Machine Learning 2018-2019

Final Exam

Yevgeny Seldin **Christian Igel**
Department of Computer Science
University of Copenhagen

You must submit your individual solution of the exam electronically via the **Digital Exam / Digital Eksamen** system. The deadline for submitting the exam is **16:00, Friday, 25 January 2019**. The exam must be solved **individually**. You are **not allowed** to work in groups or discuss the exam questions with other students. For fairness reasons any questions about the exam must be posted on the Absalon forum.

WARNING: The goal of the exam is to evaluate your personal achievements in the course. We believe that take-home exams are most suitable for this evaluation, because they allow to test both the theoretical and practical skills. However, our ability to give take-home exams strongly depends on your honesty. Therefore, any suspicion of cheating, in particular collaboration with other students, will be directly reported to the head of studies and prosecuted in the strictest possible way. It is also strictly prohibited to post the exam questions or parts thereof on the Internet or on discussion forums and to seek help on discussion forums. And you are not allowed to store your solutions in open access version control repositories, or to post them on the Internet or on discussion forums. Be aware that if proven guilty you may be expelled from the university. Do not put yourself and your fellow students at risk.

A solution consists of:

- A PDF file with detailed answers to the questions, which may include graphs and tables if needed. Do *not* include your source code in this PDF file. Do *not* include the task description or parts thereof in your report.
- Your solution source code (Matlab / R / Python scripts or C / C++ / Java code) with comments about the major steps involved in each question (see below). Source code must be submitted in the original file format, not as PDF.

- Your code should be structured such that there is one main file that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions, classes, etc.
- Your code should also include a README text file describing how to compile and run your program, as well as list of all relevant libraries needed for compiling or using your code.

1 Experiment Design

1. You are working at a hospital and you have collected an i.i.d. sample of 2000 patients and annotated it for presence or absence of some disease (binary annotation). You want to organize a competition to find a classifier for the disease. You know that you should release some of the data for the competition and keep some data aside in order to evaluate the outcomes of the competing teams. But you have to decide how much of the data you release and how much you keep for evaluation. Briefly explain the trade-off that you are facing: what are the advantages and disadvantages in keeping more or less data.

2. You have 20 teams that have signed up for the competition and your boss requires you to provide a confidence interval of 0.05 on the prediction accuracy of the best classifier that will hold with probability at least 95%. In other words, with probability at least 95% the expected zero-one error of the classifier you have selected should not exceed your estimate by more than 0.05 (one-sided error). [Side remark: the confidence interval of 0.05 means that the error will not increase by more than 5% of the total number of predictions.] How many samples do you have to keep aside in order to satisfy this requirement, assuming that you accept 1 solution from each team? Provide a complete calculation, numerical answers without any derivations or explanations will not be accepted.

3. You have conducted the competition above, but were not satisfied with the prediction accuracy of the winner. You decided to make another competition and were very lucky to convince your boss to support annotation of another 1000 patients. You decided to release the old 2000 patients data for training and keep the new 1000 samples for evaluating the outcome of the new competition. Your boss requires from you the same confidence interval of 0.05 with probability at least 95%. How many teams can you accept to take part in the competition assuming that you accept only 1 solution from each team? Provide a complete calculation, numerical answers without any derivations or explanations will not be accepted.

2 Smart Fusion

You are a team leader and you have 3 excellent data scientists in your team. You have received data for binary classification from a new client. The data consists of 2000 annotated i.i.d. samples. You gave the data to your team members and asked them to provide you a classifier. The team members are all proud individualists and they decided that each of them will work on the data independently of the rest. And they did the following:

1. The first scientist has split the data into 1000 training and 1000 test points, did all the training on the training set and then tested the final output on the test set and got a test error $\hat{L}(\hat{h}_1^*, S_1^{\texttt{test}}) = 0.1$.

2. The second scientist has split the data into 1500 training and 500 test points, did all the training on the training set and then tested the final output on the test set and got a test error $\hat{L}(\hat{h}_2^*, S_2^{\texttt{test}}) = 0.08$.

3. The second scientist has split the data into 1750 training and 250 test points, did all the training on the training set and then tested the final output on the test set and got a test error $\hat{L}(\hat{h}_3^*, S_3^{\texttt{test}}) = 0.075$.

