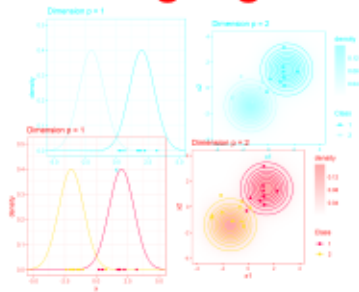


# Introduction to Machine Learning

## Curse of Dimensionality - Examples

## Curse of Dimensionality - Examples

## Learning Algorithms



### Learning goals

- See how the performance of k-NN and the linear model deteriorates in high-dimensional spaces

### Learning goals

- See how the performance of k-NN and the linear model deteriorates in high-dimensional spaces

## EXAMPLE: K-NN / 2

Minimal, mean and maximal (NN)-distances of  $10^4$  points uniformly distributed in the hypercube  $[0, 1]^p$ :

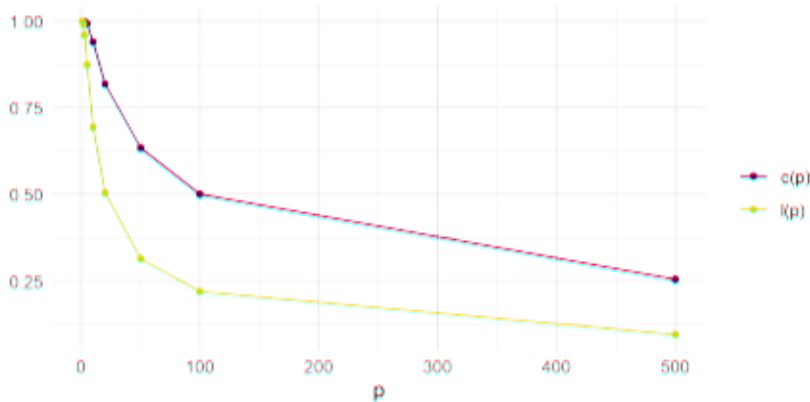
$p$	$\min d(\mathbf{x}, \tilde{\mathbf{x}})$	$\overline{d(\mathbf{x}, \tilde{\mathbf{x}})}$	$\max d(\mathbf{x}, \tilde{\mathbf{x}})$	$\overline{d_{NN1}(\mathbf{x})}$	$\max d_{NN1}(\mathbf{x})$
1	1.2e-08	0.33	1	5e-05	0.00042
2	0.00011	0.52	1.4	0.0051	0.02
3	0.0021	0.66	1.7	0.026	0.073
5	0.016	0.88	2	0.11	0.23
10	0.15	1.3	2.5	0.39	0.63
20	0.55	1.8	3	0.9	1.2
50	1.5	2.9	4.1	2	2.4
100	2.7	4.1	5.4	3.2	3.5
500	7.8	9.1	10	8.2	8.6



## EXAMPLE: K-NN / 3

We see a decrease of relative contrast<sup>1</sup>  $c := \frac{\max(d(\mathbf{x}, \bar{\mathbf{x}})) - \min(d(\mathbf{x}, \bar{\mathbf{x}}))}{\max(d(\mathbf{x}, \bar{\mathbf{x}}))}$  and

“locality”<sup>2</sup>  $l := \frac{d(\mathbf{x}, \bar{\mathbf{x}}) - d_{NN1}(\mathbf{x})}{d(\mathbf{x}, \bar{\mathbf{x}})}$  with increasing number of dimensions  $p$ :



<sup>1</sup>[Aggarwal et al., 2001]

