Nearest Neighbor Classifiers and the Curse of Dimensionality

Machine Learning Course - CS-433
Oct 26, 2022
Nicolas Flammarion

EPFL

Supervised machine learning

We observe some data $S_{\text{train}} = \{x_n, y_n\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$

Goal: given a new observation x, we want to predict its label y

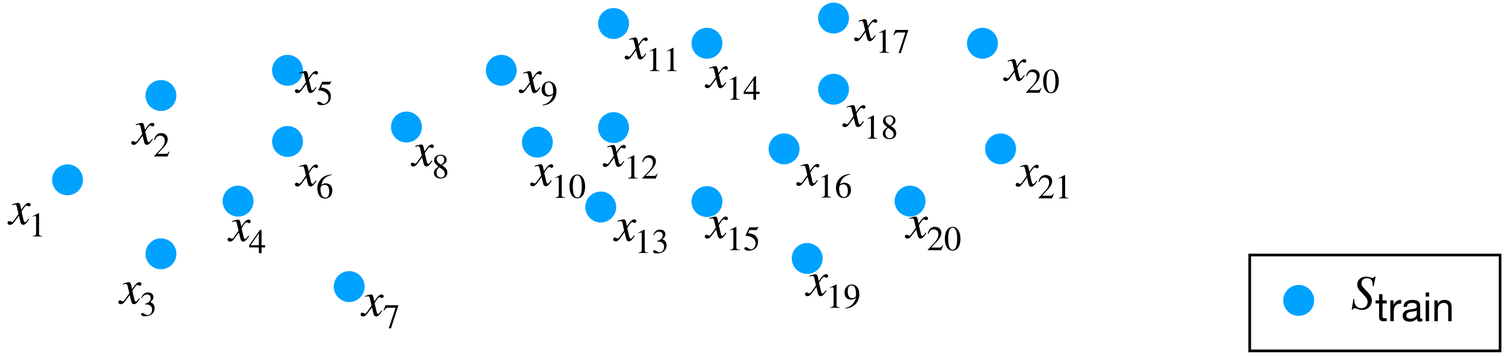
How:

$S_{\text{train}} = \{x_n, y_n\}_{n=1}^{N}$ $\quad$ $\mathcal{A}$ $\quad$ $f_S = \mathcal{A}(S)$

Nearest neighbor function

$\text{nbh}_{S_{train},k}: \mathcal{X} \to \mathcal{X}^k$

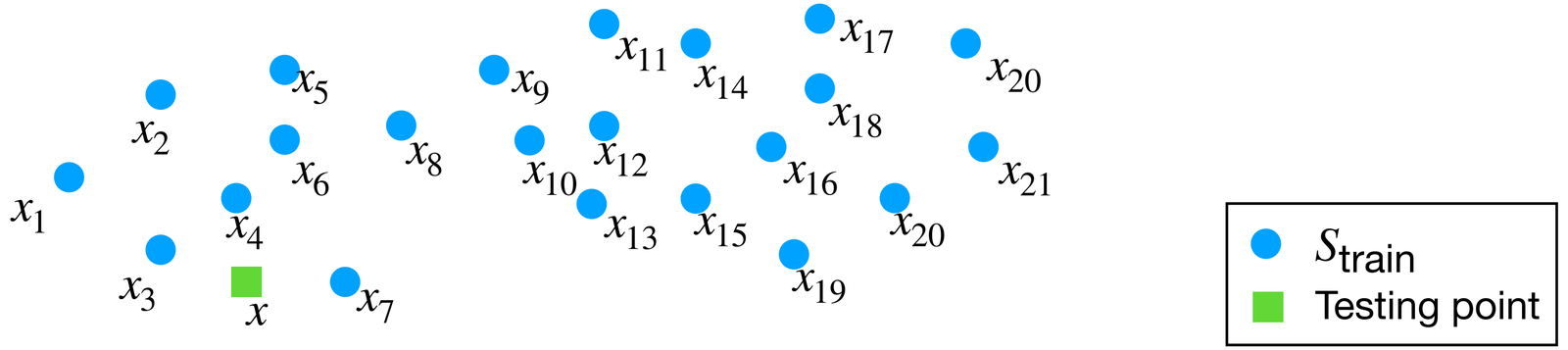
$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$