All three have reported their classifiers and test errors to you. Now you have to decide which of the three classifiers you will provide to your client. Explain how you will pick the classifier and provide a generalization bound for the classifier that you will return to the client. The generalization bound should hold with probability at least 95%.

3 VC Dimension

Let $d_{\text{VC}}(\mathcal{H})$ denote the VC-dimension of a hypothesis class $\mathcal{H}$. For a finite hypothesis class $\mathcal{H}$ let $|\mathcal{H}|$ denote the size of $\mathcal{H}$ (the number of hypotheses in $\mathcal{H}$). All the questions below consider binary classification.

To avoid any confusion: a hypothesis h is a function from $\mathcal{X}$ to $\{\pm 1\}$ and if two hypotheses h_1 and h_2 label $\mathcal{X}$ identically then they are identical and count as a single hypothesis in $\mathcal{H}$.

1. Prove that if $|\mathcal{H}| = 2$ then $d_{\text{VC}}(\mathcal{H}) = 1$.

2. Let $\mathcal{H}_4$ be a hypothesis class with 4 hypotheses, $\mathcal{H}_8$ be a hypothesis class with 8 hypotheses and $\mathcal{X}$ be a finite set of size 8. (I.e., there are just 8 possible values that the input parameter x can take.) We emphasize that $\mathcal{H}_4$ is *not* necessarily a subset of $\mathcal{H}_8$. What is the relation between $d_{\text{VC}}(\mathcal{H}_4)$ and $d_{\text{VC}}(\mathcal{H}_8)$? Your answer should be one of the following statements:

(a) We always have $d_{\text{VC}}(\mathcal{H}_4) < d_{\text{VC}}(\mathcal{H}_8)$.

(b) We always have $d_{\text{VC}}(\mathcal{H}_4) \leq d_{\text{VC}}(\mathcal{H}_8)$.

(c) It is possible to construct an example of $\mathcal{H}_4$ and $\mathcal{H}_8$, such that $d_{\text{VC}}(\mathcal{H}_4) > d_{\text{VC}}(\mathcal{H}_8)$.

Pick the right statement and prove it. Answers without a proof will not be accepted.

3. Answer the same question when the size of $\mathcal{X}$ is 3 points.

4 Cauchy-Schwarz

This exam question corresponds to exercise (4.3) in the textbook by Steinwart and Christmann (2008). Given a vector space $\mathcal{F}$ and a positive semi-definite symmetric bilinear form $\langle ., . \rangle : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, that is,

1. $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$
2. $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$
3. $\langle \alpha \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \alpha \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$

for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{F}$. Show the Cauchy-Schwarz inequality

$$| \langle \mathbf{x}, \mathbf{y} \rangle |^2 \leq \langle \mathbf{x}, \mathbf{x} \rangle \cdot \langle \mathbf{y}, \mathbf{y} \rangle$$

holds for all $\mathbf{x}, \mathbf{y} \in \mathcal{F}$.

Hint: Start with $0 \leq \langle \mathbf{x} + \alpha \mathbf{y}, \mathbf{x} + \alpha \mathbf{y} \rangle$ and consider the case $\alpha = 1$ and $\alpha = -1$ if $\langle \mathbf{x}, \mathbf{x} \rangle = \langle \mathbf{y}, \mathbf{y} \rangle = 0$. Otherwise, if, e.g., $\langle \mathbf{y}, \mathbf{y} \rangle \neq 0$, use $\alpha = -\frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{y}, \mathbf{y} \rangle}$.

Why is this relevant? The Cauchy-Schwarz inequality is of general importance. For example, it is needed for proving the following property completing the line of arguments linking kernels to scalar products in the lecture.

Recall that a *dot product* on a vector space $\mathcal{F}$ is a symmetric bilinear form $\langle ., . \rangle : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ that is *strictly* positive definite, i.e., $\forall \mathbf{x} \in \mathcal{F} : \langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ with equality only for $\mathbf{x} = \mathbf{0}$.

