CS-E3210- Machine Learning Basic Principles
Home Assignment 2 - “Regression”

Your solutions to the following problems should be submitted as one single pdf which does not contain any personal information (student ID or name). The only rule for the layout of your submission is that for each problem there has to be exactly one separate page containing the answer to the problem. You are welcome to use the LaTeX-file underlying this pdf, available under `https://version.aalto.fi/gitlab/junga1/MLBP2017Public`, and fill in your solutions there.

Problem 1: “Plain Vanilla” Linear Regression

Consider a dataset $\mathbb{X}$ which is constituted of $N=10$ webcam snapshots with filename “MontBlanc*i*.png”, $i = 1, \ldots, N$, available in the folder “Webcam” at `https://version.aalto.fi/gitlab/junga1/MLBP2017Public`. Determine for each snapshot the feature vector $\mathbf{x}^{(i)} = (x_{\rm g}^{(i)}, 1)^T \in \mathcal{X}(=\mathbb{R}^2)$ with the normalized (by the number of image pixels) greenness $x_{\rm g}^{(i)}$. Moreover, determine for each snapshot the label $y^{(i)} \in \mathcal{Y}(=\mathbb{R})$ given by the duration (in minutes) after 07:00 am, at which the picture has been taken. We want to find (learn) a predictor $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ which allows to predict the value of $y^{(i)}$ directly from the value of the feature $x_{\rm g}^{(i)}$. To this end we consider only predictors belonging to the hypothesis space $\mathcal{H} = \{h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} \text{ for some } \mathbf{w} \in \mathbb{R}^2\}$. The quality of a particular predictor is measured by the mean squared error

$$\mathcal{E}(h(\cdot)|\mathbb{X}) := \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - h(\mathbf{x}^{(i)}))^2 = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2. \tag{1}$$

Note that the mean squared error is nothing but the empirical risk obtained when using the squared error loss $L((\mathbf{x}, y), h(\cdot)) = (y - h(\mathbf{x}))^2$ (cf. Lecture 2).

The optimal predictor $h_{\rm opt}(\cdot)$ is then

$$h_{\rm opt}(\cdot) = \underset{h(\cdot)\in\mathcal{H}}{\operatorname{argmin}}\, \mathcal{E}(h(\cdot)|\mathbb{X}). \tag{2}$$

We can rewrite this optimization problem in a fully equivalent manner in terms of the weight $\mathbf{w}$ representing a particular predictor $h^{(\mathbf{w})}(\cdot) \in \mathcal{H}$ as

$$\mathbf{w}_{\rm opt} = \underset{\mathbf{w}\in\mathbb{R}^2}{\operatorname{argmin}}\, \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2. \tag{3}$$

As can be verified easily, the optimal predictor $h_{\rm opt}(\cdot)$ (cf. (2)) is obtained as $h_{\rm opt}(\cdot) = h^{(\mathbf{w}_{\rm opt})}(\cdot)$ with the optimal weight vector $\mathbf{w}_{\rm opt}$ (cf. (3)).

Can you find a closed-form expression for the optimal weight $\mathbf{w}_{\rm opt}$ (cf. (3)) in terms of the vectors $\mathbf{x} = (x_{\rm g}^{(1)}, \ldots, x_{\rm g}^{(N)})^T \in \mathbb{R}^N$, and $\mathbf{y} = (y^{(1)}, \ldots, y^{(N)})^T \in \mathbb{R}^N$?

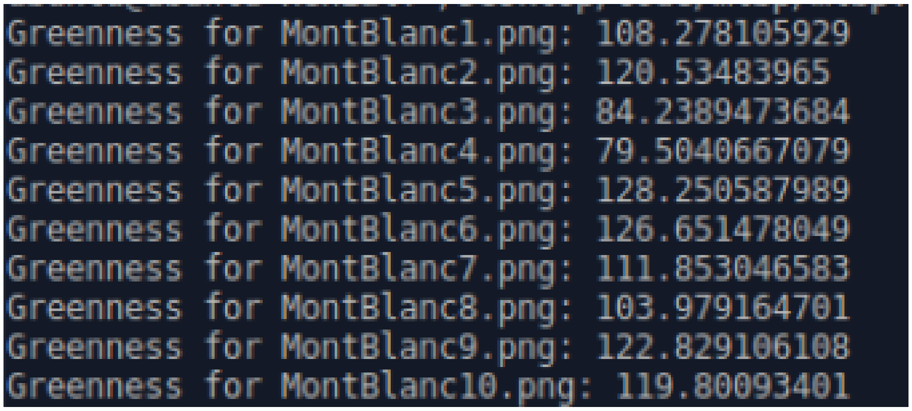
Answer.