Nearest neighbor function

$\text{nbh}_{S_{train},k}: \mathcal{X} \to \mathcal{X}^k$

$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$

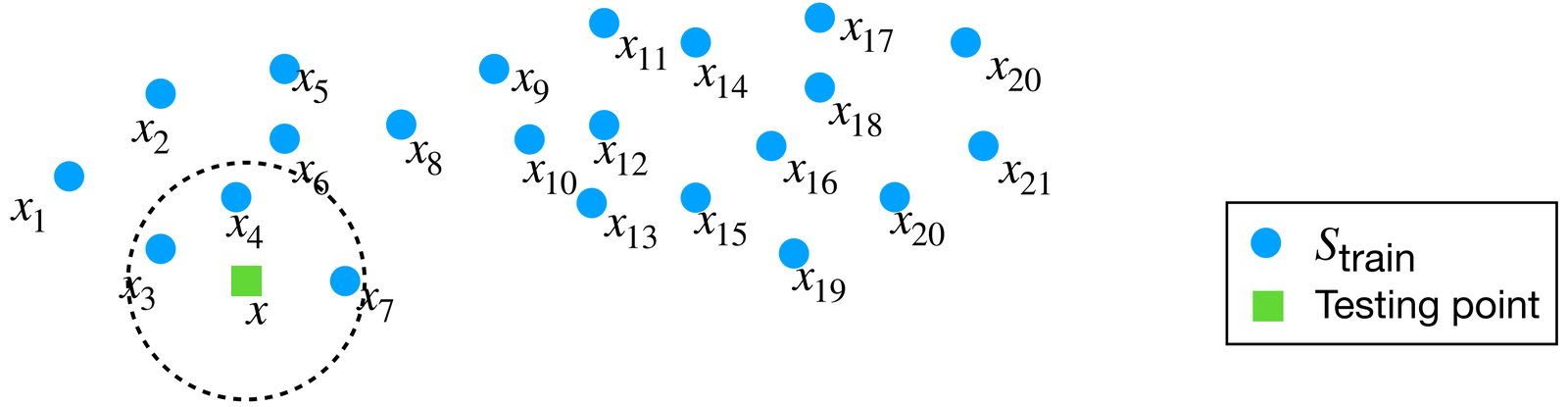


$\text{nbh}_{S_{train},3}(x) = ?$

Nearest neighbor function

$$\text{nbh}_{S_{train},k}: \mathcal{X} \to \mathcal{X}^k$$

$$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$$

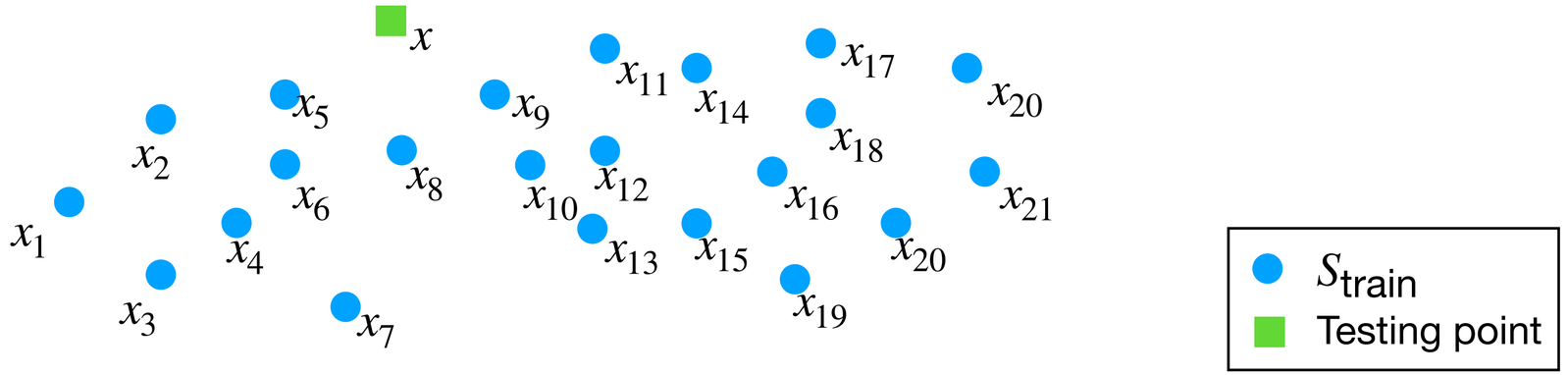


$$\text{nbh}_{S_{train},3}(x) = \{x_3, x_4, x_7\}$$

Nearest neighbor function

$\text{nbh}_{S_{train},k}: \mathcal{X} \to \mathcal{X}^k$

$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$

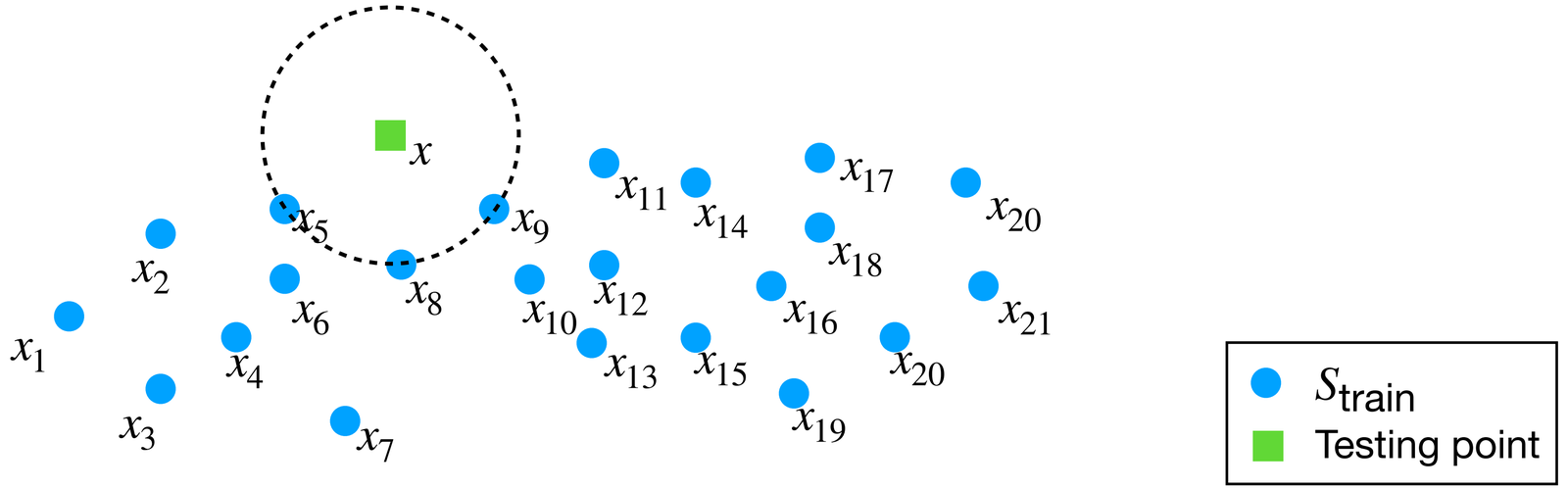


$\text{nbh}_{S_{train},2}(x) = ?$

Nearest neighbor function

$\text{nbh}_{S_{train},k} : \mathcal{X} \to \mathcal{X}^k$

$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$

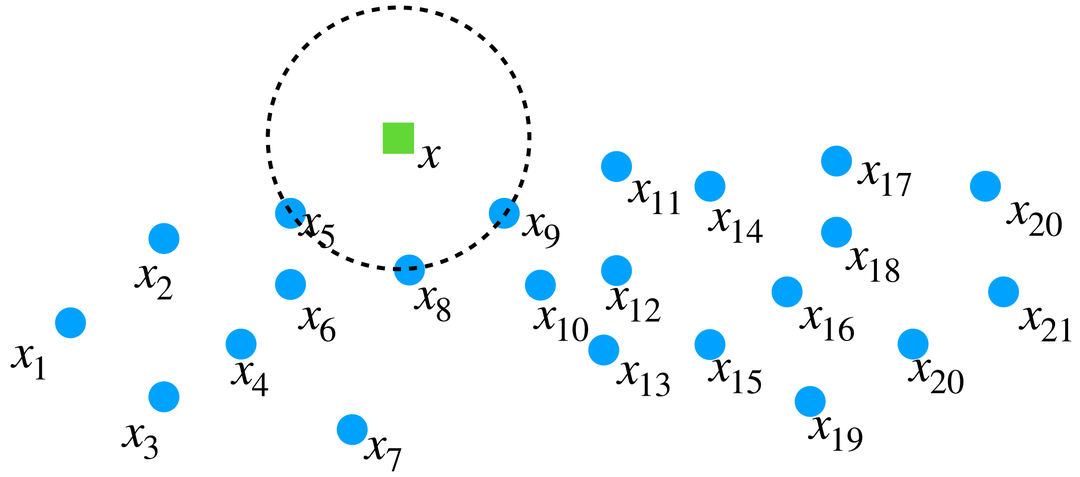


$\text{nbh}_{S_{train},2}(x) = ?$

Nearest neighbor function

$$\text{nbh}_{S_{train},k}: \mathcal{X} \to \mathcal{X}^k$$

$$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$$



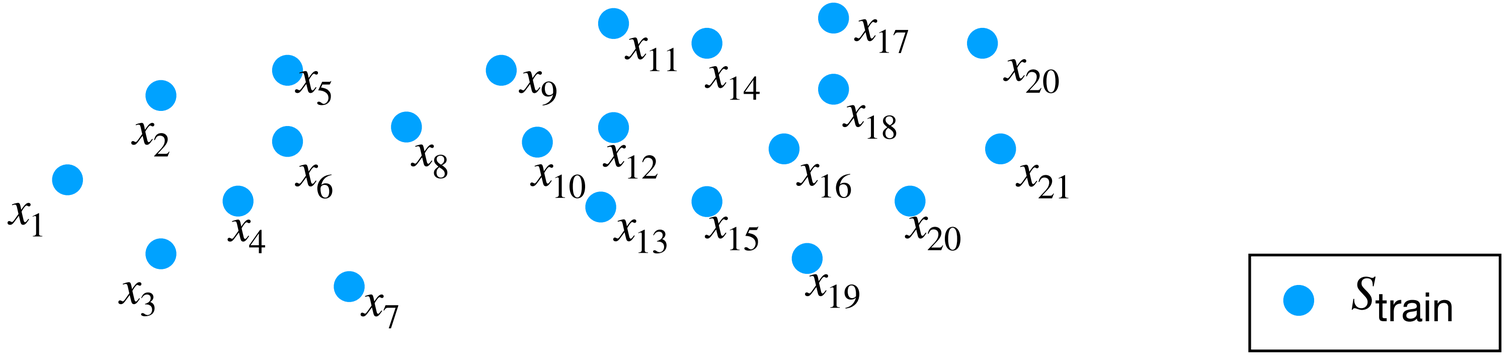
$$\text{nbh}_{S_{train},2}(x) = \{x_5, x_8\}$$

Not uniquely defined!
It will depend on the implementation
Often ties are broken randomly

Nearest neighbor function

$$\text{nbh}_{S_{train},k}: \mathcal{X} \to \mathcal{X}^k$$

$$x \mapsto \{k \text{ elements of } S_{\text{train}} \text{ the closest to } x\}$$

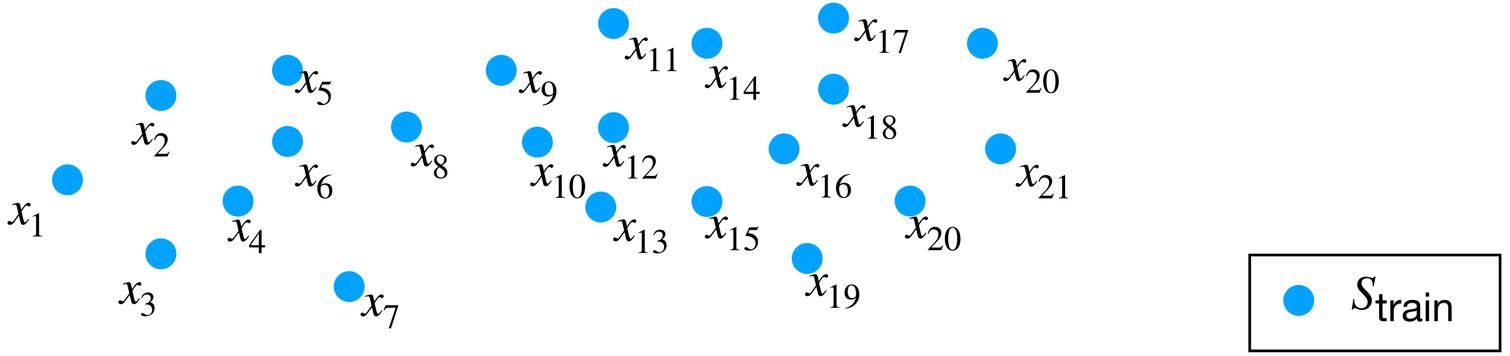


Remarks:

- Different metrics can be used
- Computational complexity when N is large (but efficient data structure may exist)

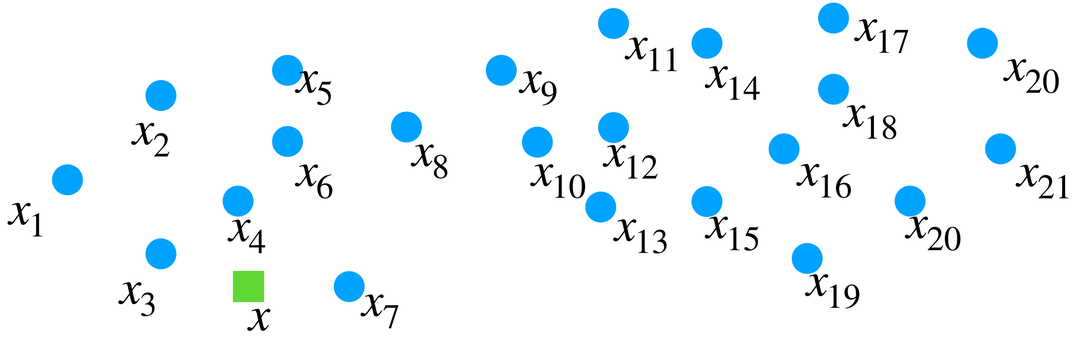
k-NN can be used for regression ($y \in \mathbb{R}$)

$$f_{S_{train},k}(x) = \frac{1}{k} \sum_{n:x_n \in nbh_{S_{train},k}(x)} y_n$$



k-NN can be used for regression ($y \in \mathbb{R}$)

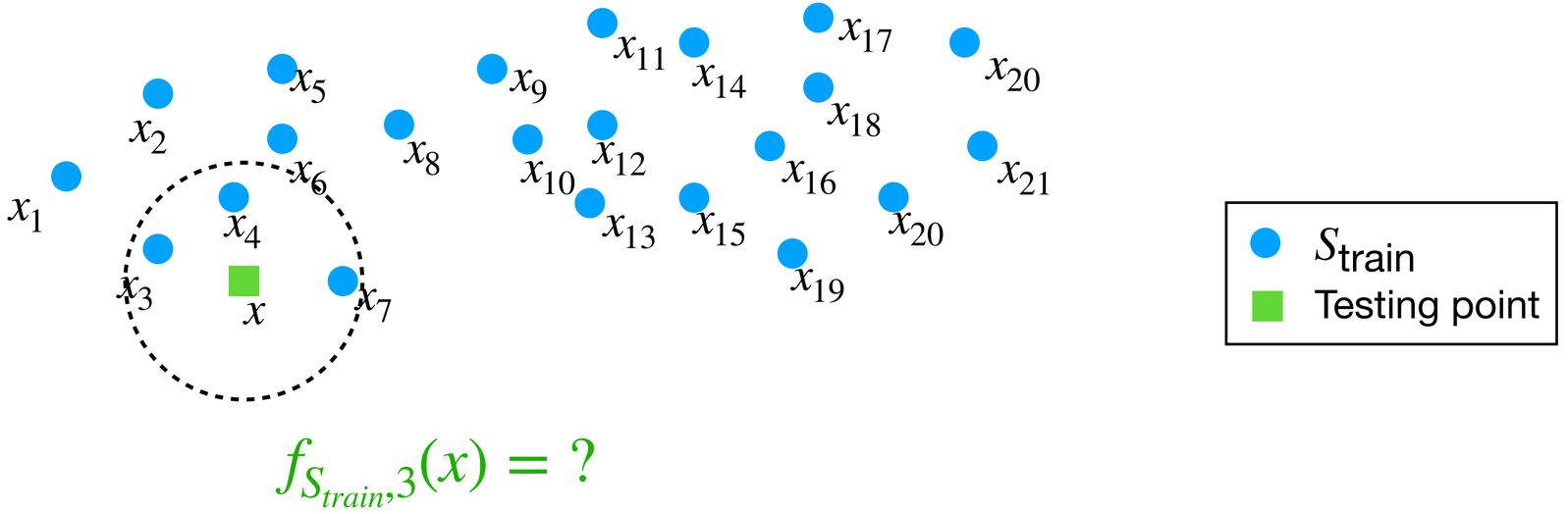
$$f_{S_{train},k}(x) = \frac{1}{k} \sum_{n:x_n \in nbh_{S_{train},k}(x)} y_n$$



$$f_{S_{train},3}(x) = ?$$

k-NN can be used for regression ($y \in \mathbb{R}$)

