Machine Learning 2016/2017

Final Graded 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, 27 January 2017**. 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. 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.
- 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 In a galaxy far, far away

In astronomy, estimating distances to galaxies is of great significance. It allows us to get a 3D view of our Universe, which we can use to understand the distribution of matter. Also, since light travels at finite speed, a larger distance means that we are looking further back in time, thus seeing how the Universe looked like billions of years ago. An accurate distance, therefore, translates to knowing the age of the Universe at the time the light was emitted.

Unfortunately, estimating distances is nontrivial. Distances are so great that light has been travelling millions or even billions of years before it hits our detectors. We cannot interact with the galaxies in any way, so we are left with no option but to infer what we can from the light. The distance of a galaxy can be estimated based on its *redshift*. Redshift is caused by the Doppler effect, which shifts the spectrum of an object towards longer wavelengths when it moves away from the observer. As the universe is expanding uniformly, we can infer the velocity of a galaxy by its redshift and, thus, its distance to Earth.

Inferring the redshift of a galaxy is relatively easy if we have a highly detailed spectrum of the light. However, if only images of the galaxy are available the task is much more challenging. By taking images of the galaxy in various band-pass filters and summing up all the light we receive in each filter, we can approximate a very coarse spectrum. Using this, we can estimate a “photometric redshift”, that is, a redshift estimated from images.

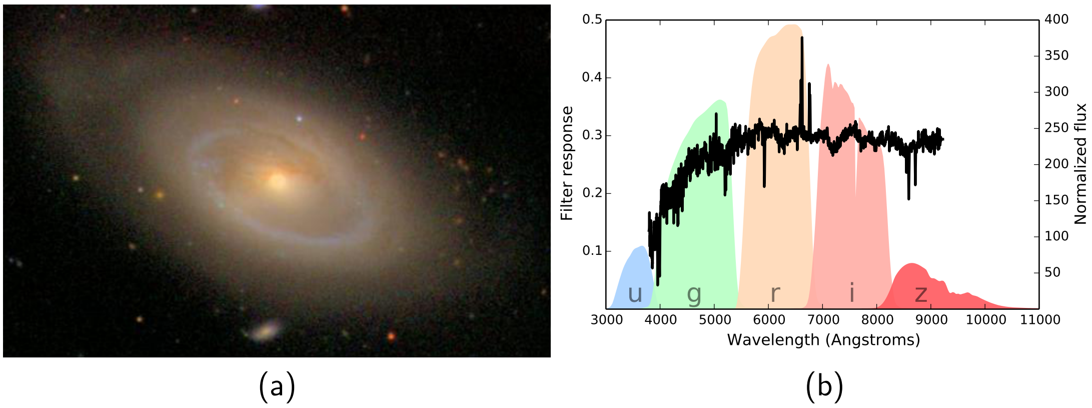
In the data files you will find the integrated light of 10,000 galaxies in five different band-pass filters (termed u, g, r, i, and z, see Figure 1), measured in two different ways (the so-called model magnitudes and Petrosian magnitudes). You also get the derived colours, which are simply the subtraction of one magnitude from another. These can sometimes be informative.

You are provided the following data sets:

`ML2016GalaxiesTrain.dt`	input features describing 5000 galaxies for training
`ML2016SpectroscopicRedshiftsTrain.dt`	labels/redshifts of the training examples
`ML2016GalaxiesTest.dt`	input features describing 5000 galaxies for testing
`ML2016SpectroscopicRedshiftsTest.dt`	labels/redshifts of the testing examples
`ML2016EstimatedRedshiftsTest.dt`	SDSS photometric predictions of the labels/redshifts in the test data

The files `ML2016SpectroscopicRedshiftsTrain.dt` and `ML2016SpectroscopicRedshiftsTest.dt` contain spectroscopically derived redshifts, which we regard as the “ground truth”.

