Applied Machine Learning
in Health Sciences
2023

Exercises

This document will be updated with additional checkpoints for each course day throughout the course.

Rev: 16 Jan 2023.

1 Introduction

Checkpoint 1: ✍

Think of some real-life applications for *supervised learning* from your research field. Describe two applications/research questions in which classification or regression might be useful. Describe the predictors/input variables as well as the response/output variable, data set size, number of variables etc. Is the goal of each application *inference* or *prediction*? Explain your answer.

Checkpoint 2: ✍

Think of some real-life applications for *unsupervised learning* from your research field. Describe two applications/research questions in which unsupervised learning might be useful. Describe the data variables, data set size, number of variables etc. What is the goal of the analysis? Explain your answer.

2 Linear regression

Checkpoint 3: ✍

Do section 3.7 conceptual exercise 4. in ISL (ISL page 122).

Checkpoint 4: ⌨) Linear regression - body density data
In this checkpoint you will analyse the body density data using linear regression. The goal is to build a linear regression model to predict *body density* from *chest circumference measurements*.

- Open the script `main1a.m` and review it to understand the different steps in the script. Some of the code needed to answer this checkpoint is already provided in the script, but you also need to do a bit of coding yourself.

Use the script `main1a.m / main1a.R` and your code to answer the following:

(a) Load the data set `bodyMeasurementsSingleTrainTest.txt`. Describe the data, what are the sizes of the training- and test sets? how many observations and features? what is the numerical range of the variables?

(b) Identify the part for of the script that is used for creating the polynomial expansion of the input variable. Try do create polynomial expansions with different polynomial order from 1 to 7, how many columns/features are there in the resulting data matrices?

(c) Part of the code is scaling the individual columns of the polynomial regressors. Explain how this scaling is done and why the scaling may be necessary (Hint: what is the numerical range of the individual columns?).

(d) Type `doc fitglm` in Matlab or `?glm` in R and use a bit of time to familiarize yourself with this function. What are the inputs to the function and what is the output?

(e) Explain how training error and test error are quantified in the script?

(f) Run the analysis with polynomial order ranging from 1 to 7 (Hint: include a for-loop in the script). For each of these seven models, make a plot of body density (ordinate) vs. chest circumference (abscissa) for the training- and test data as well as the model's predictions (all three in the same plot). Include the plots in your report and describe/discuss the plots.

(g) Write your own code to make a plot of training error and test error vs. polynomial order (order 1 to 7). Include the plot in your report and describe/discuss the plot. Do you observe severe overfitting for some polynomial orders?

3 Bias-variance decomposition

Checkpoint 5: ✍ **Training- and test errors**

(a) Explain the difference between a training- and a test data. Also explain the difference between training- and test error. (b) Argue why we typically are interested in good model performance in terms of low test error rather than in terms of low training error.

Checkpoint 6: ✍

Do section 2.4 conceptual exercise 3. in ISL (ISL page 53).

4 Model evaluation and resampling

Checkpoint 7: ✍

Explain, in your own words, what a training set is, a validation set is, and a test set is? Why is this partition of data needed? Explain how k-fold cross validation is implemented. Explain how leave-one-out (LOO) cross validation is implemented. What are the advantages and disadvantages of k-fold cross validation relative to the validation set approach and the LOO cross validation approach?

Checkpoint 8: ⌨

Matlab users: By using the Matlab function `cvpartition` we can create random partitions of data.

- Type `doc cvpartition` and use a bit of time to familiarize yourself with this function.

(a) Run the command `c = cvpartition(20,'Holdout',0.2)` and explain what the function does. Also type `c.training` and `c.test` and describe the variables contained in these fields.

(b) Run the command `c = cvpartition(20,'KFold',10)` and explain what the function does. Also type `c.training(1)` and `c.test(1)`, `c.training(2)` and `c.test(2)`, `c.training(3)` and `c.test(3)` etc. and describe the variables contained in these fields.

(c) Explain why it is generally recommended to run e.g. the command `rng('default')` before creating the random partitions.

R users: By using the functions `createDataPartition` and `createFolds` we can create random partitions of data.

- Look at the help text for these functions and use a bit of time to familiarize yourself with the functions.

(a) Run the command `c = createDataPartition(y = 1:20, p = 0.8)` and explain what the function does. Also describe the content of the variable `c`.

