

# Lecture 2b: Model evaluation and bias-variance tradeoff

---

Rebecka Jörnsten, Mathematical Sciences

**MSA220/MVE441** Statistical Learning for Big Data

24-27<sup>th</sup> March 2025



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

## **Evaluating performance of a statistical method**

---

# Goals

---

- ▶ **Model selection:** Choose a hyper-parameter or model structure, e.g.  $k$  in kNN regression/classification, or 'Choose between logistic regression, LDA and kNN'
- ▶ **Model assessment:** How well did a model do on a data set?

# UCI Breast Cancer Wisconsin (Diagnostic) Data Set

---

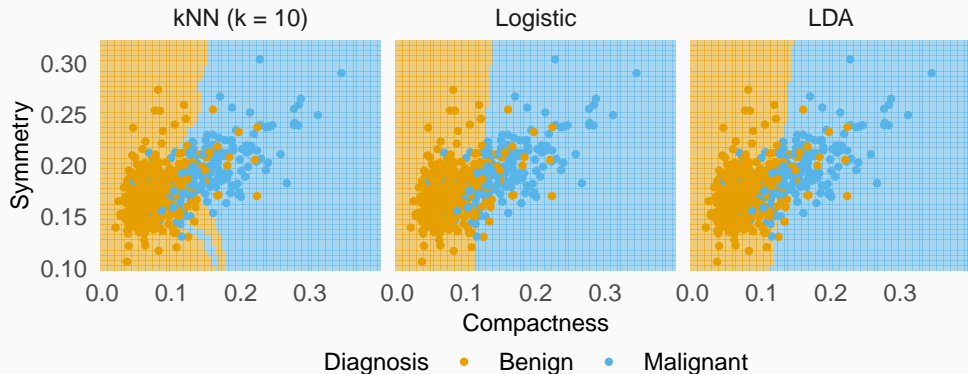
## UCI Breast Cancer Wisconsin (Diagnostic) Data Set<sup>1</sup>

- ▶ Classification data set with binary response (**malignant** or **benign** cancer)
- ▶ 569 samples (357 benign, 212 malignant)
- ▶ 10 features (given as *mean*, *standard error*, and *worst case*)
  - ▶ e.g. radius, symmetry, compactness, fractal dimension, ...

---

<sup>1</sup>[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

# Choosing the best method for prediction



Which method **generalises best** to new data, i.e. **performs class prediction well**?

## Conditional and total expected prediction error

**Recall:** To determine the optimal regression function or classifier minimize expected prediction error

$$J(f) = \mathbb{E}_{p(\mathbf{x}, y)} [L(y, f(\mathbf{x}))]$$

with respect to arbitrary functions  $f$  or a limited sub-class of functions.

- ▶ Estimate and fix  $\hat{f}(\mathbf{x}|\mathcal{T})$  from training data  $\mathcal{T} = \{(y_l, \mathbf{x}_l) : l = 1, \dots, n\}$ .
- ▶ **Conditional expected prediction error** for a fixed training set  $\mathcal{T}$

$$R(\mathcal{T}) = \mathbb{E}_{p(\mathbf{x}, y)} [L(y, \hat{f}(\mathbf{x}|\mathcal{T}))]$$

- ▶ **Note:** Training data is random too, i.e.  $p(\mathcal{T}) = \prod_{l=1}^n p(\mathbf{x}_l, y_l)$
- ▶ **Total expected prediction error**

$$R = \mathbb{E}_{p(\mathcal{T})} [R(\mathcal{T})] = \mathbb{E}_{p(\mathcal{T})} [\mathbb{E}_{p(\mathbf{x}, y)} [L(y, \hat{f}(\mathbf{x}|\mathcal{T}))]]$$

► **Training error**

$$R^{tr} = \frac{1}{n} \sum_{l=1}^n L(y_l, \hat{f}(\mathbf{x}_l | \mathcal{T}))$$

where  $(y_l, \mathbf{x}_l)$  are the samples in  $\mathcal{T}$ .

► **Test error**

$$R^{te} = \frac{1}{m} \sum_{l=1}^m L(\mathbf{y}'_l, \hat{f}(\mathbf{x}'_l | \mathcal{T}))$$

where  $(\mathbf{y}'_l, \mathbf{x}'_l)$  for  $1 \leq l \leq m$  are new samples from  $p(\mathbf{x}, y)$ .

## Common empirical error rates (II)

---

Can these empirical error rates be used to approximate total or conditional expected prediction error?