The file `ML2016EstimatedRedshiftsTest.dt` contains photometric redshifts for



(a) (b)

Figure 1: An example from the *Sloan Digital Sky Survey* (SDSS) (Aihara et al., 2011). (a) An image of the spiral galaxy NGC 5750. (b) Its associated spectrum overlapping the five photometric intensity band filters u,g,r,i,z.

the test data estimated by the researchers behind the Sloan Digital Sky Survey (SDSS, `http://www.sdss3.org/dr10/`), which has provided all the data. See if you can beat the researchers' estimates. Researchers at DIKU have also worked on these data (e.g., Stensbo-Smidt et al. 2017).

Question 1.1 (Data preparation)**.** Compute the variance σ^2_{red} of the redshifts in the training data `ML2016SpectroscopicRedshiftsTrain.dt`. Compute the error of the SDSS predictions on the test data (i.e., how good the predictions in `ML2016EstimatedRedshiftsTest.dt` match the data in `ML2016SpectroscopicRedshiftsTest.dt`) using the mean squared error.

Deliverables: Report variance of redshifts in the training data, report error of SDSS predictions on test data

Question 1.2 (Linear regression)**.** Apply linear regression to the data. Train on the training data (`ML2016GalaxiesTrain.dt` and `ML2016SpectroscopicRedshiftsTrain.dt`) and evaluate the model on the test data (`ML2016GalaxiesTest.dt` and `ML2016SpectroscopicRedshiftsTest.dt`). Report the model and its quality measured by the mean squared error on training and test set.

Divide the obtained mean squared error by the variance σ^2_{red} computed in question 1.1. What does the result tell you? In general, what would a result smaller or larger than one tell you?

Deliverables: Mean squared error on training and test set; parameters of the model; errors normalized by the variance; discussion of the meaning of normalized errors larger and smaller than 1

Question 1.3 (Non-linear regression)**.** Now apply a non-linear regression method to the data. Note that you may only use the training data for model selection and hyperparameter tuning. Please describe your approach and your results in detail.

Train on the training data set and evaluate on the test data set. Discuss the results in comparison to the linear regression model.

You are free to apply any non-linear method you like. You have to briefly argue why you selected a particular method. If you choose a method that was not introduced in the course, you are supposed to describe the algorithm in full detail.

Deliverables: Detailed description of your approach; mean squared error on training and test set; discussion of results

2 Weed

The data for the following tasks are taken from a research project financed by Miljøstyrelsen and involving researchers from DIKU and PLEN/KU. Selected results from the project are described by Rasmussen et al. (2016) and Olsen et al. (2017). While the problem setting is inspired by Olsen et al. (2017), the data were generated differently, in particular without preprocessing to compensate for color-balancing and a number of illumination effects.

Introduction to the problem. Pesticide regulations and a relatively new EU directive on integrated pest management create strong incentives to limit herbicide applications. In Denmark, several pesticide action plans have been launched since the late 1980s with the aim to reduce herbicide use. One way to reduce the herbicide use is to apply site-specific weed management, which is an option when weeds are located in patches, rather than spread uniformly over the field. Site-specific weed management can effectively reduce herbicide use, since herbicides are only applied to parts of the field. This requires reliable remote sensing and sprayers with individually controllable boom sections or a series of controllable nozzles that enable spatially variable applications of herbicides. Preliminary analysis (Rasmussen et al., 2016) indicates that the amount of herbicide use for pre-harvest thistle (Cirsium arvense) control with glyphosate can be reduced by at least 60 % and that a reduction of 80 % is within reach. See Figure 2 for an example classification. The problem is to generate reliable and cost-effective maps of the weed patches. One approach is to use user-friendly drones equipped with RGB cameras as the basis for image analysis and mapping.

The use of drones as acquisition platform has the advantage of being cheap, hence allowing the farmers to invest in the technology. Also, images of sufficiently high resolution may be obtained from an altitude allowing a complete coverage of a normal sized Danish field in one flight.

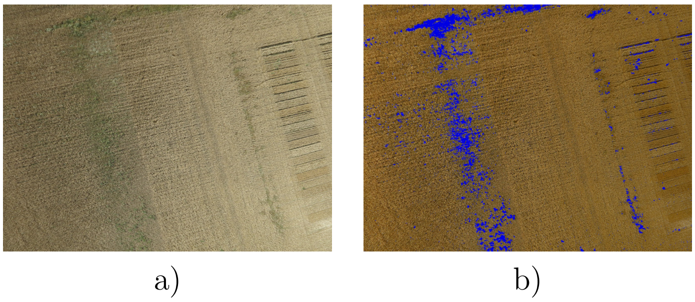

a) b)

Figure 2: Example from another approach. a) An original image. b) Initial pixel based detection.

