Exam Machine Learning

Prof. Dr. Ulrike von Luxburg, Dr. Norman Hendrich, July 2013

Name:

Student registration number:

Study program:

I confirm that I do not use any material except the one-page "controlled cheat sheet". I confirm that I do not take any copies of the exam out of the room (neither printed nor hand-written).

Signature:

Achieved number of points:

1	2	3	4	5	6	7	8	9	10	11	12	13	$\sum$

Instructions:

- Overal time is 120 min.
- The exam contains more questions than we expect you to solve. **Each student has to solve no more than 9 questions.**
  - **Computer Science students (9 CP version):** You have to solve 9 questions in total. At least one of them has to be from the set $\{11, 12, 13\}$.
  - **MathMods students (6 CP version):** You have to solve 9 questions out of questions $\{1, ..., 10\}$. (You cannot solve questions 11–13 because they refer to the material of the last part of the lecture.)

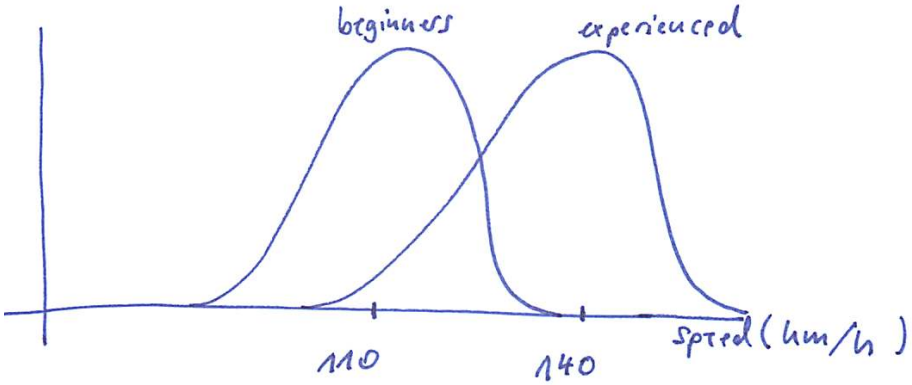
  If you provide answers to more than 9 questions, we are going to include just the best 9 of your answers in the final sum. Example: If a student achieves the following points for the questions, the ones underlined will NOT be counted in the final sum: $\underline{1}$, 3, $\underline{0}$, $\underline{0}$, 2, 2, 4, 2, 3, 2, $\underline{1}$, 3, 2. **So you should concentrate on 9 questions only. Only if you have leftover time you might want to work on the additional questions.**
- Switch off your mobile phone completely and put it in your bag (not trouser pocket).
- Please check whether your exam really contains all 14 pages.
- Please do not use your own paper, you get additional paper by us. Please use a permanent pen (ballpoint pen or so, no pencil!).

Good luck!

1. **Bayesian Decision Theory (4 points)**

   (a) Assume that our training data comes from a joint distribution P on some space $\mathcal{X} \times \{0, 1\}$. Explain the terms "Class conditional distribution", "prior distribution", "posterior distribution" (give their formulas and one sentence about what it means).

   (b) On a motorway, beginner drivers who just got their driving license tend to drive more slowly than experienced drivers. Consider the following plot that shows how the speed values are distributed for each of the classes. We assume that overall there are about 10 % beginners and 90 % experienced drivers on the road.



   Your task is now to classify drivers as "experienced" or "beginner" according to their driving speed. In the figure above, please roughly indicate the position of the decision threshold for the following two rules (you don't need to actually compute the correct threshold, we just want to see in what region you believe it is going to be):

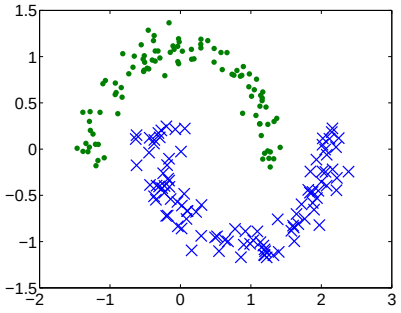
   - According to the maximum likelihood approach.
   - According to the Bayesian criterion.