A optimized and closed-form expression for w_{opt} is achieved by minimizing $\mathcal{E}(h(\cdot)|\mathbb{X})$, by solving where its gradient is 0. The below equations are based on what is found in chapter 5.1.4. of the Deep Learning book.

$$\frac{\delta\mathcal{E}(h(\cdot)|\mathbb{X})}{\delta w} = 0 \Leftrightarrow \bigtriangledown_w[\frac{1}{N}(y - Xw)] = 0 \Leftrightarrow \bigtriangledown_w[y^Ty - w^TX^Ty - y^TXw + w^TX^TXw] = 0$$
$$\Leftrightarrow -X^Ty + X^TXw = 0 \Leftrightarrow (X^TX)w = X^Ty \Leftrightarrow w = (X^TX)^{-1}X^Ty$$

Inserting the values of the images (visible in screen cap below) into the calculation gives us that for this case

$$\hat{w} = w_{opt} = (X^TX)^{-1}X^Ty$$
$$= [-439.99985992, 6.148722404]$$



```
Greenness for MontBlanc1.png: 108.278105929
Greenness for MontBlanc2.png: 120.53483965
Greenness for MontBlanc3.png: 84.2389473684
Greenness for MontBlanc4.png: 79.5040667079
Greenness for MontBlanc5.png: 128.250587989
Greenness for MontBlanc6.png: 126.651478049
Greenness for MontBlanc7.png: 111.853046583
Greenness for MontBlanc8.png: 103.979164701
Greenness for MontBlanc9.png: 122.829106108
Greenness for MontBlanc10.png: 119.80093401
```

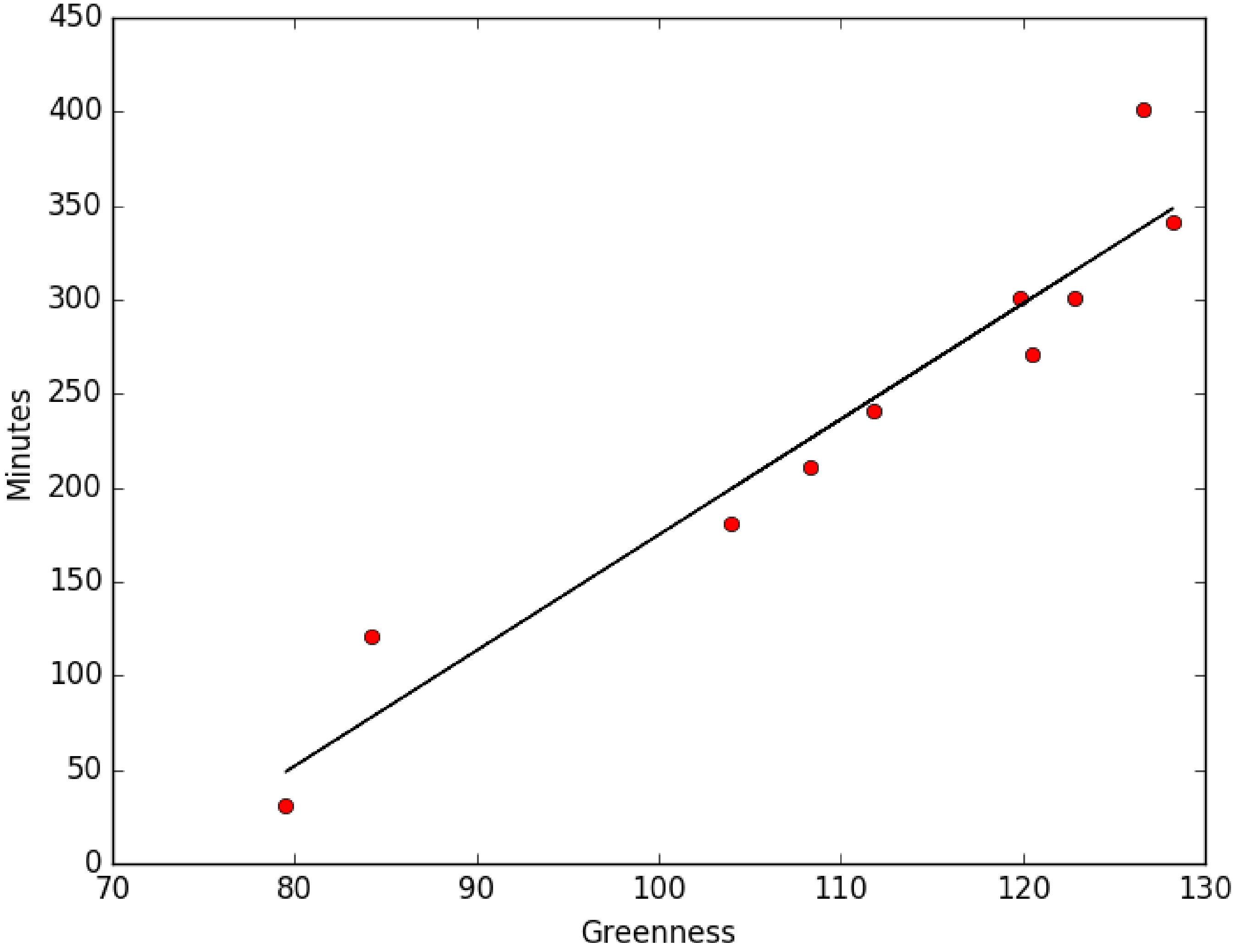
Problem 2: “Plain Vanilla” Linear Regression - Figure

