Machine Learning I

Retry Exam

Prof. Dr. Dr. Lars Schmidt-Thieme
M.Sc. Randolf Scholz

April 8th, 2019
ISMLL Universität Hildesheim

Problem	A	B	C	Sum	
1					/10
2					/10
3					/10
4					/10

Name:		Bonus:	/4
Surname:		Total:	/40
Matrikel:		Grade:	

Note:

- Time: 120 minutes
- No electronic devices besides scientific calculators are allowed!
- Write with a non-erasable pen! Do not use red color!
- Write your name and matrikel at the bottom of each odd-numbered page!
- If you run out of space continue your answer on the corresponding reserve page. Make it clear which problem you are working on!

- If you still need extra space, ask staff for additional sheets.

Reserve Page

1. Regularization

1A. Regularization (3 points)

For a given data-set, three polynomial models of degree 3 were fitted

$$\hat{y}(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

One of them without regularization, one with L1-regularization (LASSO) and one with L2-regularization (RIDGE). In Table 1, the parameters of the fitted models are shown. Explain which model belongs to which regularization scheme.

Hint: Do you notice anything suspicious about the learned parameters?

Figure 1

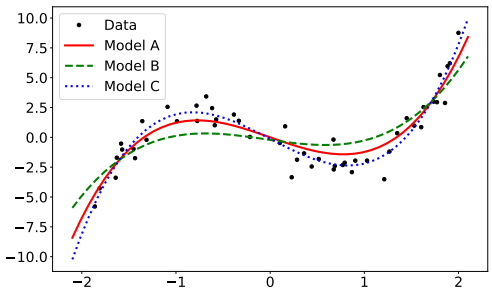


Table 1

Param	Model A	Model B	Model C
β_0	0.0000	−0.2153	−0.1271
β_1	−2.7601	−1.1376	−4.0773
β_2	0.0000	0.1478	−0.0046
β_3	1.5341	0.9434	2.0131

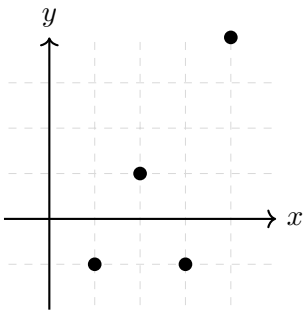
1B. Ridge Regression (3 points)

Given the data from Table 2, train a linear Regression model with L2 regularization. Perform 1 iteration of gradient descent, using the initial parameters $\beta_0 = 0$, $\beta_1 = 0$, learn-rate $\alpha = 0.01$ and regularization strength $\lambda = 1$. Sketch the graph of the learned model into Figure 2.

Table 2

x	y
1	−1
2	+1
3	−1
4	+4

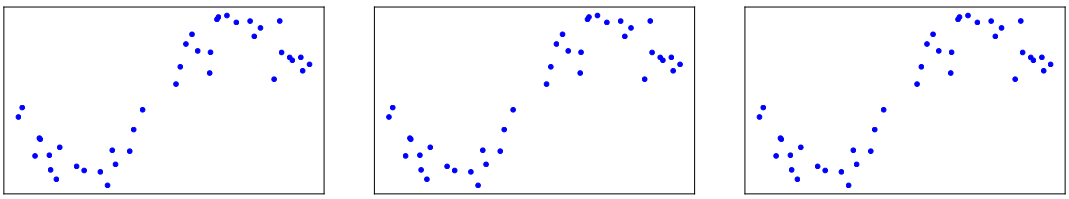
Figure 2



1C. Over- and Underfitting (4 points)

Explain what is meant by the terms over- and underfitting. How can one detect whether a model is over- or underfitting? How can one prevent a model from over- or underfitting? Finally, sketch how a typical over-, under- or well-fitted regression model would look like for the data in Figure 3.

Figure 3



(a) draw under-fit (b) draw well-fit (c) draw over-fit

2. KNN

2A. KNN Basics (2 points)

Explain what is meant by the phrase "KNN is a memory based method".