### Observations:

- ▶  $\mathcal{T}$  has already been used to determine  $\hat{f}(\cdot|\mathcal{T})$
- ▶ Training error is often smaller for more complex models (so-called **optimism of the training error**) since they can adjust better to the available data (**overfitting!**)
- ▶ How do we get new samples from the data distribution  $p(\mathcal{T})$ ? What do we do if all we have is one set of training samples?



# Splitting up the data

## Holdout method

If we have a lot of samples, **randomly split** available data into **training set** and **test set** (e.g. 75% to 25%)

## $c$ -fold cross-validation (CV)

If we have few samples

1. **Randomly split** available data into  $c$  equally large subsets  $\mathcal{F}_1, \dots, \mathcal{F}_c$ , so-called **folds**.
2. For each  $j$ 
  - ▶ Use  $c - 1$  folds, denoted by  $\mathcal{F}_{-j} = \cup_{i \neq j} \mathcal{F}_i$ , as the **training set**
  - ▶ Use fold  $\mathcal{F}_j$  as the **test set**

**Note:** No training must be done on the test set or outside of CV (see ESL Ch. 7.10.2)

## Leave-one-out cross-validation

---

CV with  $c = n$  is called **leave-one-out cross-validation (LOOCV)**.

- ▶ Popular because explicit formulas (or approximations) exist for many special cases (see ESL End of Ch. 7.10.1)
- ▶ Uses the most data for training possible
- ▶ More variable than  $c$ -fold CV for  $c < n$  since only one data point is used for testing and the training sets are very similar
- ▶ In praxis: Try out different values for  $c$ . Be cautious if results vary drastically with  $c$ .

## Approximations of expected prediction error

- ▶ Use **test error** for hold-out method, i.e.

$$R^{te} = \frac{1}{m} \sum_{l=1}^m L(y'_l, \hat{f}(\mathbf{x}'_l | \mathcal{T}))$$

where  $(y'_l, \mathbf{x}'_l)$  for  $l = 1, \dots, m$  are the elements in the test set.

- ▶ Use **average test error** for c-fold CV, i.e.

$$R^{cv} = \frac{1}{c} \sum_{j=1}^c \frac{1}{|\mathcal{F}_j|} \sum_{(y_l, \mathbf{x}_l) \in \mathcal{F}_j} L(y_l, \hat{f}(\mathbf{x}_l | \mathcal{F}_{-j}))$$

where  $\mathcal{F}_j$  is the  $j$ -th fold and  $\mathcal{F}_{-j}$  is all data except fold  $j$ .

**Note:** For the approximations to be justifiable, test and training sets need to be identically distributed

# Stratification

---

If data is unbalanced, then **stratification** is necessary.

## Examples

- ▶ Class imbalance in a classification problem

**Solution:** Sample so that each fold has the same class proportions as the original data

- ▶ Localised continuous outcome: Outcome is observed more often in some intervals than others (e.g. more high values than low values)

**Solution:**

1. Stratify outcome (divide into intervals)
2. Sample such that the relative frequency of samples from each strata (interval) in each fold is the same as in the original data

## Error estimation and tuning parameters

The holdout method and CV can be used to determine tuning parameters.

1. For a sequence of tuning parameters  $\lambda_1, \dots, \lambda_S$  calculate

$$R^{cv}(\lambda_s) = \frac{1}{c} \sum_{j=1}^c \frac{1}{|\mathcal{F}_j|} \sum_{(y_l, \mathbf{x}_l) \in \mathcal{F}_j} L(y_l, \hat{f}(\mathbf{x}_l | \lambda_s, \mathcal{F}_{-j}))$$

2. Choose

$$\hat{\lambda} = \arg \min_{\lambda_s} R^{cv}(\lambda_s)$$

Also works for a sequence of methods  $M_1, \dots, M_S$  (e.g. kNN, QDA, Logistic Regression)

## Motivating example for method selection

---

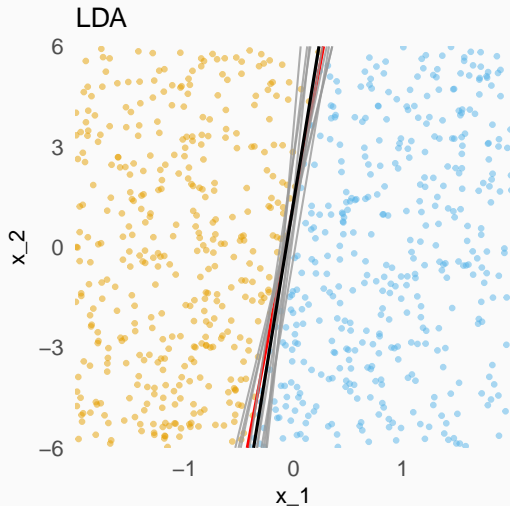
**Table 1:** Mean test error from 10-fold CV (standard deviation in parantheses)

kNN	Logistic	LDA
0.214 (0.017)	0.198 (0.017)	0.218 (0.017)