Reconsider the setup of Problem 1 and generate a plot with horizontal (vertical) axis representing greenness x_{g} (label y), which depicts the optimal predictor $h_{\mathrm{opt}}(\cdot)$ (cf. (2)) and also contains the data points $(x_{\mathrm{g}}^{(i)}, y^{(i)})$ for $i = 1, \ldots, N$. Do you consider it feasible to predict the daytime accurately from the greenness?
Answer.

Judging by the graph (found below), it seems pretty feasible on first look to predict the time of day from greenness, however looking at mean squared error for the case gives us:

$$\mathcal{E}(h(\cdot)|\mathbb{X}) = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - h(x^{(i)}))^2 = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - w^T x^{(i)})^2 = 783.424339374$$

which does not seem not very accurate in the end.



Problem 3: Regularized Linear Regression

We consider again the regression problem of Problem 1, i.e., predicting the daytime of a webcam snapshot based on the feature vector $(x_{\rm g}, 1)^T$. The prediction is of the form $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ with some weight vector $\mathbf{w} \in \mathbb{R}^2$. Assume that we only have snapshots which are taken within 7 hours after 07:00 am, i.e.,the value of the label y cannot exceed 420. Therefore, it makes sense to somehow constraint the norm of the weight vector $\mathbf{w}$ to exclude unreasonable predictions. To this end, we augment the mean squared error (1) with the "regularization term" $\lambda\|\mathbf{w}\|^2$ which penalizes "atypical" values for the weight vector. The optimal predictor $h_{\rm opt}(\cdot)$ using this regularization term is then given by

$$h_{\rm opt,r}(\cdot) = \operatorname*{argmin}_{h^{(\mathbf{w})}(\cdot)\in\mathcal{H}} \big(\mathcal{E}(h^{(\mathbf{w})}(\cdot)|\mathbb{X}) + \lambda\|\mathbf{w}\|^2\big). \tag{4}$$

Again, we can rewrite this optimization problem in a fully equivalent manner in terms of the weight $\mathbf{w}$ representing a particular predictor $h^{(\mathbf{w})}(\cdot) \in \mathcal{H}$ as

$$\mathbf{w}_{\rm opt,r} = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^2} \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \lambda\|\mathbf{w}\|^2. \tag{5}$$

As can be verified easily, the optimal predictor $h_{\rm opt,r}(\cdot) \in \mathcal{H}$ solving (2) is obtained as $h_{\rm opt,r}(\cdot) = h^{(\mathbf{w}_{\rm opt,r})}(\cdot)$ with the optimal weight vector $\mathbf{w}_{\rm opt,r}$ which is the solution of (3). Can you find a closed-form solution for the optimal weight $\mathbf{w}_{\rm opt,r}$ (cf. (3)) in terms of the vectors $\mathbf{x} = (x_{\rm g}^{(1)}, \ldots, x_{\rm g}^{(N)})^T \in \mathbb{R}^N$, and $\mathbf{y} = (y^{(1)}, \ldots, y^{(N)})^T \in \mathbb{R}^N$ and λ? **Answer:**