   Give a short reasoning for your choices.

2. **Overfitting and underfitting (4 points)**

(a) Explain the general concepts of overfitting and underfitting (few sentences).

(b) Assume we train a support vector machine with Gaussian kernel on the following 200 data points in $\mathbb{R}^2$:



The Gaussian kernel has a parameter σ called the kernel width. Please indicate which of the following statements is correct, and give a reason for each of your choices.

If σ is very small, the support vector machine is going to ...

☐ Underfit ☐ Overfit ☐ None of the two

If σ is very large, the support vector machine is going to ...

☐ Underfit ☐ Overfit ☐ None of the two

(c) Now we train a k-nearest neighbor classifier on the same data set. Please indicate which of the following statements is correct, and give a reason for each of your choices.

If k is very small, the kNN classifier is going to ...

☐ Underfit ☐ Overfit ☐ None of the two

If k is very large, the kNN classifier is going to ...

☐ Underfit ☐ Overfit ☐ None of the two

(d) In the data set plotted above, what do you guess would be a good value of σ? Please explain your choice.

☐ 10^{-2} ☐ 10^{-1} ☐ 10^{-0} ☐ 10^{1} ☐ 10^{2}

And what would be a good value for k (roughly)? Explain your choice.

3. **SVM canonical representation (4 points)**

   (a) For an SVM, what is a hyperplane in canonical representation? Why is it necessary to define the canonical representation?

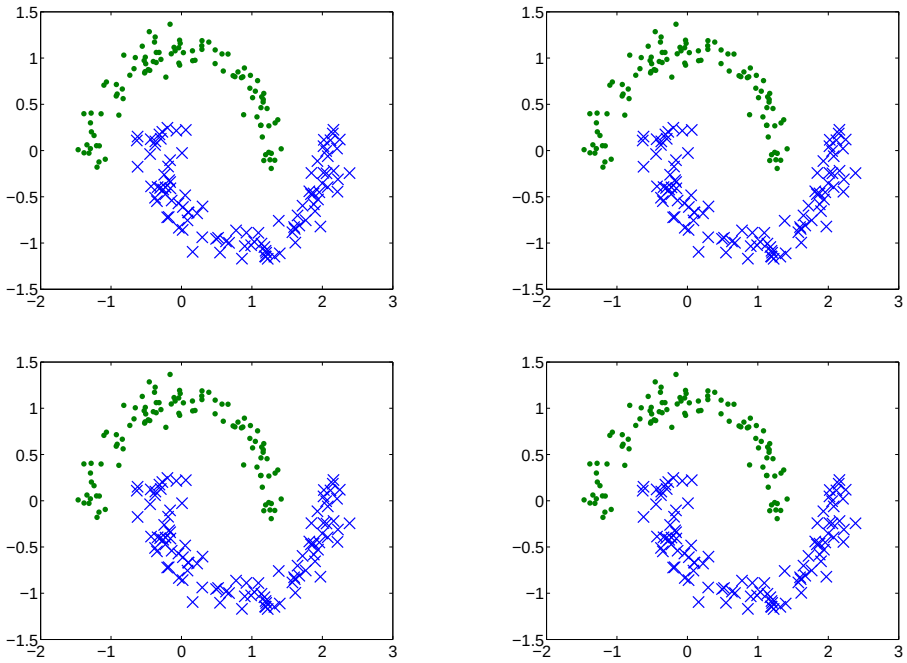
   (b) Why is the outcome of the hard margin SVM optimization problem always a hyperplane in canonical representation? (give a formal argument).

4. **Margin classifiers (4 points)**

   (a) What is the inductive bias of a large margin classifier? Why does it make sense, intuitively?

   (b) How is the margin of a linear classifier defined? Describe in words and provide a sketch. How can the margin be computed if the hyperplane is in canonical representation?