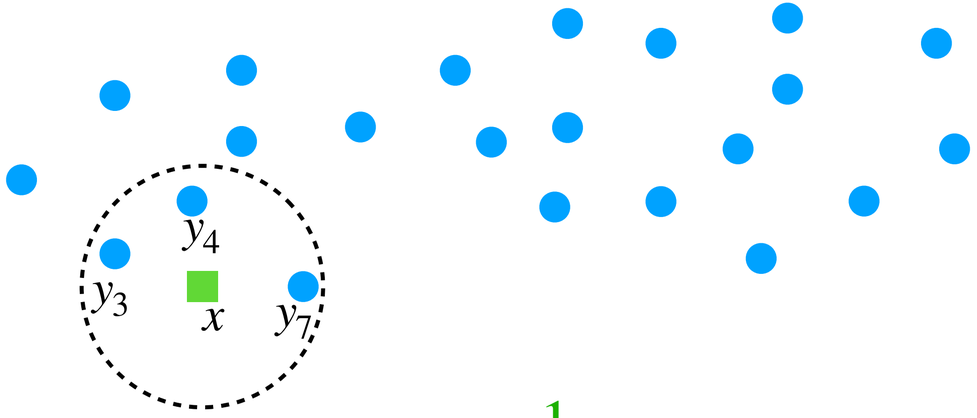
$$f_{S_{train},k}(x) = \frac{1}{k} \sum_{n:x_n \in nbh_{S_{train},k}(x)} y_n$$



$$f_{S_{train},3}(x) = ?$$

k-NN can be used for regression ($y \in \mathbb{R}$)

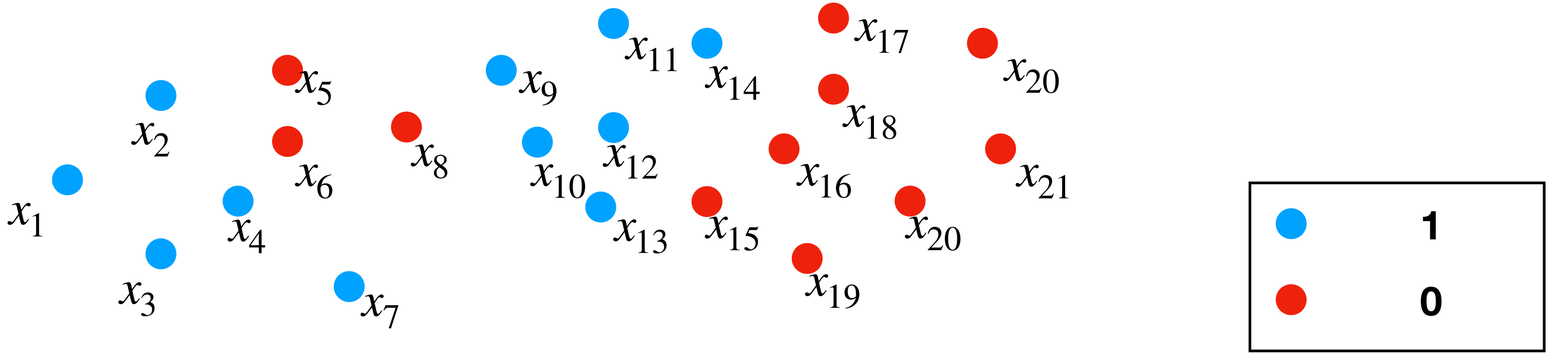
$$f_{S_{train},k}(x) = \frac{1}{k} \sum_{n:x_n \in nbh_{S_{train},k}(x)} y_n$$



$$f_{S_{train},3}(x) = \frac{1}{3}(y_3 + y_4 + y_7)$$

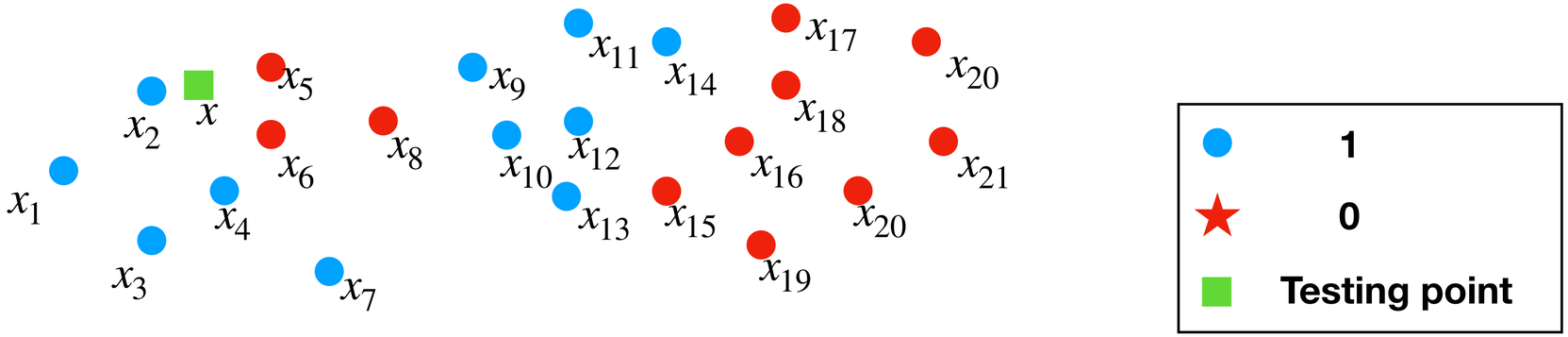
k-NN can be used for classification ($y \in \{0,1\}$)

$$f_{S_{train},k}(x) = \text{majority}\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\}$$



k-NN can be used for classification ($y \in \{0,1\}$)

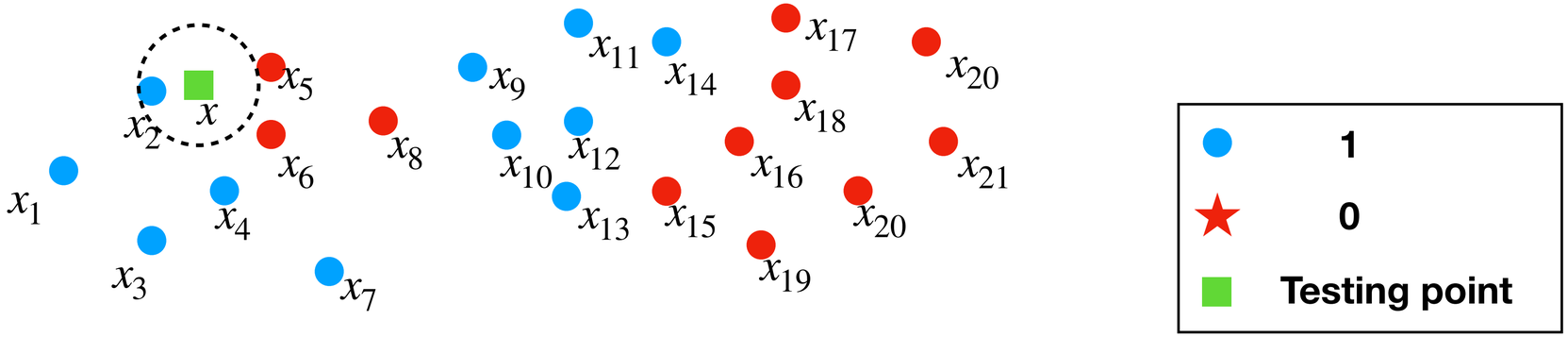
$$f_{S_{train},k}(x) = \text{majority}\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\}$$



$$f_{S_{train},1}(x) = ?$$

k-NN can be used for classification ($y \in \{0,1\}$)

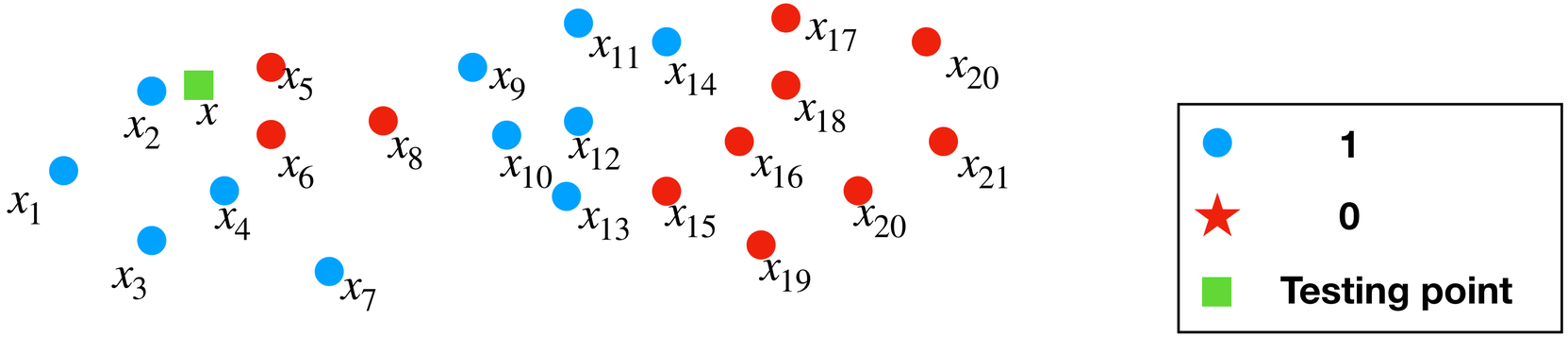
$$f_{S_{train},k}(x) = \text{majority}\left\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\right\}$$



$$f_{S_{train},1}(x) = 1$$

k-NN can be used for classification ($y \in \{0,1\}$)

