Written examination paper for PGR210 – Machine Learning and Natural Language Processing
Department of Technology, Kristiania University College
Autumn 2021

Examination paper released: 19.11.2021

Examination deadline: 17.12.2021

Academic contact during examination: Andrii Shalaginov, andrii.shalaginov@kristiania.no, +47 46 572 592 and Huamin Ren, huamin.ren@kristiania.no

Technical contact during examination: eksamen@kristiania.no

Exam type: Written home examination in groups (1-5 students)

Support materials: All support materials are allowed

Final report format: use LaTeX or Word and font 12 with 1.5 spacing. The limit is 20 pages max that includes abstract, report, list of bibliography in the end, figures and tables. Both parts (ML and NLP should be in the same PDF report)

An example of possible report structure. Abstract, Introduction, State of the Art literature, Your used approach/Methodology, Analysis of results, Conclusions

Grading scale: Norwegian grading system using the graded scale A - F where A is the best grade, E is the lowest pass grade and F is fail

Weighting: 100% or overall grade

Plagiarism control: We expect your own independent work. Please, use citations and quotations in case if there is a material you want to include in the report.

Learning outcomes as per course description:
Knowledge. The student

- can understand the basic data structures and algorithms for machine learning.
- knows the mathematical concepts underlying the design and analysis of machine learning techniques appropriate for a given data science problem
- has insights into the strengths and weaknesses of Dimensionality Reduction Algorithms: variance thresholds, correlation thresholds, principal component analysis (PCA), linear discriminant analysis (LDA)
- can explain the basic natural language processing concepts and techniques for text analytics
- understands the basic pipeline for natural language processing; for a simple topic modelling task, is able to carry out step-by-step processing, representation and come out with a solution
- knows text processing and analytics methods (such as tokenization, word representation, topic modelling and clustering) for various data science domains

Skills. The student

- can select appropriate machine learning methods (such as linear models, classification models, text classification, semantic textual similarity, word sense disambiguation and neural language models) and tools for a given data science problem
- can analyze mathematically the performance of machine learning methods and techniques
- can apply techniques from the course to new data science problems in terms of selection of appropriate machine learning methods, techniques and tools
- can use python or similar to implement machine learning methods and techniques
- can discuss concepts and applications of machine learning (including text)

Competence. The student

- can differentiate the suitability and efficiency of programs in terms of the machine learning methods and techniques (incl. text analytics) employed
- can apply the knowledge of and skills in machine learning in various data science domains
- can critically reflect on the tradeoffs in the design and implementation of machine learning methods and techniques.

Exam Task

1. Machine Learning

1.1. Theoretical Task

1. Present your understanding of Artificial Neural Network (ANN). Draw basic structure and highlight the main components.
2. What are the general types of Neural Network (according to topology and application domain)?
3. What is an activation function? Present some of the popular functions that you know.
4. What is a regression tree? Please explain how you would proceed to build such a tree in general. Then give an example of a practical problem in which this learning methodology would be useful. Please sketch it and denote its nodes and vertices. What kinds of models can be used in the leaves of such trees? Explain why it is frequently necessary to prune such trees.

1.2. Practical Task: Learning as a Search

There exists an optimization problem, which is described by the fitness function f(x; y) = 25 * (y + x^2)^2 + (1 + x)^2. In order to find an optimal solution you need to find the global minima of this function with respect to the two variables 'x' and 'y'. They are real-number variables. Perform the following actions:

1.1. Build a plot of the function (Octave, Matlab, Python are good choices) and tentatively define the possible area of global minima of the function. Make a hypothesis about the existing of local and global optimal solutions.

1.2. Implement a simple Genetic Algorithm (GA), which will minimize the fitness function and can deal with the real-values features. Crossover/Mutation operations and initial seed are up to you. Try to use GA with the following parameters:

- size of population in each generation: 4, 10
- crossover operations per generation: 25%, 50% , 75%
- mutations per generation: 1%, 5% and 50% (or at least 1 element in the generation)
- total number of generations: 10, 100 or 1000
- selection of candidates with the best values of fitness function is performed after crossover/mutation

1.3. Implement a simple Gradient Descent method. If you like, you may use an online tool or library that calculates the partial derivatives for you. Try using the following parameters:

- Employ several stop conditions: absolute differences between the function values, absolute differences between the arguments values and absolute differences between the first order derivative values on adjacent steps.
- learning steps: 1e-6, 0.1, 1.
- maximal number of iterations: 100, 1000

1.4. Compare the obtained results for finding global or local minima. From the program output, estimate which set of Genetic Algorithm parameters gives the fastest convergence to fitness function minimum and why? How many generations did it require? For gradient descent, which stop conditions, learning steps, and maximum number of iterations produced the fastest or most accurate results? Why do you think these parameters produced the best results, and were there any settings which produced incorrect results (and why did the produce incorrect results)? Present the output of the program in report and describe the obtained results.

1.3. Practical Task: Support Vector Machine

1. Find an implementation of the SVM model that offers a capability to adjust key parameters of the model. Find a classification dataset on Kaggle or UCI.

2. Train the model using your dataset, and try several values of the penalty parameter C (0.1, 1, 100).

3. Analyze the accuracy and robustness of the algorithm and and propose methods to improve them.

4. What kind of data will cause the general (standard) SVM classifier to fail, and how can this problem be mitigated? Does it work well for your dataset?

5. Try to use different kernel functions and see how this influences the accuracy on cross-validation. Explain why.

2. Natural Language Processing

2.1. Practical Task: Text processing, feature extraction and representation by using both TF and TF-IDF schemes

Given a data file (Exam_MB210_NLP.csv), where reviews on movies are provided.

1. Data preparation: load the file, access the columns, then through printing and visualization, understand the meaning in each column. Then create a new column, name it as 'description' by concatenating the strings from two columns: tagline and overview.

2. Text processing: convert words in 'description' to lower case, remove white space, remove words from stop_words (from nltk package), remove special characters (such as '/n') and add other necessary processings.

3. TF and TF-IDF representation on 'description': for each sample in the dataset, generate TF and TF-IDF representation for each sample based on the column of 'description'.

2.2. Practical Task: Topic modelling

Use TF and TF-IDF representation generated in task 2.1 to perform topic modelling. Select and compare two topic modelling algorithms from LDiA, Truncated SVD, Word2Vec or any other topic modelling algorithms, and then analyze the results.

2.3. Analysis Task: Searching for similar movies

Assume you would like to find similar movies as 'Spider-Man' based on the given dataset, what would you do? Please introduce your solution step-by-step, where such information should be provided:

1. Details on each step and espected inputs/outputs of each step

2. Major algorithm to be used to solve this problem

3. The results
4. Analysis on the results

Be noted visualization should be used when exploring the data or illustrating the results.