   (c) The following figures all show the same data set. Try to sketch how the separating hyperplane and the margin might look like in this data set, for the following four different cases:



   (a) Linear kernel, hard margin SVM

   (b) Linear kernel, soft margin SVM

   (c) Gaussian kernel with small kernel width σ, hard margin SVM

   (d) Gaussian kernel with large kernel width σ, hard margin SVM

5. **Kernel methods (4 points)**

   (a) What does it mean to kernelize an algorithm?

   (b) What are necessary conditions such that a learning algorithm can be kernelized? How do we proceed when we kernelize an algorithm?

   (c) Why might it be useful to do this?

   (d) Why is the reproducing kernel Hilbert space important in this context?

6. **Evaluating classifiers (4 points)**

   (a) Assume you are given a data set of 1000 pairs $(X_i, Y_i)_{i=1,\ldots,1000}$. Now you want to train and test a classifier that has one parameter λ. How exactly do you proceed to compute the training error, the test error, and the cross validation error for k-fold cross validation?

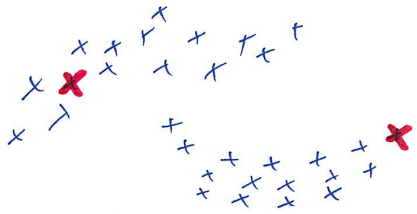
   (b) Assume you applied your procedure to three different classifiers A, B, C, with the results in the table below. Which one do you prefer? Why?

Classifier	A	B	C
Training error	0.05	0.2	0.3
Cross validation error	0.05	0.1	0.2
Test error	0.21	0.25	0.2

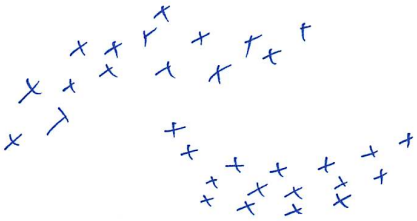
7. **K-means algorithm (4 points)**

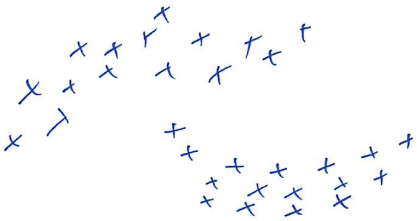
(a) Explain how the k-means algorithm (Lloyds algorithm) works.

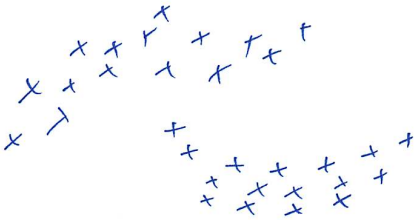
(b) Consider the data set in the first plot. The bold crosses denote the random initialization of the centers for the k-means algorithm.



Simulate the first three iterations of the k-means algorithm. In each of the following figures, indicate the new cluster centers and the point assignment after one run of the while-loop in the algorithm (just roughly).







(c) For each of the following cases, draw a data set that illustrates the following fact, and explain why this is the case in your data set:

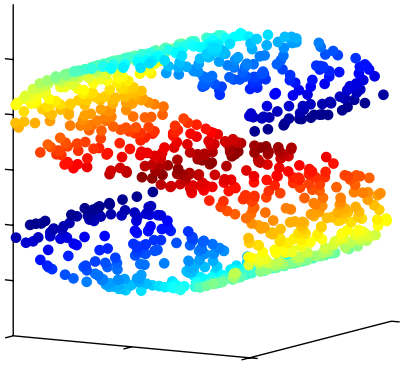
i. There exist initializations such that the outcome of Lloyd's algorithm is obviously bad, while the outcome is good for other initializations.

ii. The outcome is always bad, no matter what the initialization is.

8. **Isomap (4 points)**

(a) What is the goal of the Isomap algorithm? Describe the steps performed by the Isomap algorithm (keywords are enough).

(b) You want to run the isomap algorithm on the following data set. How is the correct output supposed to look like? (Sketch and explanation).



(c) What are the parameters we have to choose for the Isomap algorithm? In the example from the figure, for each of the parameters describe the effect of choosing it high or low.