## Bias-Variance Tradeoff

---

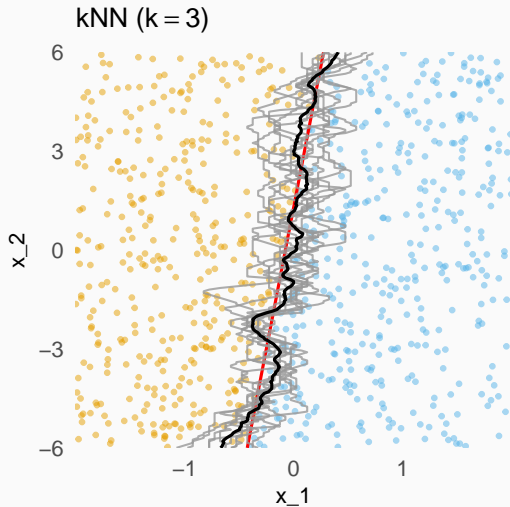
## Global rule & Simple boundary



- ▶ The red line is the true boundary.
- ▶ Each grey line represents a fit to randomly chosen 20% of all data.
- ▶ The black line is the average of the grey lines.
- ▶ Here: **low variance** and **low bias**

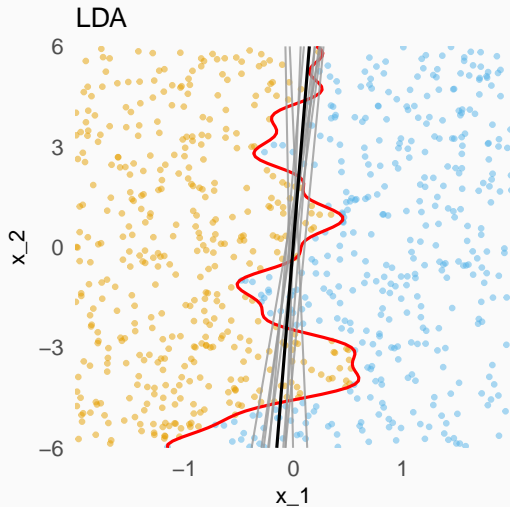


## Local rule & Simple boundary



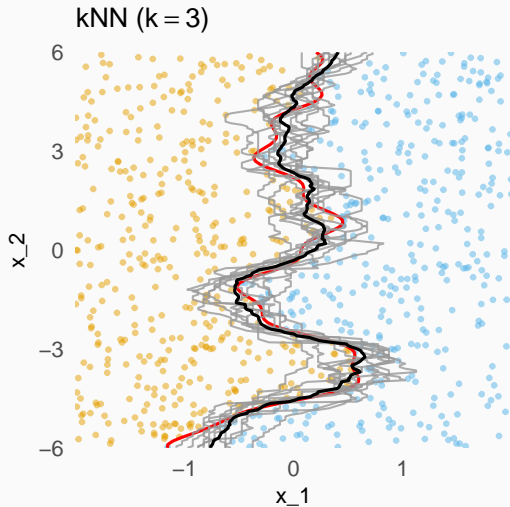
- Here: **high variance** but on average **low bias**

## Global rule & Complex boundary



- Here: **low variance** but also **large bias**

## Local rule & Complex boundary



- Here: **high variance** but on average **low bias**

## Observations

- ▶ **Local rules** are built using data in a local neighbourhood, can capture complex boundaries, but have high variance
- ▶ **Global rules** are built using all data, are usually less flexible, but have low variance
- ▶ **Bias-Variance Trade-off**: It can be theoretically motivated that bias and variance affect the expected prediction error. **The goal is to find a balance.**

# Bias-Variance Tradeoff

If  $y = f(\mathbf{x}) + \varepsilon$  where  $\varepsilon \sim N(0, \sigma^2)$ , then

$$R = \mathbb{E}_{p(\mathcal{T}, \mathbf{x}, y)} [(y - \hat{f}(\mathbf{x}))^2]$$

$$= \sigma^2$$

$$+ \mathbb{E}_{p(\mathbf{x})} \left[ \left( f(\mathbf{x}) - \mathbb{E}_{p(\mathcal{T})} [\hat{f}(\mathbf{x})] \right)^2 \right]$$