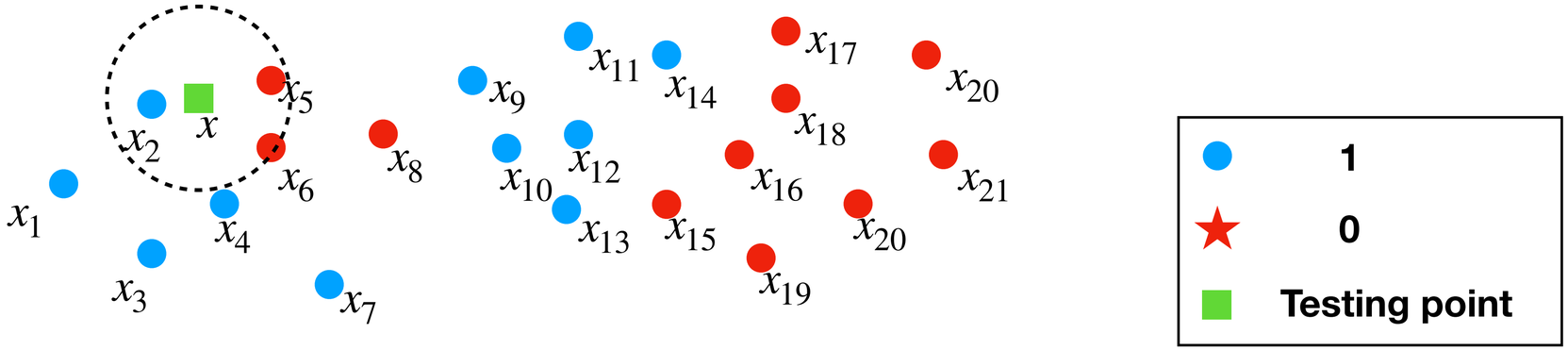
$$f_{S_{train},k}(x) = \text{majority}\left\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\right\}$$



$$f_{S_{train},3}(x) = ?$$

k-NN can be used for classification ($y \in \{0,1\}$)

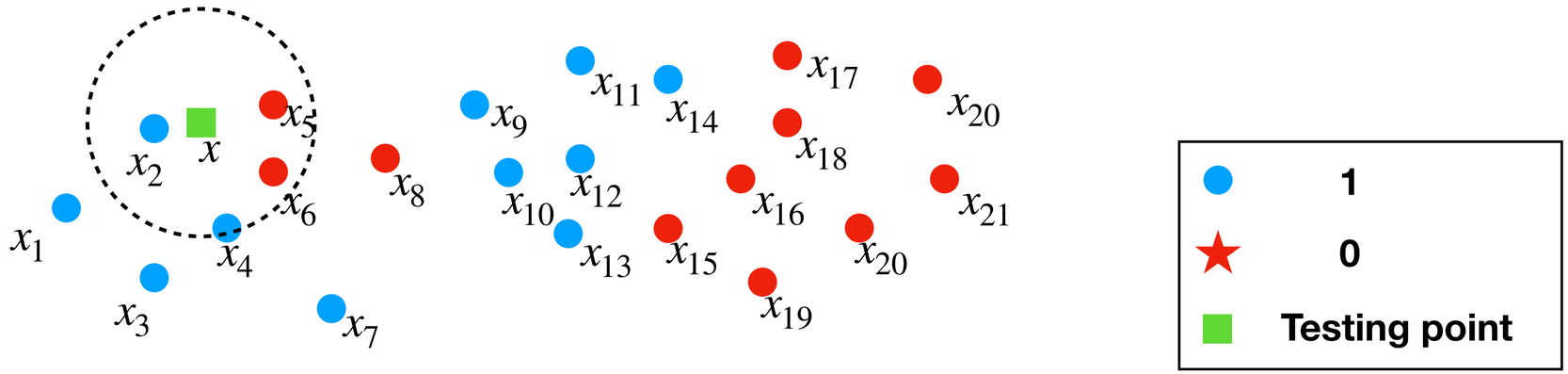
$$f_{S_{train},k}(x) = \text{majority}\left\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\right\}$$



$$f_{S_{train},3}(x) = 0$$

k-NN can be used for classification ($y \in \{0,1\}$)

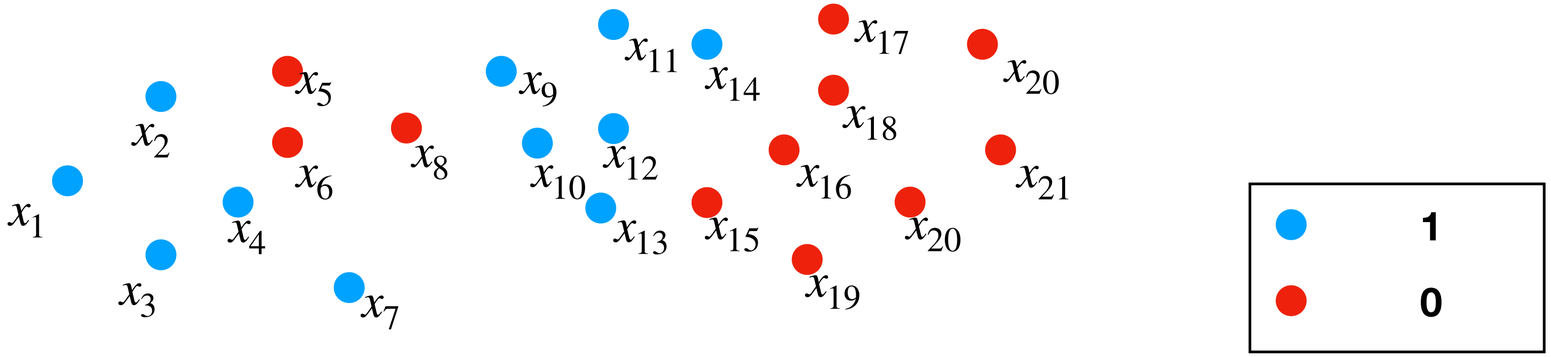
$$f_{S_{train},k}(x) = \text{majority}\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\}$$



$f_{S_{train},4}(x) = ?$ **Tie!**

k-NN can be used for classification ($y \in \{0,1\}$)

$$f_{S_{train},k}(x) = \text{majority}\{y_n : x_n \in \text{nbh}_{S_{train},k}(x)\}$$

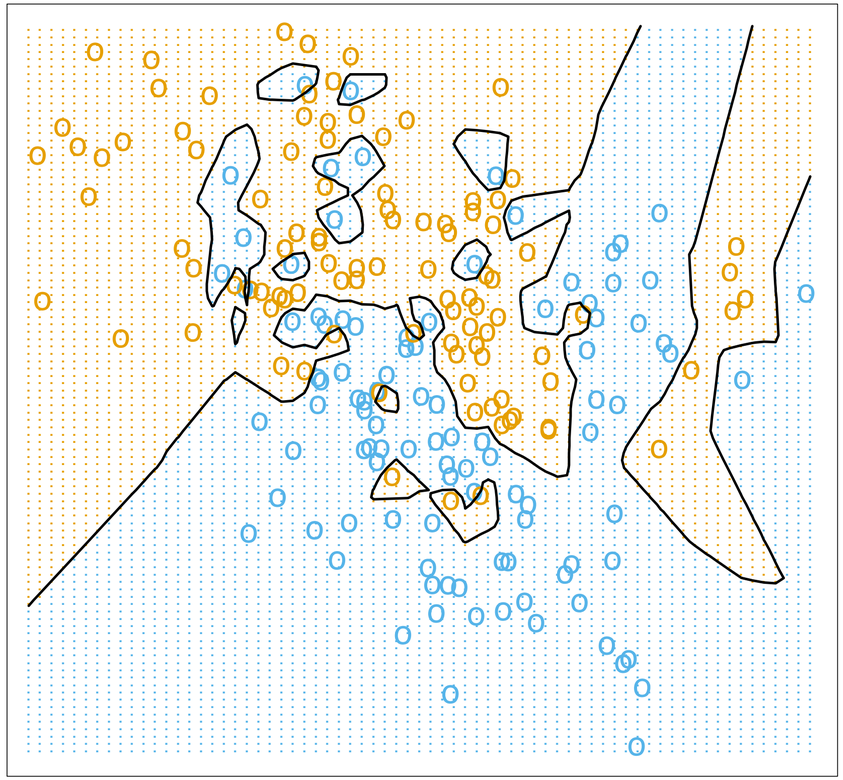


Remarks:

- k is often chosen odd to avoid ties
- Generalization: smoothing kernels; weighted linear combination of elements

Why does it make sense?