Data. Your data is taken from a number of images of wheat fields taken by a drone carrying a 3K by 4K Canon Powershot camera. The flying height was 30 meters. A number of image patches, all showing a field area of 3×3 meters were extracted. Approximately half of the patches showed crop, the remaining thistles. For each patch only the central 1×1 meter sub-patch is used for performance measurement. The full patch was presented to an expert from agriculture and classified as showing either weed (class 0) or only crop (class 1).

In Figure 3 two patches classified as crop and two patches classified as weed are shown. Two of the patches are easy to classify (expert or not), while the remaining two less clearly belong to either of the classes.

For each of the cental sub-patches (here of size 100×100 pixels), 13 rotation and translation invariant features were extracted. In more detail, the RGB-values

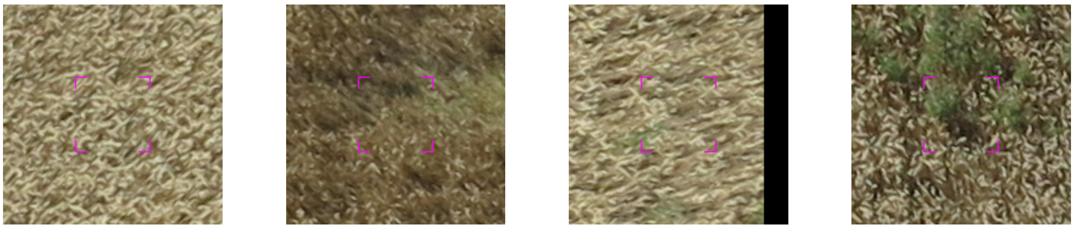


Figure 3: The two images on the left are classified as crop. The two images on the right are classified as weed. The classification of the middle two patches is debatable. The central area used for performance evaluation is indicated by the small magenta markers.

were transformed to HSV and the hue values were extracted. The 13 features were obtained by taking a 13-bin histogram of the relevant color interval.

The training and test data are the files `ML2016WeedCropTrain.csv` and `ML2016WeedCropTest.csv`, respectively. The last column corresponds to the class label. You can get pretty accurate classification results on these data. Note that for other crop types and other flying heights the classification may be more challenging.

Do not use normalization as preprocessing in questions 2.1 and 2.2. Normalization will be discussed in question 2.3.

Question 2.1 (Logistic Regression)**.** Train a logistic regression model on the training data set and evaluate it on the test set. Report the model parameters. Measure the classification errors using the 0-1 loss.

Deliverables: Description of software used; parameters of the logistic regression model; 0-1 loss on training and test set

Question 2.2 (Binary classification using support vector machines)**.** The task is to perform binary classification using support vector machines (SVMs). For this exercise, use standard C-SVMs as introduced in the lecture. Employ radial Gaussian kernels of the form

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

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

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 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

$$C \in \{b^{-3}, b^{-2}, b^{-1}, 1, b, b^2, b^3, b^4, b^5, b^6\}$$

and

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

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 the data from `ML2016WeedCropTrain.csv` in the model selection and training process.

Report the values of C and γ you found in the model selection process. Compute the classification accuracy based on the 0-1 loss on the training data as well as on the test data.

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

Question 2.3 (Normalization)**.** As discussed, normalizing each component to zero mean and variance one (measured on the training set) is a common preprocessing step, which can remove undesired biases due to different scaling.

Using this normalization affects different classification methods differently. How does it effect logistic regression and SVMs as in question 2.2? If you think that there is a difference give a formal argument why. What will happen if a random forrest is used with and without normalization?

Redo the experiments from question 2.2 with normalization as preprocessing. What do you observe?