9. **Multi-dimensional scaling (4 points)**

   Assume we are given a distance matrix D that corresponds to n points in $\mathbb{R}^d$.

   (a) How can we compute the entries of the corresponding Gram matrix (kernel matrix)?

   (b) What can you say about the rank of the matrix? Why?

   (c) Once we have the Gram matrix, how can we construct an embedding of the points in $\mathbb{R}^d$ that preserves all pairwise distances?

   (d) Is such an embedding always possible? Why?

10. **Cross validation (4 points)**

    You are given a regression training set with input space $\mathcal{X} = \mathbb{R}$ and output space $\mathcal{Y} = \mathbb{R}$. It consists of the following three points $\in \mathbb{R} \times \mathbb{R}$: $X_1 = 0, Y_1 = 1$, $X_2 = 1, Y_2 = 0$, $X_3 = 2, Y_3 = 1$.

    Assume you run linear regression and the least squares loss, and you perform 3-fold cross validation (that is, you train on 2 points and validate on the third one). What is the cross validation error in this setting? Give a brief derivation of your result (a sketch might help as well).

11. **Value functions and learning (4 points)**

This exercise can only be solved by those students who attended the last part of the lecture by Norman Hendrich, not by MathMods students.

We assume the standard RL setup with a MDP using states $s \in S$, actions $a \in A$, timestep t, immediate reward r_t, and policies $\pi \in \Pi$. The transition probabilities $P^a_{ss'}$ and reward probabilities $R^a_{ss'}$ may be known or unknown to the learning agent. The goal of the agent is to find a policy π that maximizes its long-term return, $R_T = E\{\sum_{t=0}^{T} \gamma^t r_t\}$ with discount-factor $0 \leq \gamma < 1$. T may be finite for episodic tasks or $T = \infty$ for non-episodic discounted tasks.

(a) Give your interpretation of the discount factor γ. What is the effective time-horizon considered by the agent when γ is either very small or near to one?

(b) What is the meaning of the state-value function, $V^\pi(s)$?

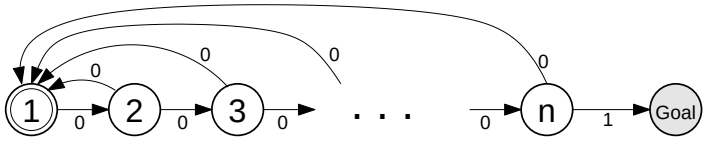
(c) Many RL algorithms are based on the Bellman-Equations. Explain the Bellman Equation for the optimal value-function, $V^*(s)$:

(d) Assuming an estimate of the action-value function $Q^\pi(s, a)$ is available: how can the agent improve its current policy?

12. **Linear world, exploration (4 points)**

This exercise can only be solved by those students who attended the last part of the lecture by Norman Hendrich, not by MathMods students.

The RL environment in the following figure consists of n states. For each episode, the agent starts in the start-state 1 on the left and should learn to reach the goal state on the right. On any timestep, the agent can take two actions: moving up (which takes it back to the starting state) or moving right. There is no reward for any move, except for entering the goal state, where a reward of $r = 1$ is given. The agent uses a greedy policy, but will take a random move with $p_{\mathsf{up}} = p_{\mathsf{right}} = 0.5$ when $Q(s, \mathsf{up}) = Q(s, \mathsf{right})$.



(a) We initialize the action-value function to $Q(s, a) = 0 \; \forall s, a$. What is the time-complexity for learning the task as a function of the number of states n? Explain your result.

(b) What happens when using optimistic initialization, e.g. $Q(s, a) = 2 \; \forall s, a$? What is the time-complexity for learning the task as a function of the number of states now? Why?

13. **Linear world, continued (4 points)**

This exercise can only be solved by those students who attended the last part of the lecture by Norman Hendrich, not by MathMods students.

Consider the scenario of the last question. Once the task has been learned, $Q^*(s, a)$ is known, and the greedy policy would reach the goal state in n steps. What is the probability of reaching the goal state in n steps for an ε-greedy policy, as a function of ε?