- Meaningful when there is spatial correlation
- Implicitly learns very complex decision boundaries in low dimension



Bias Variance for k-NN

Small k:

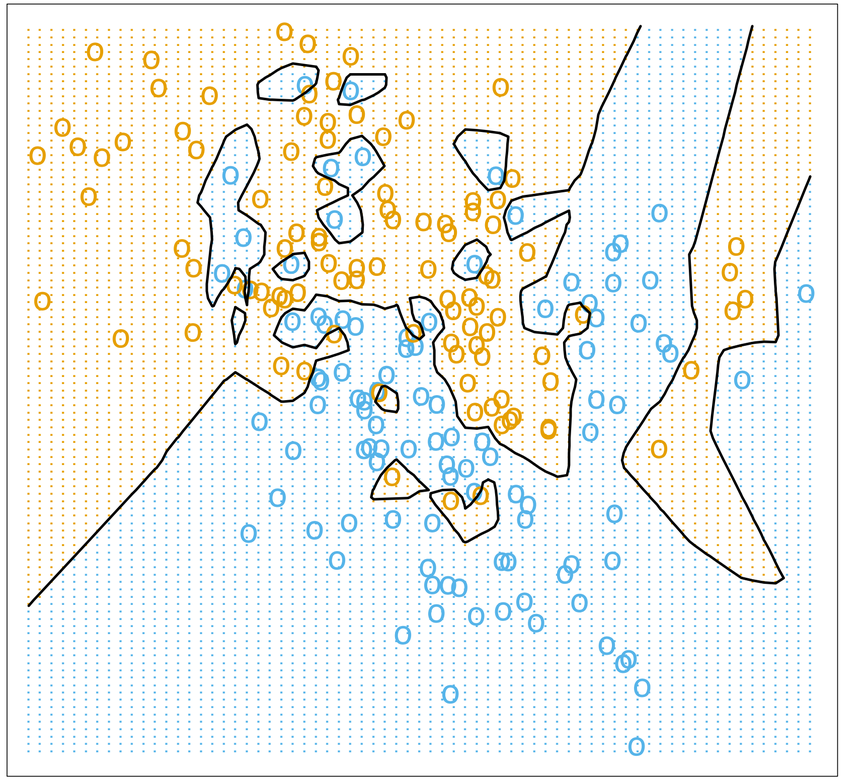
Small bias - complex decision boundary

Large variance - overfitting

Large k: ($k = N$ constant prediction)

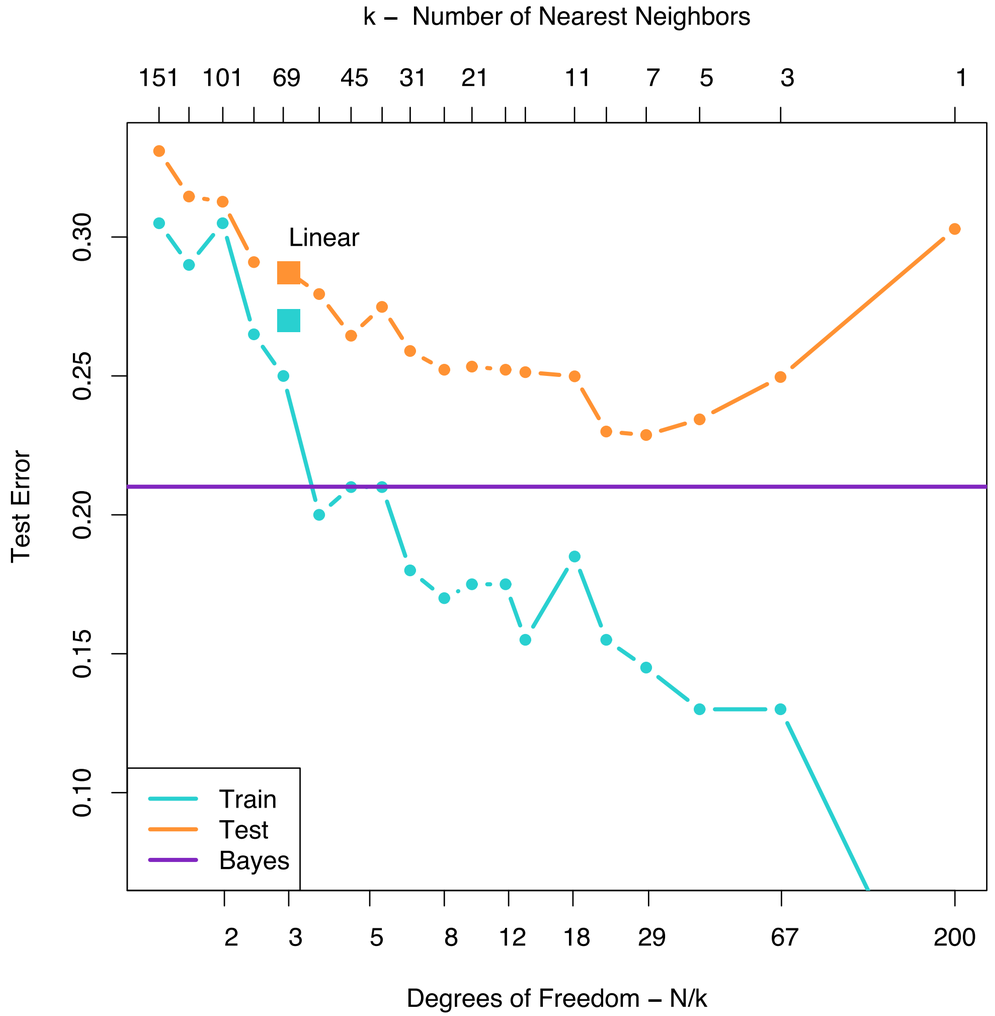
Large bias

Small variance



1-nearest neighbor classification

U-shape curve for k-NN bias-variance tradeoff



Complexity increases when k decreases

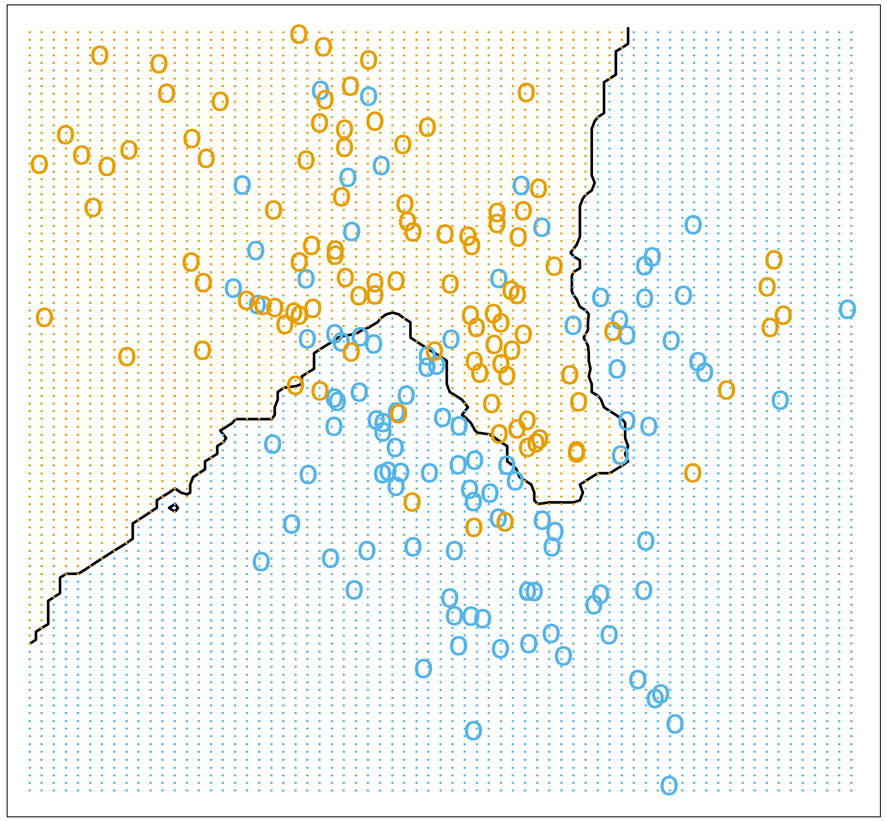
the larger the k → complex.

Find a k which balances the bias and the variance

<u>Good k:</u>

Small bias - complex enough decision boundary

Small variance - no overfitting



15-nearest neighbor classification

Curse of dimensionality

Claim 1: As the dimensionality grows, fixed-size training sets cover a dwindling fraction of the input space.

Assume the data $x \sim \mathcal{U}([0,1]^d)$

Consider a blue box around the center x_0 of size r

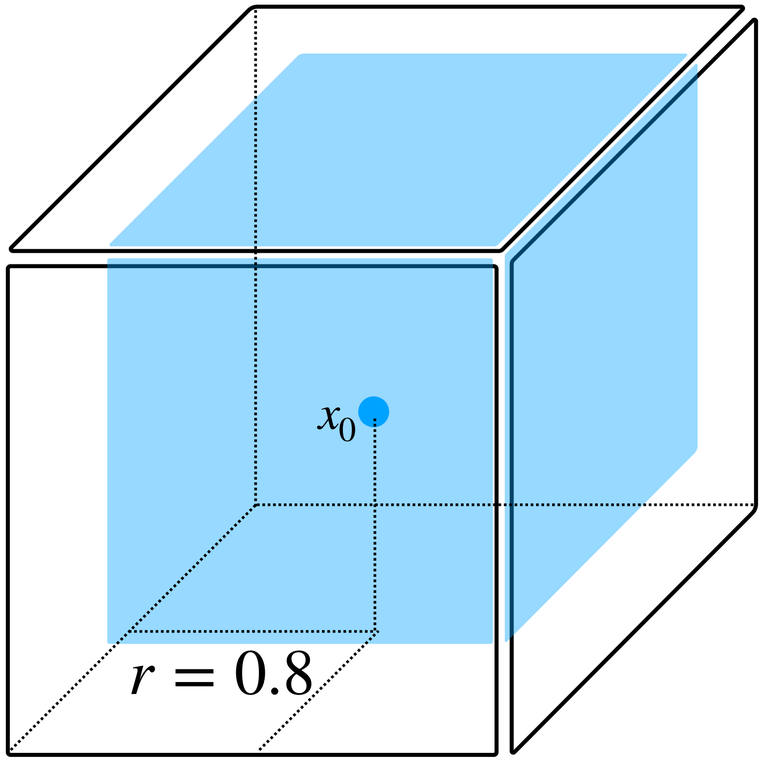
$$\mathbb{P}(x \in \square) = r^d := \alpha$$

If $d = 10$, to have:

$\alpha = 0.01$ we need $r = 0.63$

$\alpha = 0.1$ we need $r = 0.8$

We need to explore almost the whole box



Curse of dimensionality

Claim 2: In high-dimension, data-points are far from each other.