Given a kernel k and

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \qquad\qquad g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j)$$

(see lecture slides for details and context) we defined the scalar product

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)) \ .$$

In the lecture, we stated

$$\langle f, f \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \ ,$$

which follows from the kernel being positive semi-definite. However, we did not prove that $\langle f, f \rangle = 0$ implies $f = \mathbf{0}$, but this is necessary for having a scalar product. A proof can be found in the textbook by Schölkopf and Smola (2002), which is, however, not fully correct. It argues with a Cauchy Schwarz inequality for kernels, but a Cauchy Schwarz inequality for a positive symmetric bilinear form would be needed – as derived in this assignment.

5 Rotation of the inputs

If we use PCA for preprocessing, we rotate the input to our machine learning algorithm. How does rotation affect different methods? Consider supervised learning for classification. Let the input space be $\mathbb{R}^d$. Now consider the transformation of the input data by a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{d \times d}$, which only rotates the data. That is, instead of $\boldsymbol{x}$ the algorithm is provided $\boldsymbol{R}\boldsymbol{x}$. The transformation is applied to train and test data.

In mathematical terms, we say that $\boldsymbol{R}$ is a member of the d-dimensional *special orthogonal group* and write $\boldsymbol{R} \in \mathrm{SO}(d)$. The formal definition of a rotation matrix is that $\boldsymbol{R}$ is a rotation matrix if and only if $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$ and $\det \boldsymbol{R} = 1$.

Which classification methods are affected by the transformation in the sense that their classification performance may change if trained and tested on the transformed data compared to the original data? Consider three classifiers:

1. k-nearest neighbor algorithm with L_1-distance (Manhattan distance), that is, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$d_1(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_1 = \sum_{i=1}^{n} |x_i - y_i|$$

2. Support vector machine with homogeneous polynomial kernel, that is, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and positive integer d

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle)^d$$

3. Random forest

For each of them, provide a proof either showing that the classification performance will not change or may change. For the former, you may ignore numerical issues. For the latter, note that it is sufficient to give a single toy example where the classification changes. The toy examples should be small, something like 3 points in 2 dimensions. The most important part of the toy examples is to explain why rotation may influence classification outcomes. We do not expect you to start sampling hundreds of points to illustrate the effect.

6 Alzheimer's disease Diagnosis

Application Background

This section of the exam considers the task of automatic diagnosis of Alzheimer's disease. The data are a subset of the data analyzed by Sørensen et al. (2016, 2017), and we refer to these two studies for further background information.

Alzheimer's disease (AD) is a severe neurodegenerative disease. The vast majority of AD diagnoses are uncertain. Often, a magnetic resonance imaging (MRI) scan is part of the diagnosis, but only to rule out other causes of the symptoms. However, our recent work suggests that MRI scans carry much more quantitative and diagnostic information. A system developed by DIKU and the DIKU spin-off company Biomediq can separate cognitively normal, dementia, and pre-dementia patients with high accuracy when incorporating MRI data, as demonstrated by winning the MICCAI CADDementia Grand Challenge in 2014 (Sørensen et al., 2016, 2017). Here we consider data preprocessed in a similar way as in that system. However, we consider features extracted from MRI only and restrict the analysis to two classes, healthy and AD.

The Exercises

Question 1 (data understanding and preprocessing)**.** Download and extract the data.

Consider the training data `trainInput.csv` and the corresponding labels `trainTarget.csv`. Report the class frequencies, that is, for each of the 2 classes report the number of data points divided by the total number of data points) in the training data.

The ith row in `trainInput.csv` are the features of the ith training pattern. The class label of the ith pattern is given in the ith row of `trainTarget.csv`.

Deliverables: description of software used; frequency of classes

Question 2 (principal component analysis)**.** Perform a principal component analysis of the training data `trainInput.csv`. Plot the eigenspectrum (see the plot on slide 28 of the *PCA* slides for an example). How many components are necessary to "explain 90 % of the variance"? Visualize the data by a scatter plot of the data projected on the first two principal components. Use different colors for the different classes in the plot.