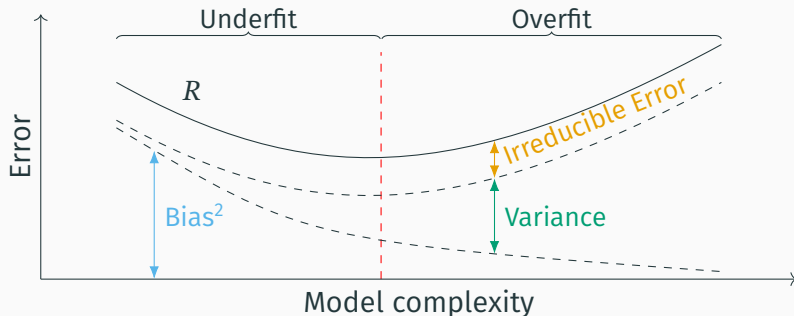
$$+ \mathbb{E}_{p(\mathbf{x})} \left[ \text{Var}_{p(\mathcal{T})} [\hat{f}(\mathbf{x})] \right]$$

Total expected prediction error

Irreducible Error

Bias<sup>2</sup> averaged over  $\mathbf{x}$

Variance of  $\hat{f}$  averaged over  $\mathbf{x}$



# Observations

- ▶ Irreducible error cannot be changed
- ▶ Bias and variance of  $\hat{f}$  are sample-size dependent
  - ▶ For a consistent estimator  $\hat{f}$

$$\mathbb{E}_{p(\mathcal{T})}[\hat{f}(x)] \rightarrow f(x)$$

for increasing sample size

- ▶ In many cases:

$$\text{Var}_{p(\mathcal{T})}(\hat{f}(x)) \rightarrow 0$$

for increasing sample size

- ▶ **Caution:** Theoretical guarantees are often dependent on the number of variables  $p$  staying fixed and increasing  $n$ . Might not be fulfilled in reality.

## Performance of LDA vs KNN

**Table 2:** Mean test error from 10-fold CV (standard deviation in parantheses)

	Boundary	
	simple	complex
LDA	0.013 (0.004)	0.091 (0.01)
kNN ( $k = 3$ )	0.021 (0.005)	0.023 (0.005)

- ▶ LDA estimates have lower variance but higher bias for complex domains
- ▶ kNN estimates can adapt locally and have low bias, but are often highly variable

## Evaluation metrics for classification

---



# What are evaluation metrics?

---

So far, we used the loss function to determine the quality of the test result.

- ▶ For regression: **Mean squared error (MSE)**
  - ▶ Low MSE ensures that the model is correct on average (given that model assumptions are correct)
- ▶ For classification: **Rate of misclassification**
  - ▶ Penalises wrong predictions across all classes, but problematic for unbalanced datasets

# Confusion matrix

		predicted outcome		total
		1	0	
actual outcome	1	True positive (TP)	False negative (FN)	TP + FN
	0	False positive (FP)	True negative (TN)	FP + TN
total		TP + FP	FN + TN	Overall total T

# Accuracy

Accuracy is defined as

$$\frac{TP + TN}{T}$$

- ▶ Measures fraction of correct predictions
- ▶ **Symmetric:** Useful if costs of false negatives and false positives are equally high
- ▶ **Weakness:** If one class is highly prevalent (unbalanced dataset), then predicting everything as the majority class can still achieve good accuracy.

		predicted outcome		total
		1	0	
actual outcome	1	True positive (TP)	False negative (FN)	TP + FN
	0	False positive (FP)	True negative (TN)	FP + TN
total		TP + FP	FN + TN	Overall total T

# Sensitivity/Recall/True positive rate (TPR)

Sensitivity is defined as

$$\frac{TP}{TP + FN}$$

- ▶ Measures fraction of correct positive predictions to all actual positive outcomes
- ▶ **Strength:** Useful if costs of false negatives are high
- ▶ **Be aware:** A predictor trained on sensitivity is likely to overpredict positive cases.
- ▶ **Typical example:** Medical test

		predicted outcome		total
		1	0	
actual outcome	1	True positive (TP)	False negative (FN)	TP + FN
	0	False positive (FP)	True negative (TN)	FP + TN
total		TP + FP	FN + TN	Overall total T

# Specificity/True negative rate

Specificity is defined as

$$\frac{TN}{TN + FP}$$

- ▶ Measures fraction of correct negative predictions to all actual negative outcomes
- ▶ **Strength:** Useful to make classifier recognize negative cases
- ▶ **Typical example:** Medical test, in balance with training on sensitivity
- ▶ False positive rate (FPR) = 1 – Specificity

		predicted outcome		total
		1	0	
actual outcome	1	True positive (TP)	False negative (FN)	TP + FN
	0	False positive (FP)	True negative (TN)	FP + TN
total		TP + FP	FN + TN	Overall total T