Consider N i.i.d. points uniform in the $[0,1]^d$

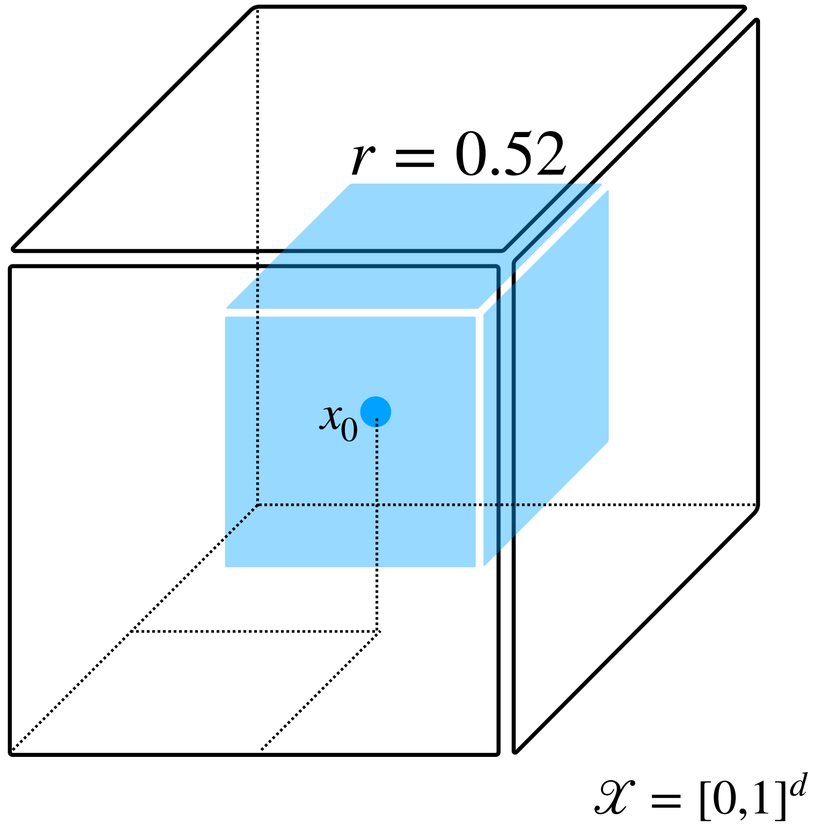
$$\mathbb{P}(\exists x_i \in \square) \geq 1/2 \implies r \geq \left(1 - \frac{1}{2^{1/N}}\right)^{1/d}$$

Proof: $\mathbb{P}(x \notin \square) = 1 - r^d$

$$\mathbb{P}(x_i \notin \square, \forall i \leq N) = (1 - r^d)^N$$

$$\mathbb{P}(\exists x_i \in \square) = 1 - (1 - r^d)^N$$

For $d = 10$, $N = 500$, we have $r \geq 0.52$



Generalization bound for 1-NN

Setup: $(X, Y) \sim \mathscr{D}$ over $\mathscr{X} \times \mathscr{Y} = [0,1]^d \times \{0,1\}$

Goal: Bound the classification error:

$$L(g) = \mathbb{P}_{(X,Y)\sim\mathscr{D}}(Y \neq g(X))$$

Baseline:

- Bayes classifier - minimizes L over all classifiers

$$g_*(x) = 1_{\eta(x)\geq 1/2} \text{ where } \eta(x) = \mathbb{P}(Y = 1 \,|\, X = x)$$

- Bayes risk - smallest probability of misclassification

$$L(g_*) = \mathbb{P}(g_*(X) \neq Y) = \mathbb{E}_{X\sim\mathscr{D}_X}[\min\{\eta(X), 1 - \eta(X)\}]$$

Generalization bound for 1-NN

Setup: $(X, Y) \sim \mathscr{D}$ ov

Goal: Bound the classif

Proof 1:

$$\eta(x) \geq 1/2 \iff \mathbb{P}(Y = 1 \,|\, X = x) \geq 1/2$$
$$\iff \mathbb{P}(Y = 1 \,|\, X = x) \geq \mathbb{P}(Y = 0 \,|\, X = x)$$
$$\iff 1 \in \arg\max_{y \in \{0,1\}} \mathbb{P}(Y = y \,|\, X = x)$$

Thus $1_{\eta(x) \geq 1/2} = \arg\max_{y \in \{0,1\}} \mathbb{P}(Y = y \,|\, X = x) = g_*(x)$

Baseline:

- Bayes classifier - minimizes L over all classifiers

$$g_*(x) = 1_{\eta(x) \geq 1/2} \text{ where } \eta(x) = \mathbb{P}(Y = 1 \,|\, X = x)$$

- Bayes risk - smallest probability of misclassification

$$L(g_*) = \mathbb{P}(g_*(X) \neq Y) = \mathbb{E}_{X \sim \mathscr{D}_X}[\min\{\eta(X), 1 - \eta(X)\}]$$

Generalization bound for 1-NN

Proof 2:

$$
\begin{aligned}
L(g_*) &= \mathbb{E}_{(X,Y)\sim\mathscr{D}}[1_{g_*(X)\neq Y}] \\
&= \mathbb{E}_{X\sim\mathscr{D}_X}[\mathbb{E}_{Y\sim\mathscr{D}_{Y|X}}[1_{g_*(X)\neq Y}|X]] \\
&= \mathbb{E}_{X\sim\mathscr{D}_X}[\mathbb{E}_{Y\sim\mathscr{D}_{Y|X}}[1_{g_*(X)\neq Y}|X]1_{\eta(X)\geq 1/2} + E_{Y\sim\mathscr{D}_{Y|X}}[1_{g_*(X)\neq Y}|X]1_{\eta(X)<1/2}] \\
&= \mathbb{E}_{X\sim\mathscr{D}_X}[\mathbb{E}_{Y\sim\mathscr{D}_{Y|X}}[1_{1\neq Y}|X]1_{\eta(X)\geq 1/2} + E_{Y\sim\mathscr{D}_{Y|X}}[1_{0\neq Y}|X]1_{\eta(X)<1/2}] \\
&= \mathbb{E}_{X\sim\mathscr{D}_X}[\mathbb{P}(Y=0\,|\,X)1_{\eta(X)\geq 1/2} + \mathbb{P}(Y=1\,|\,X)1_{\eta(X)<1/2}] \\
&= \mathbb{E}_{X\sim\mathscr{D}_X}[\min\{\eta(X),1-\eta(X)\}]
\end{aligned}
$$

- Bayes risk - smallest probability of misclassification

$$L(g_*) = \mathbb{P}(g_*(X) \neq Y) = \mathbb{E}_{X\sim\mathscr{D}_X}[\min\{\eta(X),1-\eta(X)\}]$$

Generalization bound for 1-NN

Assumption: $\exists c \geq 0,\ \forall x, x' \in \mathcal{X}$:

$$|\eta(x) - \eta(x')| \leq c\|x - x'\|_2$$

➡ Nearby points are likely to have the same label

Claim:

$$\mathbb{E}_{S_{train}}[L(g_{S_{train}})] \leq 2L(g_*) + c\mathbb{E}_{S_{train}, X\sim\mathcal{D}_X}[\|X - \text{nbh}_{S_{train},1}(X)\|]$$

$$\leq 2L(g_*) + 4c\sqrt{d}N^{-\frac{1}{d+1}}$$

geometric term:
average distance between a random point and x

Interpretation:

Fixed d and $N \to \infty$: $\mathbb{E}_{S_{train}}[L(g_{S_{train}})] \leq 2L(g_*)$

Fixed N and $d \to \infty$: bound increases exponentially fast

Interpolation method can generalize well: against common belief

Proof

We want to bound

$$\mathbb{E}_{S_{train}}[L(g_{S_{train}})] = \mathbb{E}_{S_{train}}[\mathbb{P}_{(X,Y)\sim\mathscr{D}}[g_{S_{train}}(X) \neq Y]]$$

We first sample N unlabeled examples $S_{train,X} = (X_1, \cdots X_N) \sim \mathscr{D}_X$, an unlabeled example $X \sim \mathscr{D}$ and define $X' = \text{nbh}_{S_{train},1}(X)$

Finally we sample $Y \sim \eta(X)$ and $Y' \sim \eta(X')$

We have:

$$\begin{aligned}\mathbb{E}_{S_{train}}[L(g_{S_{train}})] &= \mathbb{E}_{S_{X,train},X\sim\mathscr{D}_X,Y\sim\eta(X),Y'\sim\eta(X')}[1_{Y\neq g_{S_{train}(X)}}] \\ &= \mathbb{E}_{S_{X,train},X\sim\mathscr{D}_X,Y\sim\eta(X),Y'\sim\eta(X')}[1_{Y\neq Y'}] \\ &= \mathbb{E}_{S_{train,X}X\sim\mathscr{D}_X}[\mathbb{P}_{Y\sim\eta(X),Y'\sim\eta(X')}(Y \neq Y')]\end{aligned}$$

Proof

Consider two points $x, x' \in [0,1]^d$.

Sample their labels $Y \sim \eta(x)$ and $Y' \sim \eta(x')$

<u>Claim:</u>

$$\mathbb{P}(Y' \neq Y) \leq 2\min\{\eta(x), 1-\eta(x)\} + c\|x - x'\|$$

- Simple case: $x = x'$

$$\begin{aligned}\mathbb{P}(Y' \neq Y) &= \mathbb{E}[1_{Y' \neq Y} 1_{Y'=1} + 1_{Y' \neq Y} 1_{Y'=0}] \\ &= \mathbb{P}(Y'=1)\mathbb{P}(Y=0) + \mathbb{P}(Y'=1)\mathbb{P}(Y=0) \\ &= 2\eta(x)(1-\eta(x)) \\ &\leq 2\min\{\eta(x), 1-\eta(x)\}\end{aligned}$$

Case 1:

Y=0 $(1-\eta(x))$

Y'=1 $\eta(x)$

Case 2:

Y=1 $\eta(x)$

Y'=0 $(1-\eta(x))$

Proof

- General case:

$$
\begin{aligned}
\mathbb{P}(Y \neq Y') &= \eta(x)(1-\eta(x')) + \eta(x')(1-\eta(x)) \\
&= \eta(x)(1-\eta(x)) + \eta(x)(\eta(x)-\eta(x')) \\
&\quad + \eta(x)(1-\eta(x)) + (\eta(x')-\eta(x))(1-\eta(x)) \\
&= 2\eta(x)(1-\eta(x)) + (2\eta(x)-1)(\eta(x)-\eta(x')) \\
&\leq 2\eta(x)(1-\eta(x)) + |(2\eta(x)-1)|\,|\eta(x)-\eta(x')| \\
&\leq 2\eta(x)(1-\eta(x)) + |\eta(x)-\eta(x')| \\
&\leq 2\eta(x)(1-\eta(x)) + c\|x-x'\| \\
&\leq 2\min\{\eta(x), 1-\eta(x)\} + c\|x-x'\|
\end{aligned}
$$

