Lecture 3: Method evaluation and tuning parameter selection

Felix Held, Mathematical Sciences

MSA220/MVE440 Statistical Learning for Big Data

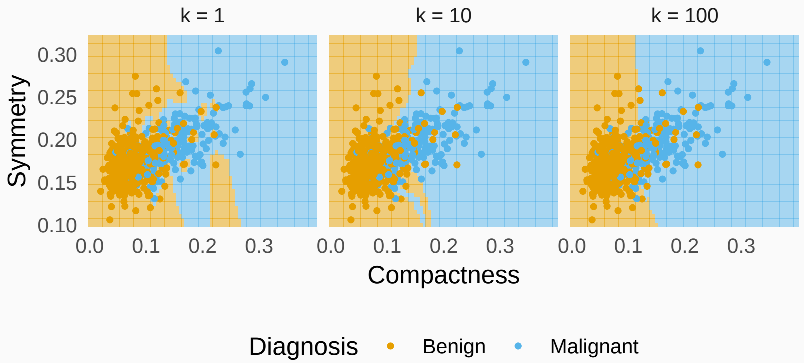
29th March 2019

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?

How to choose the best k for kNN?



- UCI breast cancer wisconsin (diagnostic) data set[1]
- Which k will do **best for class prediction** of new data?

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

Error rates (I)

- **Remember:** To determine the optimal regression function or classifier we looked at expected prediction loss

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

Note that f was thought to be an arbitrary unknown function.

- **Now:** f is estimated from data under some model assumption
- The resulting regressor/classifier $\widehat{f}(\cdot|\mathcal{T})$ is fixated after estimation but dependent on the training samples $\mathcal{T}$
- **Expected prediction error for a fixed training set $\mathcal{T}$**

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

Error rates (II)

- **Conditional expected prediction error** for a fixed training set $\mathcal{T}$

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

- Training samples are random too!
- **Total expected prediction error**

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

Empirical error rates (I)

- **Training error**

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

where

$$\mathcal{T} = \{(y_l, \mathbf{x}_l) : 1 \leq l \leq n\}$$

- **Test error**

$$R^{te} = \frac{1}{m}\sum_{l=1}^{m} L(\tilde{y}_l, \widehat{f}(\tilde{\mathbf{x}}_l|\mathcal{T}))$$

where $(\tilde{y}_l, \tilde{\mathbf{x}}_l)$ for $1 \leq l \leq m$ are new samples from the same distribution as $\mathcal{T}$, i.e. $p(\mathcal{T})$.

Empirical error rates (II)

Can we directly use these empirical rates and approximate total or conditional expected prediction error?

Observations:

- $\mathcal{T}$ has already been used to determine $\widehat{f}(\cdot|\mathcal{T})$ and usually methods aim to minimize training error
- 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 the training sample?

Splitting up the data

- **Holdout method:** If we have a lot of samples, **randomly split** available data into **training set** and **test set**
- c**-fold cross-validation:** If we have few samples
  1. **Randomly split** available data into c equally large subsets, so-called **folds**.
  2. By taking turns, use $c - 1$ folds as the **training set** and the last fold as the **test set**

Approximations of expected prediction error

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

$$R^{te} = \frac{1}{m} \sum_{l=1}^{m} L(\tilde{y}_l, \widehat{f}(\tilde{\mathbf{x}}_l | \mathcal{T}))$$

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

- Use average **test error** for c-fold cross-validation, i.e.

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

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

Careful data splitting

- **Note:** For the approximations to be justifiable, test and training sets need to be identically distributed
- **Splitting has to be done randomly**
- If data is unbalanced, then **stratification** is necessary. Examples:
  - Class imbalance
  - Continuous outcome is observed more often in some intervals than others (e.g. high values more often than low values)

Error estimation and tuning parameters

The holdout method and cross-validation can be used to determine tuning parameters.