(b) Run the command `c = createFolds(y = 1:20, k = 10, returnTrain = TRUE)` and explain what the function does. Also explain the content of the variable `c`.

(c) Explain why it is generally recommended to run e.g. the command `set.seed(0)` before creating the random partitions.

Checkpoint 9: ⌨) Linear regression - body density data - cross validation
In this checkpoint you will analyse the body density data using linear regression (with polynomial regressors), and the goal is to build a linear regression model to predict *body density* from *chest circumference measurements*. You will use k-fold cross validation to evaluate model performance for different polynomial order.

- Open the script `main1b.m` and review it to understand the different steps in the script. Some of the code needed to answer this checkpoint is already provided in the script, but you also need to do a bit of coding yourself.

Use the script `main1b.m / main1b.R` and your code to answer the following:

(a) Load the data set `bodyMeasurementsSingleCV.txt`. Describe the data, what is the size of the data set? how many observations and features?

(b) Identify the part of the script where the random partition is performed. How many folds are used in the k-fold cross validation?

(c) The script contains two for-loops (one nested within the other). Explain, in your own words, how the analysis is performed/structured in the script.

(d) Run the script. Look at the plot of training and test error (cross-validation error) (ordinate) vs. polynomial order (abscissa) (both error curves in same plot). Include the plot in your report and describe/discuss it. Are the curves as expected? do you observe severe overfitting (compare with your result from

Checkpoint 4)? if not, try to explain why not (Hint: look at the number of training observations and model flexibility). For which polynomial order do you observe the lowest test error?

5 Logistic regression

Checkpoint 10: ✍/ ⌨ The logistic regression model
Suppose that your input data x_i has a single predictor x_{i1}, and suppose that the logistic regression model has parameters $\beta_0 = 1$ and $\beta_1 = 1$.

(a) Make a drawing/plot with curves of i) the *posterior probability* of class 0 $P(y = 0|x_i)$ as a function of x_{i1} and ii) the *posterior probability* of class 1 $P(y = 1|x_i)$ as a function of x_{i1}, with x_{i1} ranging from -6 to 6. Remember to label each of the curves, to label axes in your drawing, and also remember to put tick labeling (numeric) on the axes.

(b) Explain how you can classify a given input x_i by using the posterior probabilities, and indicate the decision boundary/threshold on your drawing/plot above.

(c) Make another drawing/plot with the *log-odds* ratio as a function of x_{i1}.

(d) Explain how you can classify a given input x_i by using the log-odds ratio, and indicate the decision boundary/threshold on your drawing/plot above.

(e) Suppose that we have three test samples

Sample number i	Predictor value x_{i1}
1	-2
2	0
3	2

For each of these three samples, compute $P(y = 1|x_i)$ and compute the predicted the class label.

Checkpoint 11: ⌨ Logistic regression - CSF biomarker data
In this checkpoint you will analyse the CSF biomarker data using logistic regression. The goal of the analysis is to build a logistic regression model to predict *group membership* (control/impaired) from a single CSF feature *tau*.

- Open the script `main2a.m / main2a.R` and review it to understand the different steps in the script.

Use the script to answer the following:

(a) Run the first code section `%% Import data etc.`. What is the size of the data set? How many features and observations? Describe the response variable `y`, what type of variable is it and what is its content? How many subjects are there in each group?

(b) Explain how data is divided into a training set and a test set, and explain the meaning of *stratification*. Run the second code section `%% Divide into training and test sets`. Compute the class proportions in the training set and in the test set and report these, and verify that class proportions are preserved after the data partitioning.

(c) Run the code section `%% Train model, predict, and plot model`. Describe the variable `catInfo` and its content. Describe the model outputs `yhatTrainProb` and `yhatTestProb`, what does these represent? Describe the variables `yhatTrain` and `yhatTest`, what do these represent? Include the plot of model output vs. input, describe the plot, and explain how classification can be performed based on this plot.

(d) In the third code section, the model predictions are converted to categorical data to be used as input to the `confusionchart` (Matlab) `confusionMatrix` (R) function for plotting the confusion matrix. Look at the help text for this function and use a bit of time to familiarize yourself with the function. Run the forth code section `%% Plot confusion matrix`, include the plot in your report, and describe/discuss it. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: *classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.*

(e) You will now do almost the same analysis, but you will now use 10-fold cross-validation for model evaluation.