Deliverables: Results of SVMs with normalization and a short description of how you did the normalization; discussion of the difference when using normalization before applying logistic regression and SVMs with Gaussian kernel, including formal arguments; discussion of what would happen if a random forrest is used

Question 2.4 (Principal component analysis)**.** Perform a principal component analysis of `ML2016WeedCropTrain.csv`. Plot the eigenspectrum. How many components are necessary to "explain 90 % of the variance"? Visualize the data by a scatter plot of the first two principal components. Use different colors or symbols for weed and crop.

Deliverables: Description of software used; plot of the eigenspectrum; number of components necessary to explain 90 % of variance; scatter plot for first two principal components with different colors indicating the different classes

Question 2.5 (Clustering)**.** Perform 2-means clustering of `ML2016WeedCropTrain.csv`. For the submission, initialize the cluster centers with the first two data points in `ML2016WeedCropTrain.csv` (that is not a recommended intialization technique, but makes it easier to corrrect the exam). Plot the cluster centers projected to the first two principal components. That is, add the cluster centers to the plot from the previous question. Briefly discuss the results: Did you get meaningful clusters?

Deliverables: Description of software used; one plot with cluster centers; explain how you projected the high dimensional cluster centers down to two dimensions; short discussion of results

3 Generalization Bound for Learning with Multiple Feature Mappings

As we have seen in the course, a feature mapping can transform a non-linear classification problem into a linear one or, at least, bring it closer to linear separation. In practice, however, the most suitable transformation for a given dataset is typically unknown and it is a common practice to try multiple different feature mappings (multiple kernels) and select the one that works best. In this question we study how the selection influences the generalization properties.

Every answer in this question must be supported either by citation or by derivation (or both). You are allowed to cite our lecture notes, slides, and the course book (Abu-Mostafa et al., 2012). If you would like to cite other sources it is advisable to double-check with us their trustworthiness.

In this question we use the following terminology. A *high-probability bound* is a bound that holds with high probability for all hypotheses h in the relevant hypothesis space. We call a bound *trivial* if the statement can be obtained without making any calculations (for example, if a bound states that the probability of some event is bounded by 2, then it is a trivial bound, because we know without any calculations that probabilities are always bounded by 1).

The subpoints of this question progressively build on their predecessors. However, the question is built in such a way that even if you get stuck at one point you may still be able to proceed with subsequent points, so if it happens to you do not give up too early.

1. Consider a problem with d input features. Assume that we apply to the data the Q-th order polynomial transform $\Phi_Q(\mathbf{x})$. The transform is a vector of all monomials of degrees from zero up to Q (a monomial of degree q is a product of elements of $\mathbf{x}$ with their total degree summing up to q, for example, x_1x_2 is a monomial of degree 2 and $x_1^2x_2$ or $x_1x_2x_3$ are monomials of degree 3). There are two ways to define $\Phi_Q(\mathbf{x})$: without repetitions (see Pages 103-104 in Abu-Mostafa et al. (2012)) and with repetitions (see Page 8-34 in online Chapter 8 of Abu-Mostafa et al. (2012)). In Abu-Mostafa

et al. (2012) the same notation is used to denote both transforms, but to avoid confusions let us denote the transform with repetitions by $\Phi_Q^+(\mathbf{x})$. For example, if $d = 2$, then $\mathbf{x} = (x_1, x_2)$; $\Phi_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$; and $\Phi_2^+ = (1, x_1, x_2, x_1^2, x_1x_2, x_2x_1, x_2^2)$. Put attention that in $\Phi_2(\mathbf{x})$ we only have x_1x_2 and in $\Phi_2^+(\mathbf{x})$ we have both x_1x_2 and x_2x_1.

Put attention that the polynomial transform includes a leading coordinate "1". Therefore, the bias term can be absorbed into the first coordinate of $\mathbf{w}$ (the vector that defines a separating hyperplane) and we are talking about linear separation by *homogeneous* hyperplanes. Homogeneous hyperplanes are hyperplanes that are passing through the origin. A homogeneous hyperplane is defined by equation $\mathbf{w}^T\Phi_Q(\mathbf{x}) = 0$ (without the bias term b).

Let $\mathcal{H}_Q$ be the space of homogeneous linear separators in the space defined by $\Phi_Q(\mathbf{x})$. Prove that the VC-dimension of $\mathcal{H}_Q$ is bounded by $(Q + 1)d^Q$.