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

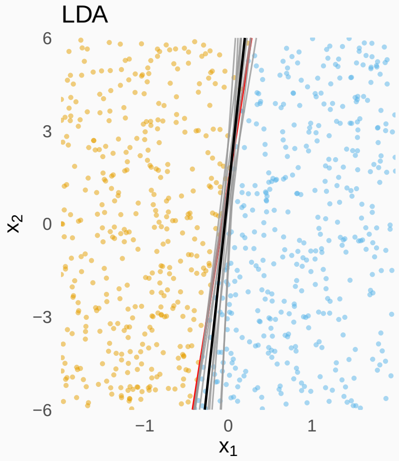
$$R^{cv}(\lambda_s) = \frac{1}{n} \sum_{j=1}^{c} \sum_{(y_l, \mathbf{x}_l) \in \mathcal{F}_j} L(y_l, \widehat{f}(\mathbf{x}_l | \lambda_s, \mathcal{F}_{-j}))$$

2. Choose

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

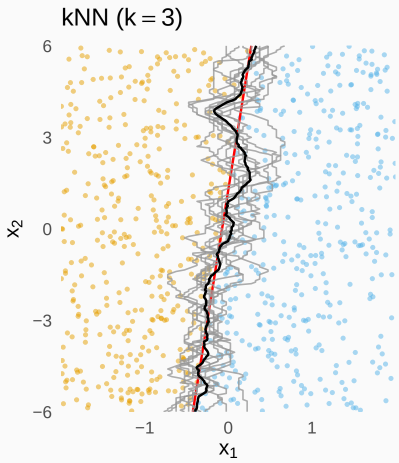
Also works for a sequence of methods $M_1, \dots, M_S$ (e.g. kNN, QDA, Logistic Regression)

Global rule & Simple boundary



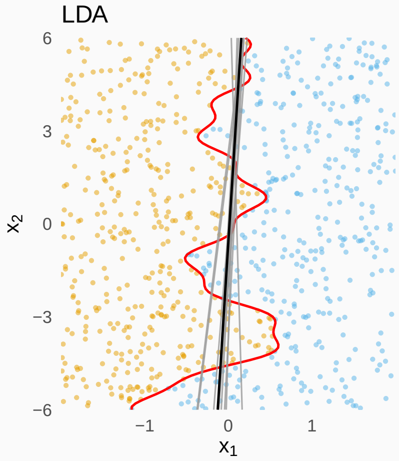
- The red line is the true boundary.
- Each grey line represents a fit to randomly chosen 20% of all data.
- The black line is the average of the grey lines.
- Here: **low variance** and **low bias**

Local rule & Simple boundary



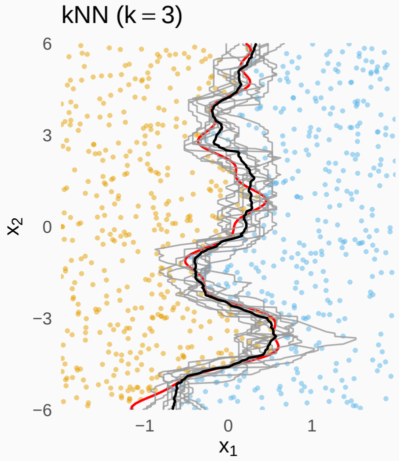
- Here: **high variance** but on average **low bias**

Global rule & Complex boundary



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

Local rule & Complex boundary



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

Global vs local rules

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**.

Performance of LDA vs KNN

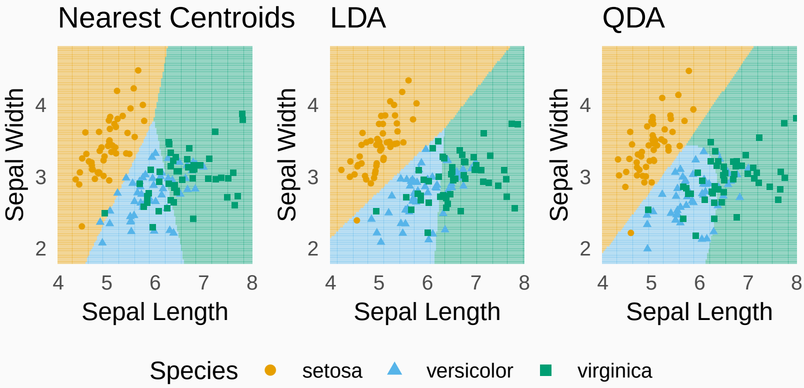
Table 1: Average cross-validation errors for ten folds

	Boundary	
	simple	complex
LDA	0.011	0.092
kNN (k = 3)	0.018	0.021

LDA does better for simple boundaries, while kNN has an advantage for more complicated boundaries.