<sup>2</sup>our non-standard definition

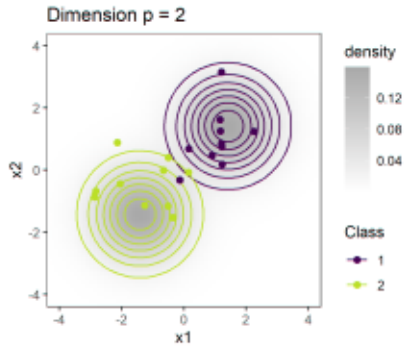
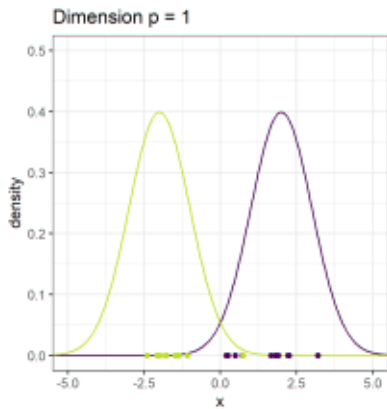
## EXAMPLE: K-NN / 2

To demonstrate this, we generate an artificial data set of dimension  $p$  as follows: We define  $a = \frac{2}{\sqrt{p}}$  and

- with probability  $\frac{1}{2}$  we generate a sample from class 1 by sampling from a Gaussian with mean  $\boldsymbol{\mu} = (a, a, \dots, a)$  and unit covariance matrix
- with probability  $\frac{1}{2}$  we generate a sample from class 2 by sampling from a Gaussian with mean  $-\boldsymbol{\mu} = (-a, -a, \dots, -a)$  and unit covariance matrix



# EXAMPLE: K-NN / 3



## EXAMPLE: K-NN / 4

This example is constructed such that the Bayes error is always constant and does not depend on the dimension  $p$ .

The Bayes optimal classifier predicts  $\hat{y} = 1$  iff

$$\begin{aligned}\mathbb{P}(y = 1 | \mathbf{x}) &= \frac{p(\mathbf{x} | y = 1)\mathbb{P}(y = 1)}{p(\mathbf{x})} = \frac{1}{2} \cdot \frac{p(\mathbf{x} | y = 1)}{p(\mathbf{x})} \\ &\geq \frac{1}{2} \cdot \frac{p(\mathbf{x} | y = 2)}{p(\mathbf{x})} \\ &= \frac{p(\mathbf{x} | y = 2)\mathbb{P}(y = 2)}{p(\mathbf{x})} = \mathbb{P}(y = 2 | \mathbf{x}).\end{aligned}$$

This is equivalent to

$$\begin{aligned}\hat{y} = 1 &\Leftrightarrow \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top(\mathbf{x} - \boldsymbol{\mu})\right) \geq \exp\left(-\frac{1}{2}(\mathbf{x} + \boldsymbol{\mu})^\top(\mathbf{x} + \boldsymbol{\mu})\right) \\ &\Leftrightarrow \mathbf{x}^\top \boldsymbol{\mu} \geq 0.\end{aligned}$$





## EXAMPLE: K-NN / 6

We can calculate the corresponding expected misclassification error (Bayes error)

$$\begin{aligned} & \rho(\hat{y} = 1 | y = 2)\mathbb{P}(y = 2) + \rho(\hat{y} = 2 | y = 1)\mathbb{P}(y = 1) \\ &= \frac{1}{2} \cdot \rho(\mathbf{x}^\top \boldsymbol{\mu} \geq 0 | y = 2) + \frac{1}{2} \cdot \rho(\mathbf{x}^\top \boldsymbol{\mu} \leq 0 | y = 1) \\ &\stackrel{\text{symm.}}{=} \rho(\mathbf{x}^\top \boldsymbol{\mu} \leq 0 | y = 1) = \rho\left(\sum_{i=1}^p a\mathbf{x}_i \leq 0 | y = 1\right) \\ &= \rho\left(\sum_{i=1}^p \mathbf{x}_i \leq 0 | y = 1\right). \end{aligned}$$

$\sum_{i=1}^p \mathbf{x}_i | y = 1 \sim \mathcal{N}(p \cdot a, p)$ , because it is the sum of independent normal random variables  $\mathbf{x}_i | y = 1 \sim \mathcal{N}(a, 1)$  (the vector  $\mathbf{x} | y = 1$  follows a  $\mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$  distribution with  $\boldsymbol{\mu} = (a, \dots, a)$ ).