2B. KNN Regression (4 points)

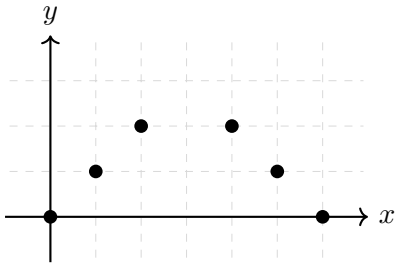
Predict the missing value in Table 3 by applying a KNN regression. Compute the prediction for $K = 2$, $K = 3$ and $K = 4$. Comment on the result.

Secondly, assume that the true value is $y(3) = 3$. Why is it impossible to predict this value with a KNN? Propose a model that is able to do better than the KNN.

Table 3

x	0	1	2	3	4	5	6
y	0	1	2	?	2	1	0

Figure 5



2C. Kernel Regression (3 points)

The prediction of a KNN-regression model is always of the form

$$\hat{y}(x) = \sum_{i=1}^{N} w_i(x) y_i \qquad w_i(x) = \begin{cases} \frac{1}{K} : & x_i \text{ is a KNN of } x \\ 0 : & \text{else} \end{cases} \tag{1}$$

I.e. the prediction is a weighted sum of all training points, where the weight is either $\frac{1}{K}$, if x_i is considered a neighbor of x, or 0 else. An alternative approach, called Kernel Regression, is to choose the weights anti-proportional to the distance, i.e. the closer x_i is to x, the higher the weight $w_i(x)$. For instance a kernel regression is given by

$$\hat{y}(x) = \frac{1}{w(x)} \sum_{i=1}^{N} w_i(x) y_i \qquad w_i(x) = e^{-\gamma \operatorname{dist}(x, x_i)^2} \qquad w(x) = \sum_{i=1}^{N} w_i(x) \tag{2}$$

where $\gamma > 0$ is a hyperparameter controlling the shape of the Gaussian. What could be possible advantages and/or disadvantages of this model over a standard KNN model?

Note: The purpose of dividing by $w(x)$ in (2) is to normalize the weights.

3. Neural Networks

3A. Classification (2 points)

Can the neural network depicted in Figure 6 (using biases and sigmoid activation function) correctly classify the dataset from Figure 7? Give a short explanation for why or why not.

Figure 6: MLP model

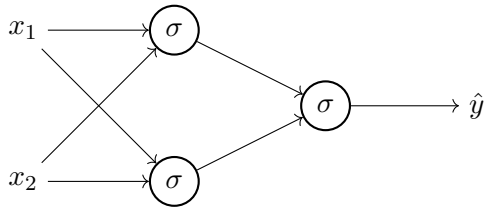
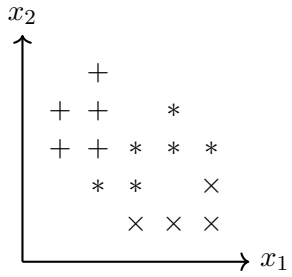


Figure 7

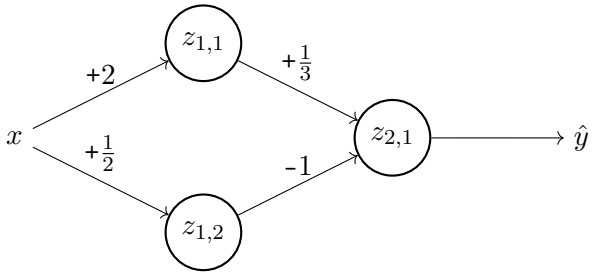


3B. Back-propagation (6 points)

For the scalar input $x = 1$ and target $y = 1$ perform 1 forwards pass and 1 iteration of back-propagation with the neural network depicted in Figure 8. Use the ReLU activation function (3), no bias, set the learning rate to $\eta = \frac{1}{10}$ and use the L2 loss $L = \frac{1}{2}|y - \hat{y}|^2$. After updating the parameters, perform a second forward pass to verify that the loss decreased.

$$\text{ReLU}(x) = \max(x, 0) = \begin{cases} x: & x > 0 \\ 0: & x \leq 0 \end{cases} \qquad \text{ReLU}'(x) = \begin{cases} 1: & x > 0 \\ 0: & x \leq 0 \end{cases} \tag{3}$$

Figure 8: NN model



3C. Neural Networks vs Decision Trees (2 points)

Explain how a Neural Network can simulate a Decision Tree. You can assume that all the data is numerical, and the Decision Tree only uses univariate splits.

