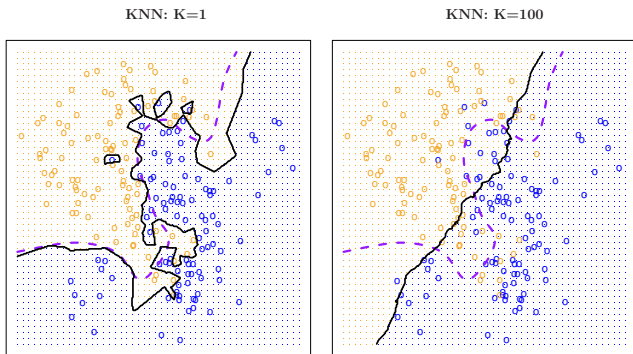
ISL Figure 2.14

K -nearest neighbors also has a decision boundary



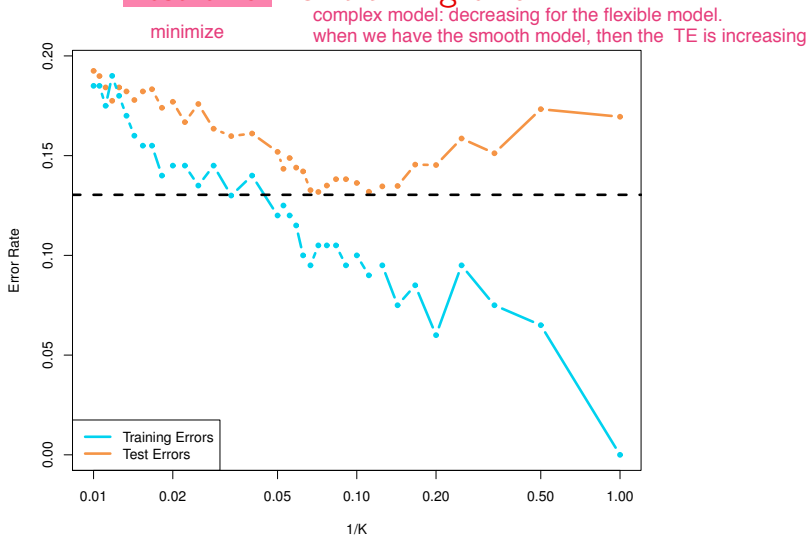
ISL Figure 2.15

The higher K , the smoother the decision boundary



ISL Figure 2.16

Test error vs. training error



ISL Figure 2.17

Curse of dimensionality

K-nearest neighbors can fail in high dimensions, because it becomes difficult to gather K observations close to a target point x_0 :

Curse of dimensionality

K-nearest neighbors can fail in high dimensions, because it becomes difficult to gather K observations close to a target point x_0 :

- ▶ near neighborhoods tend to be spatially large, the estimates are biased.

Curse of dimensionality

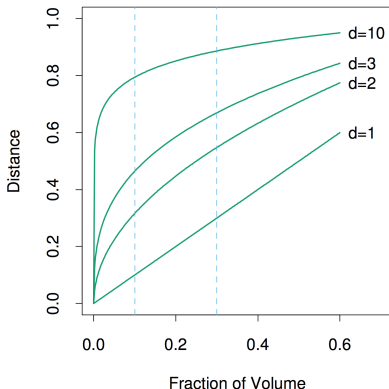
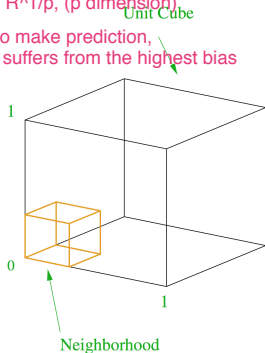
K-nearest neighbors can fail in high dimensions, because it becomes difficult to gather K observations close to a target point x_0 :

- ▶ near neighborhoods tend to be spatially large, the estimates are biased.
- ▶ reducing the spatial size of the neighborhood means reducing K , and the variance of the estimate increases.

In order to:
 point is close enough
 cover the fixed fraction of data
 then we want the volume to be R ,
 and edge is $R^{1/p}$, (p dimension)

Curse of dimensionality

if we want to make prediction,
 we need to suffers from the highest bias



ESL Figure 2.6

- ▶ We want to obtain a hypercubic neighborhood about a target point to capture a fraction r of the observations.
- ▶ The expected edge length will be $e_p(r) = r^{1/p}$. In ten dimensions, $e_{10}(0.01) = 63\%$.

Example

- ▶ 1000 training examples x_i generated uniformly on $[-1, 1]^p$.

Example

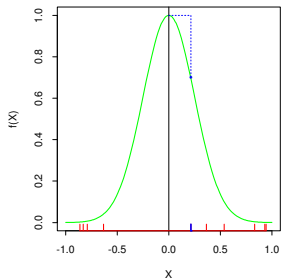
- ▶ 1000 training examples x_i generated uniformly on $[-1, 1]^p$.
- ▶ $Y = f(X) = e^{-8\|X\|^2}$ (no measurement error).

Example

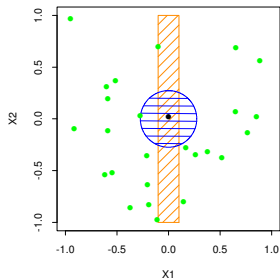
- ▶ 1000 training examples x_i generated uniformly on $[-1, 1]^p$.
- ▶ $Y = f(X) = e^{-8\|X\|^2}$ (no measurement error).
- ▶ use the 1-nearest-neighbor rule to predict y_0 at the test-point $x_0 = 0$.

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

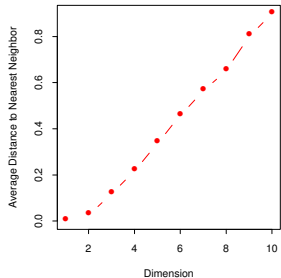
1-NN in One Dimension



1-NN in One vs. Two Dimensions



Distance to 1-NN vs. Dimension



MSE vs. Dimension