2. Write down a high-probability bound on $L(h)$ that holds for all $h \in \mathcal{H}_Q$.

3. Assume that the norm of all input points is bounded by 1, i.e., $\|\mathbf{x}\| \leq 1$. Derive a bound on $\|\Phi_Q(\mathbf{x})\|$.

   Hint: bound the norm of $\Phi_Q(\mathbf{x})$ by the norm of $\Phi_Q^+(\mathbf{x})$ (explain in one sentence why you can do that) and use polynomial kernel (see Page 8-34 in online Chapter 8 of Abu-Mostafa et al. (2012)) to bound the norm of $\Phi_Q^+(\mathbf{x})$.

4. Let $\mathbf{w}$ be a vector defining a homogeneous hyperplane in the space defined by $\Phi_Q(\mathbf{x})$. Let $h = \mathbf{w}$ denote the corresponding hypothesis. Derive a high-probability margin-based bound on $L(\mathbf{w})$ that depends on the norm of the vector $\mathbf{w}$. (The bound should look similar to Theorem 3.9 in Yevgeny's lecture notes. Note that Theorem 3.9 does not apply because of your answer to the previous point. Also, since we are working with homogeneous hyperplanes you should drop the "+1" term in Theorem 3.8, which comes from the bias term. I.e., you should replace $\mathcal{H}_\rho$ in Theorem 3.8 with $\mathcal{H}_\rho = \left\{\mathbf{w} : \|\mathbf{w}\| \leq \frac{1}{\rho}\right\}$ and the bound should be $d_{VC}(\mathcal{H}_\rho) \leq \lceil R^2/\rho^2 \rceil$.)

5. Let $\mathcal{H} = \bigcup_{Q=1}^{\infty} \mathcal{H}_Q$ be the union of all hypothesis spaces defined by polynomial transformations. What is the VC-dimension of $\mathcal{H}$?

6. Derive a high-probability margin-based bound on $L(\mathbf{w})$ that holds for all $\mathbf{w} \in \mathcal{H}$. (If you have failed to solve Point 4 you are allowed to use your result from Point 2 and derive a high-probability bound that holds for all $h \in \mathcal{H}$, but does not depend on the margin. If you do this correctly you will get partial points and you can use this simpler result to answer Points 7, 8, and 9.)

7. Assuming that you have a finite number of samples n, what is the largest polynomial degree Q as a function of n that still gives a non-trivial bound in the point above? (You can ignore all constants and just write the order of magnitude, as in the big-O notation.)

8. Explain why there is no contradiction between your answers to Points 5, 6, and the VC lower bound (Corollary 3.11 in Yevgeny's lecture notes).

9. Assume that the number of samples n and the confidence parameter δ are fixed. Explain the trade-off(s) that arise in minimization of the bound in Point 6.

References

Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from data.* AMLbook, 2012.

H. Aihara, C. A. Prieto, D. An, S. F. Anderson, É. Aubourg, E. Balbinot, T. C. Beers, A. A. Berlind, S. J. Bickerton, D. Bizyaev, et al. The eighth data release of the Sloan digital sky survey: first data from SDSS-III. *The Astrophysical Journal Supplement Series*, 193(2):29, 2011.

S. Olsen, J. Nielsen, and J. Rasmussen. Thistle detection. In *Scandinavian Conference on Image Analysis*, 2017. Submitted.

J. Rasmussen, J. Nielsen, S. I. Olsen, K. Steenstrup Petersen, J. E. Jensen, and J. Streibig. Droner til monitorering af flerårigt ukrudt i korn. Technical report, Miljøstyrelsen, Miljøministeriet, 2016. To be publish at `http://mst.dk/service/publikationer/`.

K. Stensbo-Smidt, F. Gieseke, C. Igel, A. Zirm, and K. S. Pedersen. Sacrificing information for the greater good: How to select photometric bands for optimal accuracy. *Monthly Notices of the Royal Astronomical Society*, 464(3):2577–2596, 2017. doi:10.1093/mnras/stw2476.