# Precision

Precision is defined as

$$\frac{TP}{TP + FP}$$

- ▶ Measures fraction of correct positive predictions to all positively predicted outcomes
- ▶ **Strength:** Useful if costs of false positives are high
- ▶ **Be aware:** A predictor trained on precision is likely to overpredict negative cases.
- ▶ **Typical example:** Spam filter

		predicted outcome		total
		1	0	
actual outcome	1	True positive (TP)	False negative (FN)	TP + FN
	0	False positive (FP)	True negative (TN)	FP + TN
total		TP + FP	FN + TN	Overall total T

## Combined measures (I)

- ▶  **$F_1$  score**

$$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \in [0, 1]$$

- ▶ **Matthew's correlation coefficient**

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \in (-1, 1)$$

where

- ▶  $\text{MCC} = 0$  for a random classifier
- ▶  $\text{MCC} < 0$  if worse than random and  $\text{MCC} > 0$  if better than random.
- ▶ Takes both classes into account.

- ▶ **Receiver Operating Characteristic (ROC) curve**

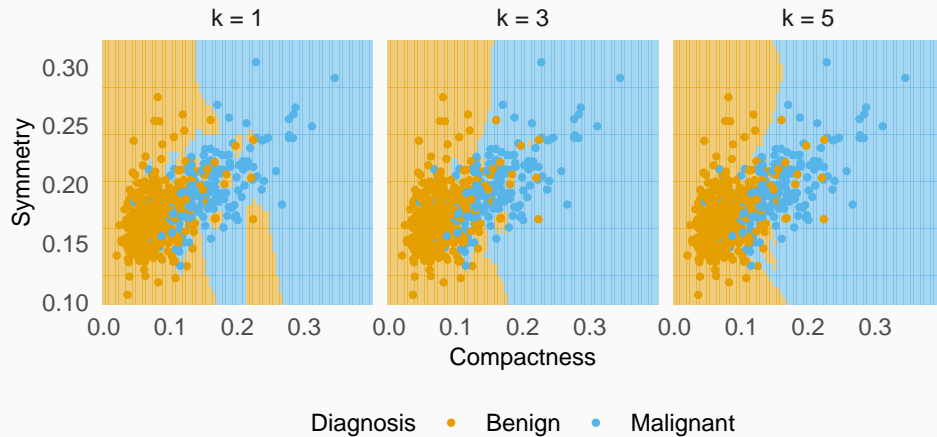
- ▶ Given a test sample and the respective estimated probabilities for the positive class, plot the trade-off between FPR and TPR.
- ▶ Diagonal line from (0, 0) to (1, 1) for a random classifier
- ▶  $\text{TPR} < \text{FPR}$  for a worse than random classifier and  $\text{TPR} > \text{FPR}$  if better than random

- ▶ **Area under the ROC curve (AUC)**

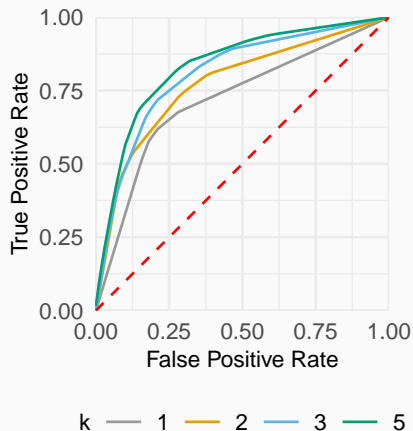
- ▶ Integral over the ROC curve
- ▶ 0.5 for a random classifier and  $> 0.5$  for better classifiers.
- ▶  $\text{AUC} \in [0, 1]$



## Choose $k$ in kNN



## Best $k$ for kNN



**Table 3:** Mean training and testing 5-fold CV errors (standard deviation in parantheses)

$k$	$R^{tr}$	$R^{cv}$
1	0 (0)	0.26 (0.018)
2	0.147 (0.007)	0.246 (0.018)
3	0.14 (0.007)	0.23 (0.018)
5	0.163 (0.008)	0.207 (0.017)

## Take-home message

---

- ▶ Cross-validation or splitting data into a training and test set are valuable approaches for model selection and model assessment
- ▶ Method complexity and global/local rules exhibit a bias-variance trade-off
- ▶ There is no single best measurement of classification quality, use multiple!