Proof

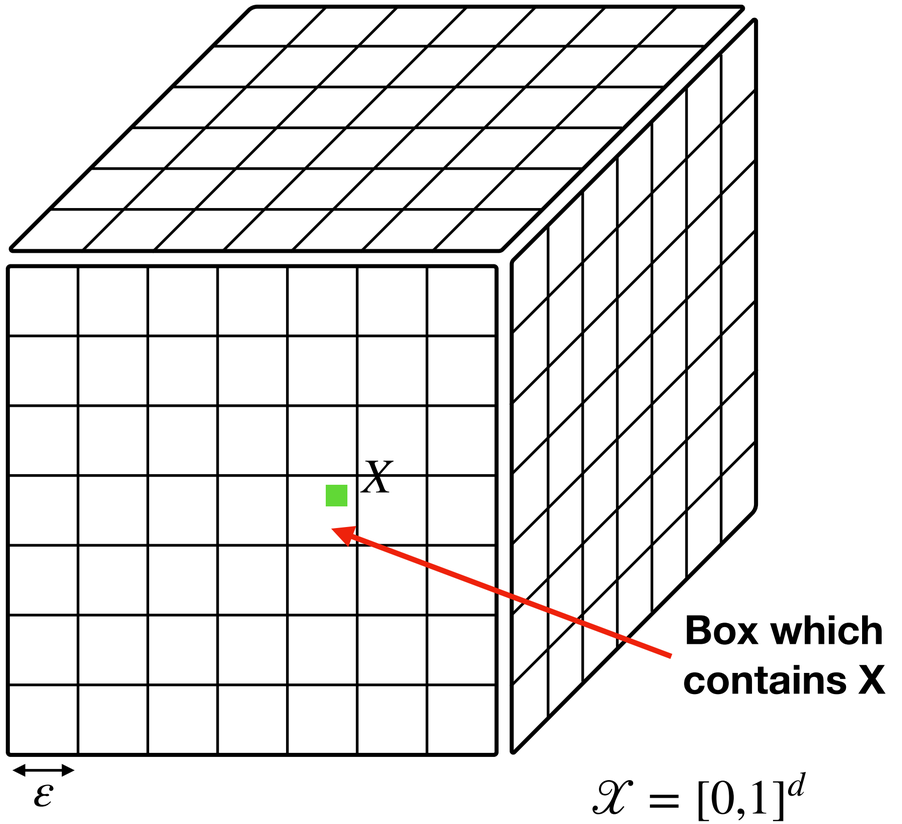
$$
\begin{aligned}
\mathbb{E}_{S_{train}}[L(g_{S_{train}})] &= \mathbb{E}_{S_{X,train}, X\sim\mathscr{D}_X, Y\sim\eta(X), Y'\sim\eta(X')}[1_{Y\neq g_{S_{train}}(X)}] \\
&= \mathbb{E}_{S_{X,train}, X\sim\mathscr{D}_X, Y\sim\eta(X), Y'\sim\eta(X')}[1_{Y\neq Y'}] \\
&= \mathbb{E}_{S_{train,X} X\sim\mathscr{D}_X}[\mathbb{P}_{Y\sim\eta(X), Y'\sim\eta(X')}(Y \neq Y')] \\
&\leq \mathbb{E}_{S_{train,X} X\sim\mathscr{D}_X}[2\min\{\eta(X), 1-\eta(X)\} + c\|X - X'\|] \\
&\leq 2L(g_*) + c\mathbb{E}_{S_{train} X\sim\mathscr{D}_X}[\|X - \mathsf{nbh}_{S_{train},1}(X)\|]
\end{aligned}
$$

Bound on the geometric term

Consider a fresh sample $X \sim \mathscr{D}$ and denote by

$p_k = \mathbb{P}(X \in \text{Box}_k)$

Consider the box which contains X. Two options:



ε- cover of the Hypercube

Bound on the geometric term

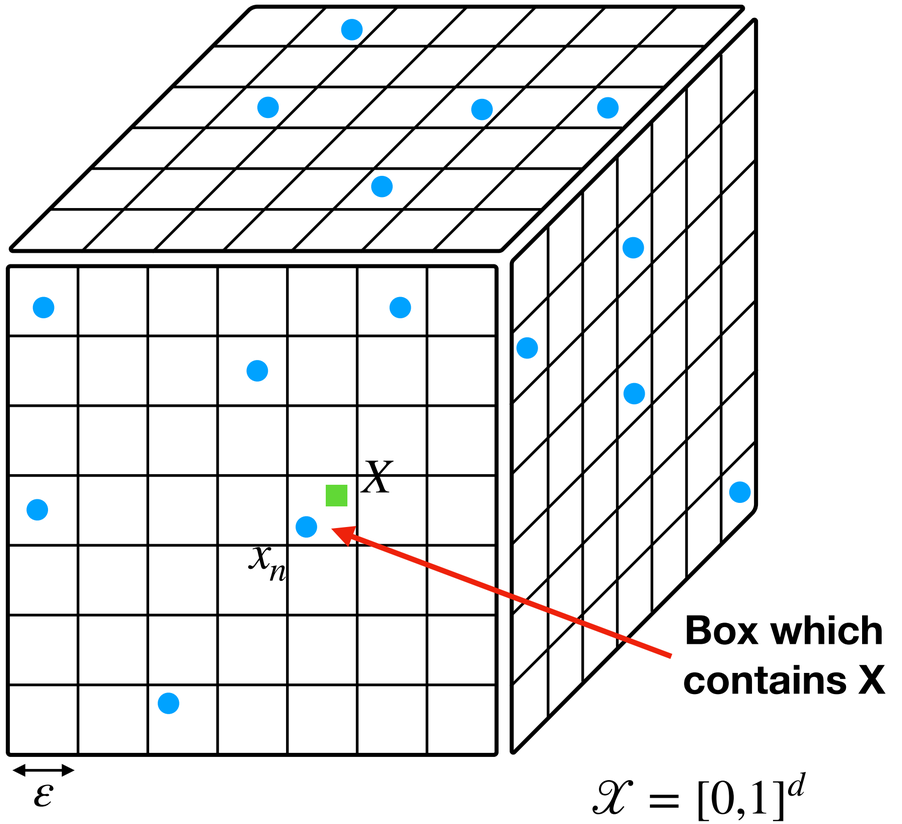
Consider a fresh sample $X \sim \mathscr{D}$ and denote by

$p_k = \mathbb{P}(X \in \text{Box}_k)$

Consider the box which contains X. Two options:

- The box contains an element of S_{train}. X has a neighbor in S_{train} at distance at most $\sqrt{d}\varepsilon$

It happens with probability $1 - (1 - p_k)^N$



ε- cover of the Hypercube

Bound on the geometric term

Consider a fresh sample $X \sim \mathcal{D}$ and denote by

$p_k = \mathbb{P}(X \in \text{Box}_k)$

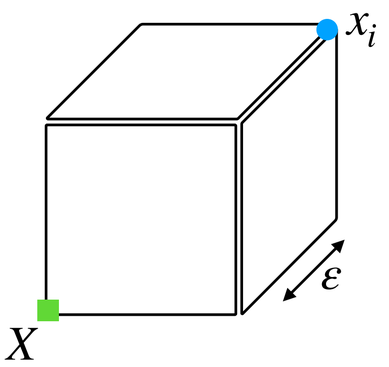
Consider the box which contains X. Two options:

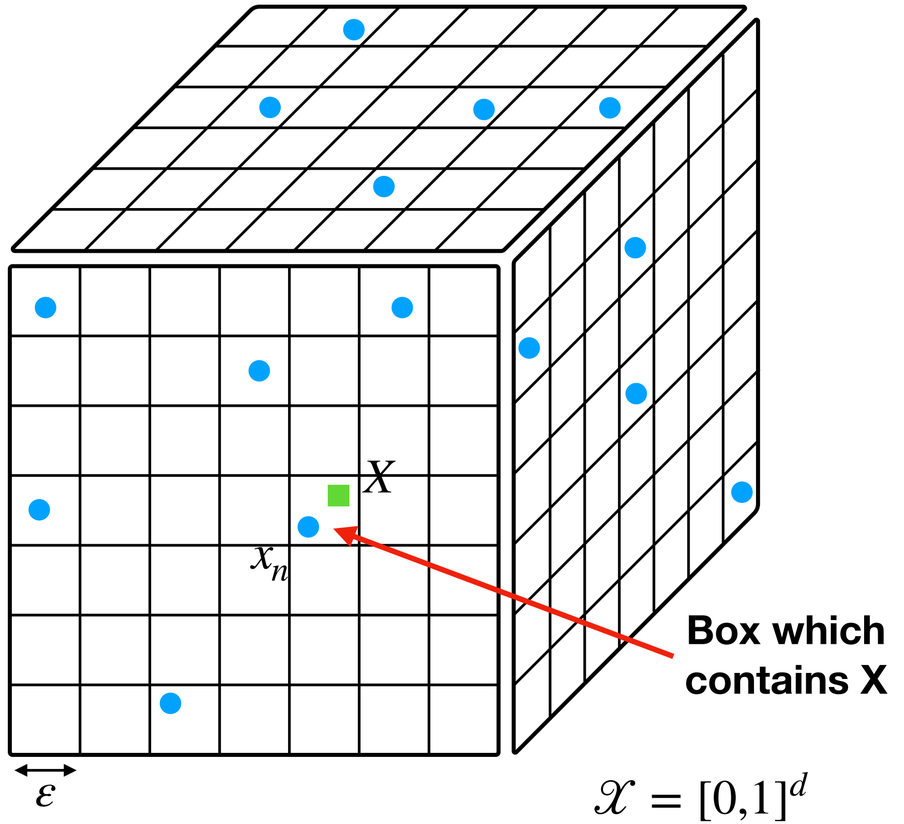
- The box contains an element of S_{train}. X has a neighbor in S_{train} at distance at most $\sqrt{d}\varepsilon$