## EXAMPLE: K-NN / 7

We get for the Bayes error:

$$\begin{aligned} &= p \left( \frac{\sum_{i=1}^p \mathbf{x}_i - p \cdot a}{\sqrt{p}} \leq \frac{-p \cdot a}{\sqrt{p}} \mid y = 1 \right) \\ &= \Phi(-\sqrt{p}a) \stackrel{a=\frac{2}{\sqrt{p}}}{=} \Phi(-2) \approx 0.0228, \end{aligned}$$

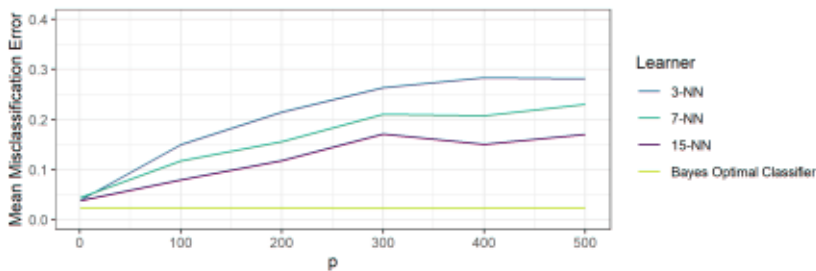
where  $\Phi$  is the distribution function of a standard normal random variable.

We see that the Bayes error is independent of  $p$ .



## EXAMPLE: K-NN / 8

We also train a k-NN classifier for  $k = 3, 7, 15$  for increasing dimensions and monitor its performance (evaluated by 10 times repeated 10-fold CV).



→ k-NN deteriorates quickly with increasing dimension



## EXAMPLE: LINEAR MODEL / 3

- The regression coefficients of the noise features can not be estimated precisely as zero in the unregularized LM due to small random correlations.
- With an increasing number of these noise features, the prediction error rises.
- To see this, we can quantify the influence of the noise features on the prediction of each observation.

Therefore we decompose the response  $\hat{y}^{(i)}$  of each iterations' test set into  $\hat{y}_{\text{true}}^{(i)}$  (predicted with noise features set to 0) and  $\hat{y}_{\text{noise}}^{(i)}$  (predicted with true features set to 0), s.t.

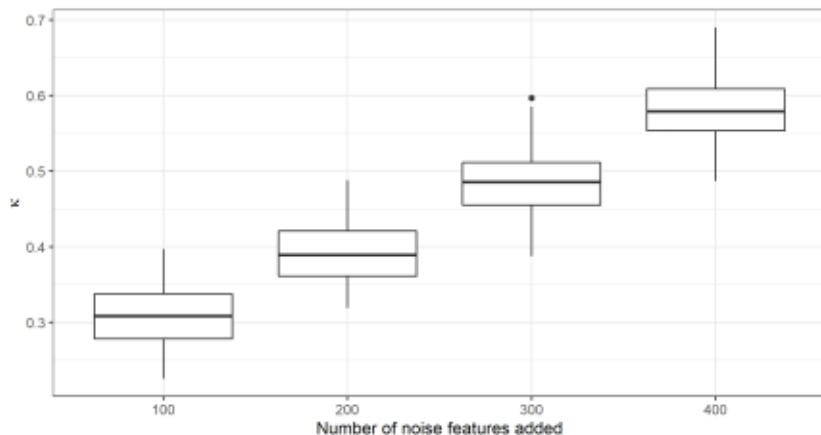
$$\hat{y}^{(i)} = \hat{y}_{\text{true}}^{(i)} + \hat{y}_{\text{noise}}^{(i)} + \text{intercept.}$$

With this, we can define the "average proportional influence of the

noise features"  $\kappa := \overline{\left( \frac{|\hat{y}_{\text{noise}}^{(i)}|}{|\hat{y}_{\text{true}}^{(i)}| + |\hat{y}_{\text{noise}}^{(i)}|} \right)}$ .



## EXAMPLE: LINEAR MODEL / 4



When we add 400 noise features to the model, most of the time, on average, over 50% of the flexible part of the prediction ( $\hat{y}^{(i)} - \text{intercept}$ ) is determined by the noise features.