Open the script `main2b.m / main2b.R` and review it to understand the different steps in the script.

Run the script. Include the plot of the confusion matrices for the training- and test/validation data in you report and describe/discuss it. Also explain how the confusion matrices are computed across the cross-validation iterations. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: *classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.*

6 Regularization

In *linear regression* we can find coefficient estimate by minimizing the residual sum of squares *cost function*

$$RSS\left(\beta\right) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2, \tag{1}$$

which quantifies the agreement/mismatch between between the observed responses and the model's predictions.

In many real-world analysis tasks it is common practice to add a *penalty/shrinkage* term, $J(\beta)$, to the cost function, so that the optimization objective becomes

$$RSS\left(\beta\right) + \lambda J(\beta). \tag{2}$$

Popular choices for the penalty term are the *ridge* or ℓ_2 penalty

$$J(\beta) = \sum_{j=1}^{p} \beta_j^2 \tag{3}$$

and the *lasso* or ℓ_1 penalty

$$J(\beta) = \sum_{j=1}^{p} |\beta_j|. \tag{4}$$

In *logistic regression*, given a set of training samples $\{x_i, y_i\}_{i=1}^n$, the *likelihood function* can be written as[1]

$$\ell\left(\beta\right) \quad = \quad \prod_{i:y_i=1} P(y=1|x_i) \prod_{i:y_i=0} \left(1 - P(y=1|x_i)\right) \tag{5}$$

The likelihood function quantifies the agreement between the predicted posterior probabilities e.g. $P(y=1|x_i)$ and the observed/true labels y_i, and we can estimate optimal model parameters β by maximizing the likelihood function $\ell\left(\beta\right)$ above. However, instead of *maximizing* the likelihood function, with respect to the model parameters β, it is common practice to *minimize* the *negative log-likelihood function*

$$-\log\left\{\ell\left(\beta\right)\right\} \tag{6}$$

This is usually done for mathematically and numerical convenience. Since the logarithm is a monotonically increasing function we still maximize the likelihood if we maximize

[1] This formula is mathematically equivalent to eqn. (4.5) in the ISL textbook. $P\left(y=1|x_i\right)$ corresponds to $p(x_i)$ in ISL eqn. (4.5)

the log-likelihood instead. Furthermore, *minimizing* the negative log-likelihood function is equivalent to *maximizing* log-likelihood function. Hence, we can interpret the negative log-likelihood function as an *error function*, that quantifies the disagreement between the model's predictions/posterior probabilities and the observed data/true class labels.

In many real-world analysis tasks it is common practice (as for linear regression) to add a *penalty* term, $J(\beta)$, to the cost function, so that the optimization objective becomes

$$-\log\left\{\ell\left(\beta\right)\right\} + \lambda J(\beta). \tag{7}$$

Checkpoint 12: ✍ Regularization

(a) Explain when and why it may be necessary to use model regularization.

(b) (1) Write down the *penalized* cost function for linear regression for each of the following penalty/shrinkage terms: (i) ridge (ℓ_2), (ii) lasso (ℓ_1).

(2) Explain meaning of the different elements of the expressions in (1).

Checkpoint 13: ✍ Regularization, training- and test errors, and bias-variance trade-off

(a) Provide a sketch of how coefficient estimates typically change with the strength of the regularization parameter λ for the ridge and the lasso penalty, respectively. Describe/discuss the curves and their similarities/differences.

(b) Provide a sketch of typical training error and test error on a single plot, as a function of the regularization parameter λ. λ should be on the x-axis, and the y-axis should represent the values for each curve. Make sure to label each one.

(c) Explain why each of the two curves has the shape displayed in (a).

(d) Explain how model complexity changes with λ, and discuss your answer in terms of the bias-variance trade-off.

(e) Explain how the suitable regularization strength is chosen in a real-world analysis.

You will now run analyses in which you will analyse the body density data using linear regression with ridge regularization and analyse the CSF biomarker data using logistic regression with lasso regularization, respectively. Note that we could also have used linear regression with lasso regularization and logistic regression with ridge regularization, respectively.

Using the Matlab function `lassoglm` or the R function `glmnet` we can fit regularized linear regression models and regularized logistic regression models. Have a look at the help text for this function and use a bit of time to familiarize yourself with the function. What are the input parameters *alpha* and *lambda* controlling? What are the `lassoglm` functions inputs and outputs?