Deliverables: description of software used; plot of the eigenspectrum; number of components necessary to explain 90 % of variance; scatter plot of the data projected on the first two principal components with different colors indicating the 7 different classes

Question 3 (clustering)**.** Perform 2-means clustering of `trainInput.csv`. After that, project the cluster centers to the first two principal components of the training data. Then visualize the clusters by adding the cluster centers to the plot from the previous exercise. Briefly discuss the results: Did you get meaningful clusters? Initialize the cluster centers with training data points from different classes. Take the *first data point* from each class you can find in `trainInput.csv` (i.e., the first class center is the data point in the very first line in the file).[1]

Deliverables: description of software used; one plot with cluster centers and data points; short discussion of results

Question 4 (binary classification using support vector machines)**.** Now we consider binary classification using support vector machines (SVMs). Please use the data files `trainSubsetInput.csv`, `trainSubsetTarget.csv`, `testInput.csv`, and `testTarget.csv`. These are real-world data. To speed up the model selection process, only a subset of the dataset analysed in the previous subtasks is used. The splitting into training and testing data has been done by the people providing the data. Please do *not* use asymmetric loss functions, class-dependent regularization parameters (i.e., different "C-values" depending on the class), etc.[2]

For this exercise, use standard C-SVMs as introduced in the lecture. Employ radial Gaussian kernels of the form

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|^2) \ . \tag{1}$$

Here $\gamma > 0$ is a bandwidth parameter that has to be chosen in the model selection process. Note that instead of γ often the parameter $\sigma = \sqrt{1/(2\gamma)}$ is considered.

Jaakkola's heuristic provides a reasonable initial guess for the bandwidth parameter σ or γ of a Gaussian kernel. To use Jaakkola's heuristic to estimate a good

[1] This is done in order to simplify the grading. It is in general not necessarily a good idea to take the first occurence.

[2] Be careful, because, for instance, the SVM from the Matlab Bioinformatics Toolbox may by default use different regularization parameters depending on the class and the class frequency – at least that was the case in the past.

value for σ, consider for every training example $\boldsymbol{x}_i$ the distance to the closest training example $\boldsymbol{x}_j$ having a different label (i.e., $y_i \neq y_j$). The median of these distances can be used as a measure of scale and therefore as a guess for σ. More formally, compute

$$G = \left\{ \min_{(\boldsymbol{x}_j, y_j) \in S \wedge y_i \neq y_j} \{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\} \mid (\boldsymbol{x}_i, y_i) \in S \right\}$$

based on your training data S. Then set σ_{Jaakkola} equal to the median of the values in G:

$$\sigma_{\text{Jaakkola}} = \text{median}(G)$$

Compute the bandwidth parameter γ_{Jaakkola} from σ_{Jaakkola} using the identity given above.

Use grid-search to determine appropriate SVM hyperparameters γ and C. Look at all combinations of

$$C \in \{b^{-1}, 1, b, b^2, b^3\}$$

and

$$\gamma \in \left\{ \gamma_{\text{Jaakkola}} \cdot b^i \mid i \in \{-3, -2, -1, 0, 1, 2, 3\} \right\} \ ,$$

where the base b can be chosen to be either 2, the base e of the natural logarithm (Euler's number), or 10. Feel free to vary this grid. For each pair, estimate the performance of the SVM using 5-fold cross validation. Pick the hyperparameter pair with the lowest average 0-1 loss (classification error) and use it for training an SVM with the complete training dataset. Only use `trainSubsetInput.csv` and `trainSubsetTarget.csv` in the model selection and training process.

Report the values for C and γ you found in the models selection process. Compute the classification accuracy based on the 0-1 loss on the training data as well as on the test data `testInput.csv` and `testTarget.csv`. An accuracy on the test set significantly larger than 75 % can be expected.

Deliverables: description of software used; a short description of how you proceeded; initial γ or σ value suggested by Jaakkola's heuristic; optimal C and γ found by grid search; classification accuracy on training and test data

References

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

L. Sørensen, C. Igel, N. Liv Hansen, M. Osler, M. Lauritzen, E. Rostrup, and M. Nielsen. Early detection of Alzheimer's disease using MRI hippocampal texture. *Human Brain Mapping*, 37(3):1148–1161, 2016.

L. Sørensen, C. Igel, A. Pai, I. Balas, C. Anker, M. Lillholm, and M. Nielsen. Differential diagnosis of mild cognitive impairment and Alzheimer's disease using structural MRI cortical thickness, hippocampal shape, hippocampal texture, and volumetry. *NeuroImage: Clinical*, 13:470–482, 2017.

I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer-Verlag, 2008.