Choosing a classification method (I)

Remember: We looked at different classification methods for solving the same classification problem



Choosing a classification method (II)

Table 2: Average cross-validation errors for ten folds

NC	LDA	QDA
0.193	0.2	0.22

Quality of a classification result

How to quantify classification quality, When we receive a classification result from our classifier?

Setting:

- Language/notation comes from medical studies where the presence or absence of a disease/condition is determined
- Binary classification with classes 0 and 1
- 0s are interpreted as **negative outcomes** (e.g. not sick = healthy individual) and 1s are interpreted as **positive outcomes** e.g. sick individuals

Confusion matrix

Table 3: Confusion matrix

Predicted class	**True class**	
	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Measures of classification quality

- **Accuracy:** $\frac{TP + TN}{TP + FP + FN + TN}$
- **Precision:** $\frac{TP}{TP + FP}$
- **Sensitivity/True positive rate (TPR)/Recall:** $\frac{TP}{TP + FN}$
- **Specificity:** $\frac{TN}{TN + FP}$
- **False positive rate (FPR)/fall out:** 1 - Specificity

Combined measures

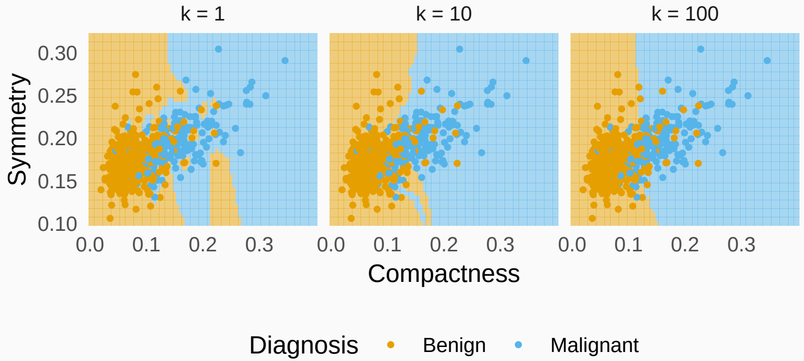
- **F_1 score** $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
- **Matthew's correlation coefficient:**

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

  where $MCC = 0$ for a random classifier and $MCC < 0$ if worse than random and $MCC > 0$ if better than random. Takes both classes into account.
- **Receiver Operating Characteristic (ROC) curve**: Trade-off between FPR and TPR. Equal for a random classifier, TPR < FPR for a worse than random classifier and FPR > TPR is better than random
- **Area under the ROC curve (AUC)**: 0.5 for a random classifier and > 0.5 for better classifiers. Maximum 1.

How to choose the best k for kNN? (revisited, I)

Reminder: This motivated our discussion



How to choose the best k for kNN? (revisited, II)

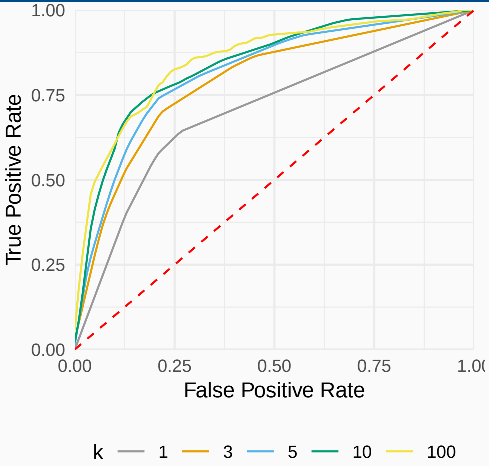


Table 4: Average training and cross-validation errors for five folds

k	R^{tr}	R^{cv}
1	0.000	0.276
3	0.137	0.243
5	0.160	0.228
10	0.182	0.204
100	0.204	0.207

$k = 100$ leads to the best measurable results. Judging from the plots for $k = 1$, $k = 10$ and $k = 100$, kNN is trying to approximate a linear decision boundary and "tries to become a global method".

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!