Checkpoint 14: ⌨) Linear regression - body density data - ridge regularization
In this checkpoint you will analyse the body density data using linear regression, and the goal is to build a ridge regularized linear regression model to predict *body density* based on subjects' *age*, *height*, *weight*, and 10 *circumference* measurements (13 input features in total). You will use k-fold cross-validation to evaluate model performance for different regularization strengths.

- Open the script `main3a.m / main3a.R` and review it to understand the different steps in the script.

Use the script and your code to answer the following:

(a) Load the data set `bodyMeasurements.txt`. Describe the data, what is the size of the data set? how many observations and features?

(b) Identify the part of the script where the random partition is performed. How many folds are used in the k-fold cross-validation?

(c) Identify the part of the script where the range of the regularization parameter λ is defined. Which sequence of λ-values is used?

(d) The script contains two for-loops (one nested within the other). Explain, in your own words, how the analysis is performed/structured in the script.

(e) Identify the lines where data is being *standardized* in the script. Explain how standardization is done, and why it is generally recommended to standardize data when using shrinkage regularization.

(f) The fitting function has a parameter `alpha`. Explain what this parameter is?

(g) Run the analysis. Plot the training error and the test error as a function of the regularization parameter λ. Include the plot in your report and describe/discuss it.

(h) How would you choose the "best" model? For the selected model report the training and test error.

(i) Look at the β coefficient array. What is the dimensionality of β?

(j) Include the plot with coefficient traces as a function of λ in your report and describe/discuss it. What happens with coefficients with decreased model complexity/regularization strength? Are any coefficients exactly zero?

(k) Look at the coefficients for your chosen model. Identify the most important coefficients to the model.

Checkpoint 15: ⌨ Logistic regression - CSF biomarker data - lasso regularization

In this checkpoint you will analyse the CSF biomarker data using logistic regression with lasso regularization. The goal of the analysis is to build a logistic regression model to predict *group membership* (control/impaired) based on 130 CSF features.

- Open the script `main3b.m / main3b.R` and review it to understand the different steps in the script.

Use the script to answer the following:

(a) Load the data set `csfBiomarkers.txt`. Describe the data, what is the size of the data set? how many observations and features?

(b) Identify the part of the script where the random partition is performed. How many folds are used in the k-fold cross-validation?

(c) Identify the part of the script where the range of the regularization parameter λ is defined. Which sequence of λ-values is used?

(d) The script contains two for-loops (one nested within the other). Explain, in your own words, how the analysis is performed/structured in the script.

(e) Describe meaning of the fitting function's parameter `alpha`.

(f) Run the analysis. Plot the training error and the test error as a function of the regularization parameter λ. Include the plot in your report and describe/discuss it.

(g) How would you choose the "best" model? For the selected model report the training and test error.

(h) Look at the β coefficient array. What is the dimensionality of β?

(i) Include the plot with coefficient traces as a function of λ in your report and describe/discuss it. What happens with coefficients with decreased model complexity/regularization strength? Are any coefficients exactly zero?

(j) Look at the coefficients for your chosen model. Identify the most important coefficients to the model.

(k) Plot the confusion matrix for your chosen model. Include it in your report and describe/discuss it. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: *classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.*

7 Support vector machines

Checkpoint 16: ✍ **maximal margin classifier**

(a) Do section 9.7 conceptual exercise 3. in ISL (ISL page 399).

Checkpoint 17: ✍ **Support vector classifier** (aka. soft margin SVM)

(a) Discuss the difference between the maximal margin classifier (hard margin SVM) and the support vector classifier (soft margin SVM).

(b) Describe why it may be necessary or beneficial to use the soft margin SVM instead of a hard margin SVM.

(c) What are slack variables ϵ?

(d) Discuss the meaning of the regularization parameter C.

(e) On your sketch from the previous checkpoint, draw an example of a soft margin.

(f) Mark data points with positive slack variables ϵ.

(g) Sketch what happens to the canonical hyperplanes with increases in C and decreases in C, and discuss how this is related to the bias-variance trade-off.

Checkpoint 18: ✍ **Regularization**

(a) Provide a sketch of typical training error and test error on a single plot, as a function of the regularization parameter C. C should be on the x-axis, and the y-axis should represent the values for each curve.

