582744 Advanced Course in Machine Learning

Bonus exercises **Due May 17, 10:00 AM**

Rules:

1. Your submission is composed of two parts: (a) A single PDF file containing your answers to all questions, including both the pen and paper questions and the written answers and various plots for the programming questions. (b) a single compressed file (zip or tar.gz) containing your code (and nothing else). If your code is in a single file, it can be sent also a plain source code.
2. The submission should be sent directly to BOTH Arto (`arto.klami@cs.helsinki.fi`) and Aditya (`aditya.jitta@cs.helsinki.fi`).
3. All material must be renamed to your student ID. Always mention your name and student ID also in the written report.
4. The subject field of your email should be [AML][your student ID][exercise 8].
5. Please typeset your work using appropriate software such as LaTeX. However, there is no need to typeset the pen and paper answers – you can also include a scanned hand-written version.
6. The pen and paper exercises can alternatively be returned in paper form during the Tuesday lecture.

This set of exercises is due on Tuesday May 17th, before 10:00 AM.

1 Polynomial kernel (6 pts, programming)

The lecture slides of Lecture 8 defined the polynomial kernel as $k_{ij} = (\gamma + \mathbf{x}_i^T \mathbf{x}_j)^p$ for some positive *gamma* (in fact, $\gamma = 1$ was used on the slides). Another alternative would be to use the kernel $k_{ij} = (\mathbf{x}_i^T \mathbf{x}_j)^p$, corresponding to $\gamma = 0$. Write down the feature maps $\psi(\mathbf{x})$ for both alternatives for $p = 2$ and explain the main difference. What kind of an effect the additive constant has for the kernel values?

Now use polynomial kernels of degrees $p = 1, 2, 3$ for solving regression tasks on the data set https://www.cs.helsinki.fi/u/aklami/teaching/AML/data_bonus.csv, using the first 200 samples for training and the last 200 for validation. Each row has a training example with two-dimensional input and one-dimensional output (the last element). Use l_2 regularization and try out a few alternative values for γ. Describe the results as a function of the three parameters (p, the regularization constant λ and γ), so that you characterize the main effects and observations.

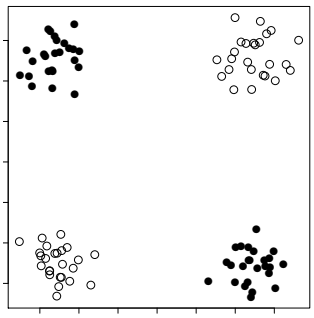
2 Sparsity (6 points)

Consider the regularized loss function $L(\theta_1, \theta_2) = (\theta_1 - 2)^2/4 + (\theta_2 - 2)^2 + \lambda|\boldsymbol{\theta}|$ (l_1-regularization), where $\boldsymbol{\theta} = [\theta_1, \theta_2]$.

Characterize the sparsity of the solution as a function of the regularization parameter λ. That is, find out for which values of λ the optimal solution has two non-zero elements, for which values it has one non-zero element, and for which values the result is the zero vector.

3 AdaBoost (12 points)

Consider the two-dimensional binary classification problem depicted below, corresponding to the XOR-problem. Linear classifiers ($\text{sign}(w_1 x_1 + w_2 x_2 + b)$) are here weak classifiers – they are clearly better than random but do not solve the whole classification problem.



Apply the AdaBoost algorithm (Algorithm 16.2 on page 561 of the course book) manually to construct the final classifier $f(\mathbf{x}) = \sum_{m=1}^{3} \beta_m f_m(\mathbf{x})$. That is, learn the first three weak classifiers to be used as parts of the final ensemble. Remember that the weights β_m are determined by the error rate of the classifier using $\beta_m = \frac{1}{2} \log \frac{1-\text{err}_m}{\text{err}_m}$ and the weights for the incorrectly classified training examples are multiplied by e^{β_m} for training the next classifier.

You should not use any learning algorithm to determine the decision boundaries – you can easily determine them manually once you know the weights for the samples (breaking ties arbitrarily). Besides drawing the decision boundaries, characterize how the weights of the samples change and provide the classifier weights β_m. What is the final classifier and how good is it? Would using more base learners improve it?

Hint: Each group has exactly 25 data points – no need to count. The axes are scaled so that the visual separation corresponds to the actual distance.