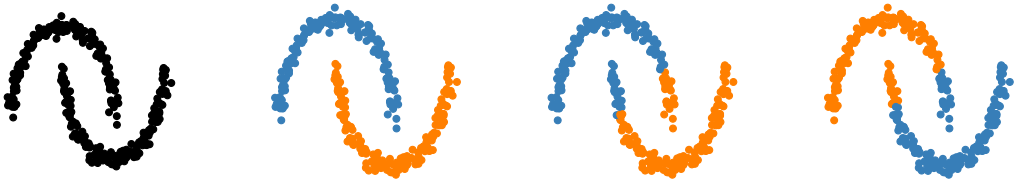
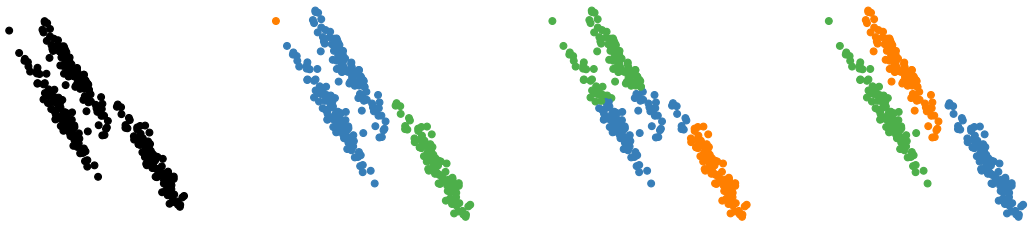
4. Unsupervised Learning

4A. Clustering I (3 points)

Two datasets ("MOONS" and "STRIPES") were each clustered by 3 different methods: K-means clustering, Gaussian-Mixture-Models and Hierarchical Clustering (single link). The results are shown in Table 4. Decide which method corresponds to A, B and C and explain your decision.

Note: K-means and GMM clustering were performed with K=3 and the hierarchical clustering process was stopped once only 3 clusters were left.

Table 4: Different Clustering Methods

Dataset	Method A	Method B	Method C
MOONS			
STRIPES			

4B. Clustering II (5 points)

Apply Hierarchical Clustering to the data from Table 5. Use the Manhattan distance and single (i.e. minimum) linkage as the distortion measure! Use the provided distance matrix (Table 6). Stop once only 2 clusters are left. Draw them into Figure 9. How would the final clusters look like if complete (i.e. maximum) linkage was used instead?

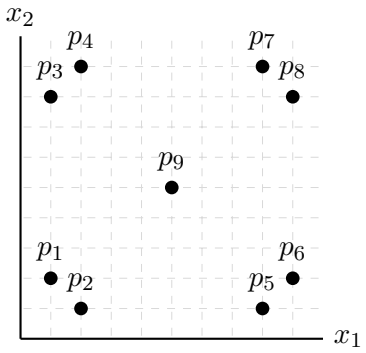
Table 5

id	x_1	x_2
p_1	0	1
p_2	1	0
p_3	0	7
p_4	1	8
p_5	7	0
p_6	8	1
p_7	7	8
p_8	8	7
p_9	4	4

Table 6: manhattan distances

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
p_1	0	2	6	8	8	8	14	14	7
p_2	2	0	8	8	6	8	14	14	7
p_3	6	8	0	2	14	14	8	8	7
p_4	8	8	2	0	14	14	6	8	7
p_5	8	6	14	14	0	2	8	8	7
p_6	8	8	14	14	2	0	8	6	7
p_7	14	14	8	6	8	8	0	2	7
p_8	14	14	8	8	8	6	2	0	7
p_9	7	7	7	7	7	7	7	7	0

Figure 9



4C. Singular Value Decomposition (2 points)

As a dimensionality reduction tool, the SVD can also be used for tasks such as image compression. Consider the gray-scale image of the moon below (Figure 10), which can be encoded as a matrix A with values between 0 (=black) and 1 (=white). In the box on the right (Figure 11) sketch your best guess on what the optimal rank-1 approximation of the image would look like.

Hint: All rank-1 matrices are of the form $vv^\intercal$ for some vector v

Figure 10: Moon

Figure 11: draw here