(b) Explain why each of the two curves has the shape displayed in (a) (Hint: see g in the previous checkpoint).

(c) Explain how a suitable amount of regularization is chosen in a real-world application.

Checkpoint 19: ✍ **Support vector machine** (aka. kernel SVM or non-linear SVM)

(a) Describe a scenario where use of a non-linear SVM may lead to better performance relative to a linear classifier.

(b) What is a *kernel* function?

(c) Describe what an evaluation of the kernel function corresponds to.

Checkpoint 20: ⌨ **Support vector machine - CSF biomarker data**
In this checkpoint you will analyse the CSF biomarker data using a support vector classifier (soft-margin SVM). The goal of the analysis is to build a support vector machine to predict *group membership* (control/impaired) based on 130 CSF features.

- Open the script `main4a.m / main4a.R` and review it to understand the different steps in the script.

Use the script to answer the following:

(a) Load the data set `csfBiomarkers.txt`. Describe the data, what is the size of the data set? how many observations and features?

(b) Identify the part of the script where the random partition is performed. How many folds are used in the k-fold cross-validation?

(c) Identify the part of the script where the range of the regularization parameter C is defined. Which sequence of C-values is used?

(d) The script contains two for-loops (one nested within the other). Explain, in your own words, how the analysis is performed/structured in the script.

(e) The function `fitcsvm` (Matlab) `svm` (R) is here used to fit the SVM. Describe the input parameter `C`.

(f) Run the analysis. Plot the training error and the test error as a function of the regularization parameter C. Include the plot in your report and describe/discuss it. Also discuss the curves in terms of the bias-variance trade-off. How would you choose the "best" model? For the selected model report the training and test error.

(g) Look at the β coefficient array. What is the dimensionality of β?

(h) Include the plot with coefficient traces as a function of C in your report and describe/discuss it. What happens with coefficients with decreased model complexity/regularization strength? Are any coefficients exactly zero? Look at the coefficients for your chosen model. Identify the most important coefficients to the model.

(i) Include the plot with the number of support vectors as a function of C in your report and describe/discuss it.

(j) Plot the confusion matrix for your chosen model. Include it in your report and describe/discuss it. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: *classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.*

8 Neural networks

Checkpoint 21: ✍ Neural networks vs. linear- and logistic regression

(a) Discuss limitations of linear regression and logistic regression and possible advantages of neural networks.

Checkpoint 22: ✍ Design a neural network
Think of a classification task where the feature dimensionality is *two* and there are *two* possible classes (binary classification). The aim of this checkpoint is to draw a graphical representation of a NN with *one* hidden layer containing *three* hidden nodes.

(a) Begin with drawing input nodes, how many input nodes?

(b) Add hidden nodes to your sketch and draw connections between input nodes and the hidden nodes.

(c) Add a bias node your sketch and draw connections between this bias node and the hidden nodes.

(d) Add two output nodes to the sketch. Draw connections between hidden nodes and the output nodes.

(e) Add another bias node your sketch and draw connections between this bias node and the output node.

(f) How are the inputs to each of the hidden nodes computed? Write down the formula for the input to a hidden node.

(g) Explain what a hidden node activation function is. Write down the formula for a *ReLU* activation function and make a sketch of hidden node output/activation A_k as a function of its input z_k.

(h) How are the inputs to the output nodes computed. Write down the formula for the inputs to the output nodes.

(i) We will use the *softmax* activation function for the model's output. Describe what the network's two outputs represent. What is the numerical range of the outputs, and how can the network's outputs be interpreted?

(j) Can this network produce non-linear decision boundaries? (Hint: Consider whether linear or non-linear activation functions are used and the network architecture?).

Checkpoint 23: ✍ Neural networks - regularization

(a) Explain when and why it may be necessary to use model regularization in neural networks.

(b) Write down a cost function comprising the error function and a penalty term that penalizes the squared weight values (Hint: See equations 10.14 and 10.31 in ISL). Explain the meaning of two terms in this expression.

(c) Compare the expression in **(b)** with the cost function in ridge regularized logistic regression.

(d) Explain how a suitable value for the regularization parameter λ can be chosen.