It happens with probability $1 - (1 - p_k)^N$

Proof: Consider the worst case:

$$\|X - x_i\| = \sqrt{\sum_{i=1}^{d} \varepsilon^2} = \sqrt{d}\varepsilon$$





ε- cover of the Hypercube

Bound on the geometric term

Consider a fresh sample $X \sim \mathscr{D}$ and denote by

$p_k = \mathbb{P}(X \in \text{Box}_k)$

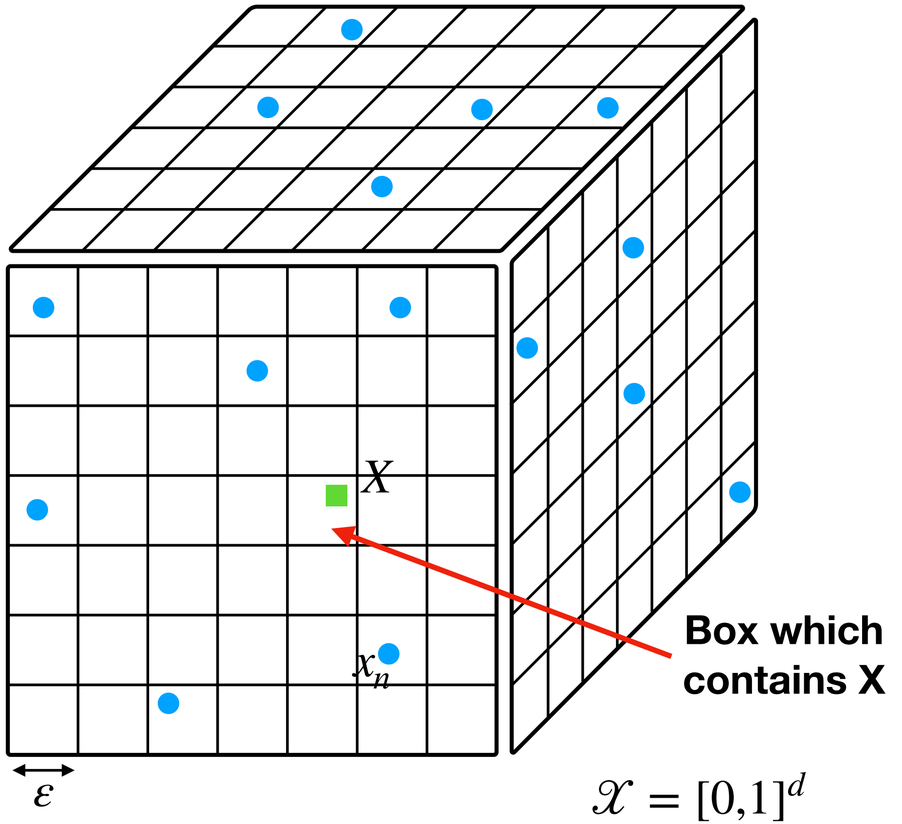
Consider the box which contains X. Two options:

- The box contains an element of S_{train}. X has a neighbor in S_{train} at distance at most $\sqrt{d}\varepsilon$

It happens with probability $1 - (1 - p_k)^N$

- There is no element of S_{train}. The nearest neighbor of X can be at worst at a distance $\sqrt{d}$

It happens with probability $(1 - p_k)^N$



ε- cover of the Hypercube

Bound on the geometric term

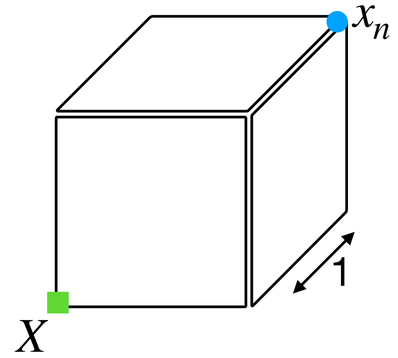
Consider a fresh sample $X \sim \mathcal{D}$ and denote by

$p_k = \mathbb{P}(X \in \text{Box}_k)$

C

Proof: Consider the worst case:

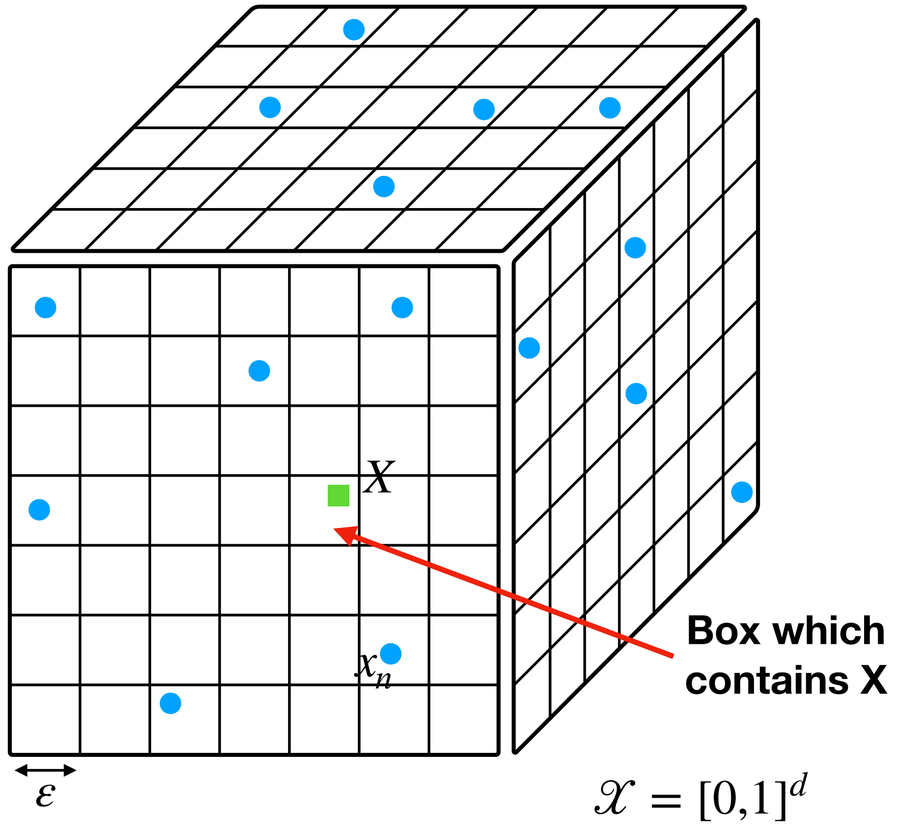
$$\|X - x_i\| = \sqrt{\sum_{i=1}^{d} 1} = \sqrt{d}$$



of

X can be at worst at a distance $\sqrt{d}$

It happens with probability $(1 - p_k)^N$



ε- cover of the Hypercube

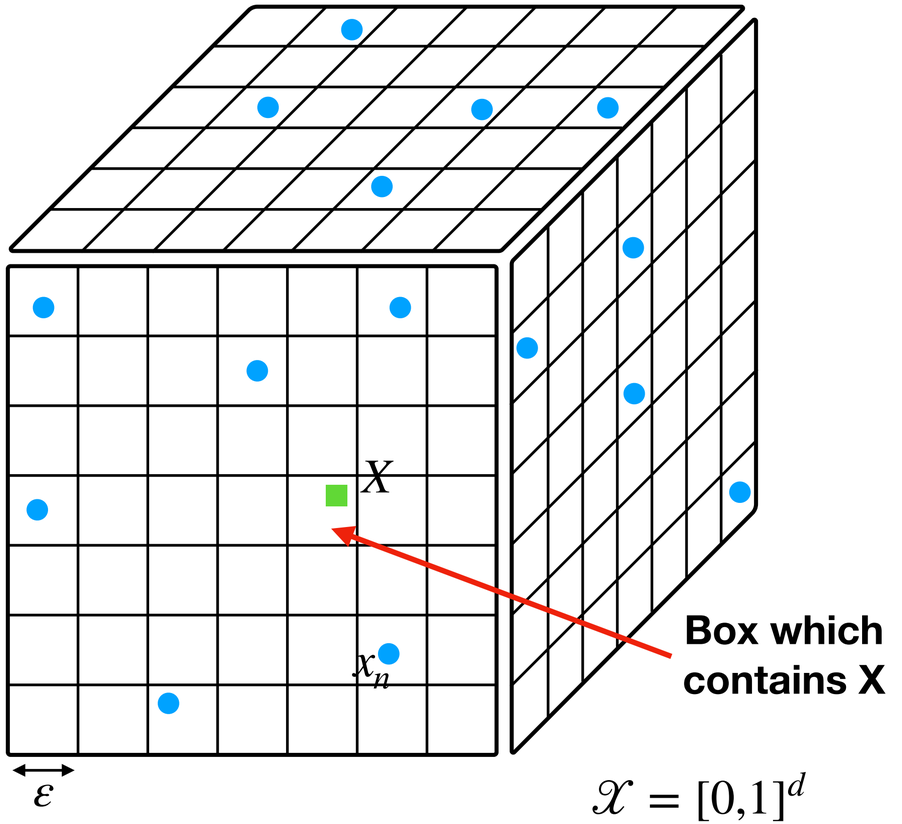
Bound on the geometric term

$$\mathbb{E}[\|X - \text{nbh}(X)\|] \leq \sum_k p_k[(1-p_k)^N\sqrt{d} + (1-(1-p_k)^N)\sqrt{d}\varepsilon]$$

Claim: we get the bound by maximizing over p_k and ε

Intuition:

- If p_k is large: it is likely that I pick that box but it is also likely that I find a training point in that box
- If p_k is small, then we are fine since by definition this does not happen very often.



ε- cover of the Hypercube

Bonus: Nearest Neighbors is a local averaging method

Local averaging methods aim at approximating the Bayes predictor directly - without optimization

This is done by approximating the condition distribution $p(y \mid x)$ by some $\hat{p}(y \mid x)$

These “plug-in” estimators are:

- $g(x) \in \arg\max_{y \in \mathcal{Y}} \hat{\mathbb{P}}(Y = y \mid x)$ for classification with the 0-1 loss
- $g(x) = \hat{\mathbb{E}}[Y \mid x] = \int_{\mathcal{Y}} y \hat{p}(y \mid x) dy$ for regression with the square loss

In the case of nearest neighbors:

$$\hat{p}(y \mid x) = \sum_{n=1}^{N} \hat{w}_n(x) 1_{y=y_n}$$

where $\hat{w}(x) = 1/k$ for the k nearest neighbors (0 otherwise)