Checkpoint 24: ⌨ Neural network - CSF biomarker data

In this checkpoint you will analyse the CSF biomarker data using a feed-forward neural network. The goal of the analysis is to build a neural network to predict *group membership* (control/impaired) based on 130 CSF features.

- Open the script `main5a.m / main5a.R` and review it to understand the different steps in the script.

Use the script to answer the following:

(a) Load the data set `csfBiomarkers.txt`. Describe the data, what is the size of the data set? how many observations and features?

(b) Describe the neural network architecture. How many hidden nodes and hidden layers?

(c) Make a sketch showing the structure of this neural network. What is the number of parameters in the neural network?

(d) Identify the part of the script where the random partition is performed. How many folds are used in the k-fold cross-validation?

(e) Identify the part of the script where the range of the regularization parameter λ is defined. Which sequence of λ-values is used?

(f) The script contains two for-loops (one nested within the other). Explain, in your own words, how the analysis is performed/structured in the script.

(g) Run the analysis. Plot the training error and the test error as a function of the regularization parameter λ. Include the plot in your report and describe/discuss it. Also discuss the curves in terms of the bias-variance trade-off. How would you choose the "best" model? For the selected model report the training and test error.

(h) Can we interpret the weights/coefficients of a neural network in the same way as we did for logistic regression? Can we identify important features by looking at the weights directly?

(i) Plot the confusion matrix for your chosen model. Include it in your report and describe/discuss it. Based on the numbers in the confusion matrix, compute and report the following performance metrics for the test set: *classification accuracy, error rate, true positive rate, true negative rate, false positive rate, false negative rate, sensitivity, and specificity.*

Checkpoint 25: ✍ Neural networks, challenges

(a) Discuss challenges with neural networks.

9 PCA

Principal Component Analysis (PCA) is a popular method for dimensionality reduction. Given an $n \times p$ dataset $\mathbf{X}$, PCA computes principal components $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots \boldsymbol{\phi}_M$ which are the M (normalized and orthogonal) directions of maximal variance in the data. Having computed the principal components, one can compute an M-dimensional embedding $z_i \in R^M$ from a point $x_i \in R^p$:

$$z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}. \tag{8}$$

We can also reconstruct an approximation of x_i from z_i, using the principal components:

$$x_{ij} \approx \sum_{m=1}^{M} z_{im} \phi_{jm}. \tag{9}$$

Finally, we can quantify how big a proportion of the total variance in the data which is captured by the m'th principal component $\boldsymbol{\phi}_m$ by

$$PVE_m = \frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2} = \frac{\sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}. \tag{10}$$

Checkpoint 26: PCA - using principal components
Suppose that we have the following five data points in R^3 (3-dimensional data)

	x_1	x_2	x_3	x_4	x_5
x_{i1}	-2	-2	0	2	2
x_{i2}	-2	2	0	-2	2
x_{i3}	-4	0	0	0	4

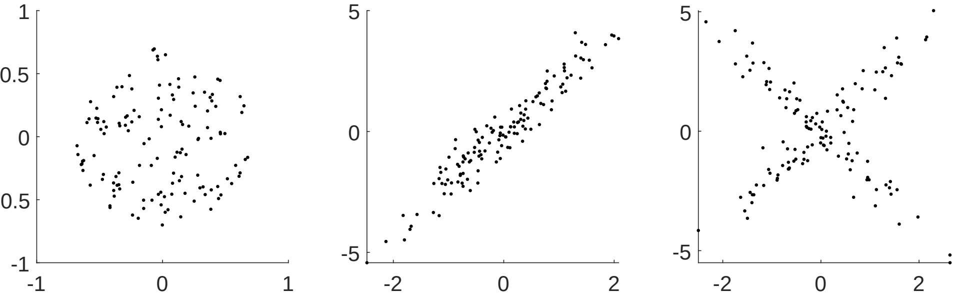
The variables in this data set has zero mean. Principal component analysis of the data set has provided the following three vectors $\boldsymbol{\phi}_1 = (0.41, 0.41, 0.82)^T$, $\boldsymbol{\phi}_2 = (-0.71, 0.71, 0.00)^T$ and $\boldsymbol{\phi}_3 = (0.58, 0.58, -0.58)^T$.

(a) Calculate and report the principal component scores z_{i1}, z_{i2}, and z_{i3} for each of the five data points.

(b) Calculate and report how big a proportion of the total variance is captured by each of the individual individual principal components (PVE_1, PVE_2, and PVE_3). Create a scree plot from these and describe and comment on the plot.

(c) Make a plot of the five data points embedded into the 2-dimensional subspace that accounts for most of the variance in the data.

(d) From the embeddings in **(c)** compute the reconstructions of the embedded points (eqn. (9)). Compare these reconstructions with the original data points and discuss the quality/accuracy of the reconstruction.



PCA is a *linear* method, which means that every principal component is a linear combination of the features, as indicated by eqn. (8). One might expect that this has some limitations. We will investigate such limitations here.

Checkpoint 27: Limitations of PCA

(a) In the figure above, we have plotted three different 2-dimensional data sets. Make a sketch/drawing of how you expect the *scree* plot to look for each of the three data sets. Include the drawing in your report and explain your answer.

(b) For which of the data sets will the proportion of explained variance be largest for the first principal component? Explain your answer.

(c) For which of the data sets would dimensionality reduction by PCA make most sense? Explain your answer.

Checkpoint 28: PCA - body density data

The body density data set has 252 observations and 13 features To explore the data, we could do 2-dimensional scatterplots of the data, each of which contains the 252 observations' measurements plotted according two of the features. However, plotting all combinations of two different features would give $\binom{13}{2} = \frac{13*(13-1)}{2} = 78$ plots, which quickly becomes overwhelming.

Instead, we can use PCA to project the data onto a low-dimensional subspace that captures most of the variance in the data. E.g. a 2-dimensional space spanned by the two principal components that accounts for most of the variance and then plot the data in that space. Open the script `main6a.m / main6a.R` and review it to understand the different steps in the script.

(a) Load the data, and write your own code to make scatter plots of a few feature-pairs. Try to find examples "interesting" feature pairs, include the corresponding scatter plots in your report and describe/discuss these.

(b) Use the script to create a correlation plot showing the correlation between individual feature pairs. Include the plot in your report, describe the plot and discuss the general correlation structure between features.

(c) Use the script to generate a figure showing the standard deviation of individual features. Describe and discuss the plot. Explain the impact of large differences in standard deviation across features, and explain why it may be preferable to scale individual features before PCA.

(d) Use the script to make scatter plots of feature pairs with lines representing the PCA axes. Run the code for `featureIdx = [6 13];`, include the plots in your report and describe/discuss the plots. Does standardization influence the result? Also report and discuss the percentage variance explained for the standardized vs. non-standardized data. Also try to explore a few other combinations of feature pairs, include the plot in your report and discuss the results.

(e) Use the script to run PCA on data set with all 13 features and to create a third figure with three subplots. Include the plots in your report and describe what the plots show. Why are there two different plots of the first two principal components? Describe their differences and discuss the advantage/disadvantage of standardization.

(f) Discuss the limitations of representing the data in 2-dimensional scatter plots. (Hint: consider the amount of variance explained). Argue whether PCA is suitable for providing a low dimensional representation of this particular data set (Hint: look at the third plot in the figure from **(e)** above).

10 K-Means Clustering

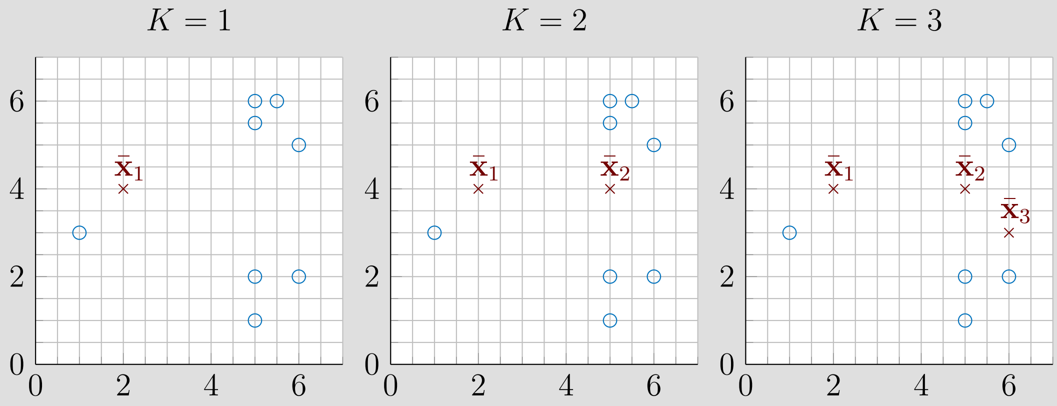
K-Means is an iterative clustering algorithm that partitions a dataset into K non-overlapping clusters $C_1, \dots, C_K$. The partitioning is done to minimize the objective

$$\min_{C_1,\dots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{i,j} - x_{i',j}\right)^2 \right\} = \min_{C_1,\dots,C_K} \left\{ 2 \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \left(x_{ij} - \bar{x}_{kj}\right)^2 \right\}, \quad (11)$$

where $\bar{x}_{kj}$ is the k'th cluster mean, i.e., $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i \in C_k} x_{ij}$. Recall that the algorithm has two steps (after initialization), which are repeated until the clustering no longer changes:

1. Assign each observation to the cluster whose centroid is closest.
2. For each of the K clusters, compute the cluster centroids $\bar{\mathbf{x}}_k$.

Checkpoint 29: K-Means computations
Consider the following three initializations of the K-Means algorithm:



(a) Apply the K-means algorithm in the three examples $K = 1$, $K = 2$, and $K = 3$ above. For each example, you should start from the cluster centroids $\bar{\mathbf{x}}_k$. In your report, you should include plots showing the position of cluster centers for each iteration of steps 1. and 2. above as the algorithm proceeds.

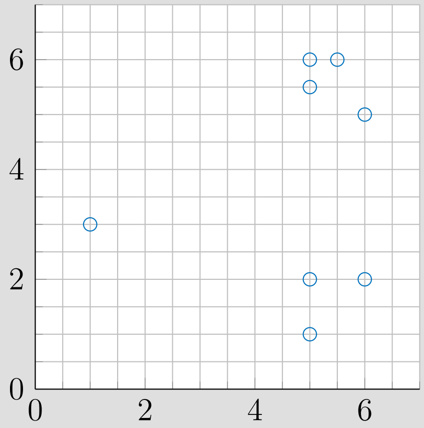
(b) Discuss the final result of the K-Means clustering for the three examples $K = 1$, $K = 2$, and $K = 3$ above. For each of the three examples, explain whether K-Means clustering was successful in discovering clusters/structure in the data.

(c) Are there any *outlier points* in the data set. Describe how outliers can impact the clustering result, and explain why it is generally recommended to run K-Means clustering several times with random initial configurations (initial centroids).

(d) Discuss how to choose an suitable value of K in a real-world analysis scenario.

11 Hierarchical Clustering

Checkpoint 30: Hierarchical clustering computations:
Considering the same data as for K-Means. You will here do hierarchical clustering using *Eucledian distance* as dissimilarity measure.



(a) Draw the dendrogram for the data set using *single linkage*. It is okay to only look at the points and not compute the actual/numerical distances.

(b) Cut the dendrogram to get two clusters. Do the clusters match with the two clusters you found in the previous exercise with K-Means?

(c) Draw the dendrogram for the data set using *complete linkage*. It is okay to only look at the plot and not compute the actual distances.

(d) How do the two dendrograms in **(a)** and **(c)** differ?

Checkpoint 31: Hierarchical clustering on CSF biomarker data
You will now apply hierarchical clustering on the CSF biomarker data set. In the analysis you will cluster the *features* (instead of clustering observations). Hence, you will try to find groups of features that are similar across observations.

(a) Open the script `main7a.m / main7a.R`. What does it do? In particular, which distance measure and what linkage is used to build the dendrogram?

(b) The script creates a plot of feature standard deviations. Include the plot in you report and describe/discuss it. Explain why it may be preferable to standardize features in cluster analysis.

(c) Alter the code to make plots of dendrograms with the three different linkages *single*, *complete*, and *average*.

(d) Include the plots in **(c)** in your report and describe/discuss the plots. How do the three plots differ? Which of the three linkage methods would you prefer to use?

(e) For each plot in **(c)** indicate where you will cut the dendrograms in order to obtain clusters. How many clusters do you obtain at the chosen cuts.

Checkpoint 32: Clustering, practical issues

(a) Discuss pros and cons of K-Means and hierarchical clustering as well as practical issues.

- -
PMR, pmr@cfin.au.dk, Jan 2023.