A closed-form solution can be achieved with the following, providign an optimal w_{opt}:

$$\begin{aligned} \frac{\mathcal{E}(h(\cdot)|\mathbb{X}) + \lambda||w||^2}{\delta w} = 0 \Leftrightarrow \bigtriangledown_w[\frac{1}{N}(y - Xw)^T(y - Xw) + N\lambda w^T w] = 0 \\ \Leftrightarrow \bigtriangledown_w[y^y - w^T X^T y - y^T Xw + w^T X^T Xw + N\lambda w^T w] = 0 \\ \Leftrightarrow -X^T y + X^T Xw + N = \Leftrightarrow (X^T X + N\lambda I)w = X^T y \\ \Leftrightarrow w = (X^T X + N\lambda I)^{-1} X^T y \end{aligned}$$

Based on the above, the optimal paramater is: $\hat{w} = w_{opt,r} = (X^T X + N\lambda I)^{-1} X^T y$

Problem 4: Regularized Linear Regression - Figure

Reconsider the setup of Problem 3 and generate a plot with horizontal (vertical) axis representing greenness $x_{\rm g}$ (label y) which contains the data points $(x_{\rm g}^{(i)}, y^{(i)})$, for $i = 1, \ldots, N$, and depicts the optimal predictor $h_{\rm opt,r}(\cdot)$ (cf. (4)) for the two particular choices $\lambda = 2$ and $\lambda = 5$. Which choice for λ seems to be better for the given task?

Answer:

Problem 5: Gradient Descent for Linear Regression

Consider the same dataset as in Problem 1, i.e., the set of $N = 10$ webcam snapshots which are labeled by the daytime $y^{(i)}$ when the image has been taken. As in Problem 1, we are interested in predicting the daytime directly from the image. However, by contrast to Problem 1 where we only used the greenness $x_{\rm g}^{(i)}$ of the i-th image, we know use the green intensity values for the upper-left area consisting of 100×100 pixels, which we stack into the feature vector $\mathbf{x}^{(i)} \in \mathbb{R}^{d}$. What is the length d of the feature vector $\mathbf{x}^{(i)}$ here? Based on the feature vector, we predict the daytime by a predictor of the form $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^{T}\mathbf{x}$ with some weight vector $\mathbf{w} \in \mathbb{R}^{d}$. The optimal predictor is obtained by solving an empirical risk minimization problem of the form (2), or directly in terms of the weight vector, (3). This minimization problems can be solved by a simple but powerful iterative method known as gradient descent (GD):

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha \nabla f(\mathbf{w}^{(k)}) \tag{6}$$

with some positive step size $\alpha > 0$ and the mean-squared error cost function (cf. (1))

$$f(\mathbf{w}) := \mathcal{E}(h^{(\mathbf{w})}|\mathbb{X}) \overset{(1)}{=} \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^{T}\mathbf{x}^{(i)})^{2}.$$

In order to implement the GD iterations (6), we need to compute the gradient $\nabla f(\mathbf{w})$. Can you find a simple closed-form expression for the gradient of $f(\mathbf{w})$ at a particular weight vector $\mathbf{w}$?

The performance of GD depends crucially on the particular value chosen for the step size α in (6). Try out different choices for the step size α and, for each choice plot the evolution of the empirical risk $\mathcal{E}(h^{(\mathbf{w}^{(k)})}|\mathbb{X})$ as a function of iteration number k into one single figure. Use the initialization $\mathbf{w}^{(0)} = \mathbf{0}$ for the GD iterations for each run.

Another crucial issue when using GD is the question of when to stop iterating (6). Can you state a few stopping criteria that indicate when it would be reasonable to stop iterating (6)?

Answer:

Problem 6: Gradient Descent for Regularized Linear Regression

Redo Problem 5 by using regularized linear regression (cf. Problem 3) instead of linear regression.
Answer:

Problem 7: Kernel Regression

Consider the data set of Problem 1, i.e., the set of $N = 10$ webcam snapshots. Let us now represent each webcam snapshot by the single feature $x^{(i)} = x_{\rm g}^{(i)}$, i.e., the total greenness of the ith snapshot. We aim at predicting the daytime $y^{(i)}$ based solely on the greenness. In contrast to Problem 1 and Problem 2 we will now use a different hypothesis space of predictors. In particular, we only consider predictors out of the hypothesis space

$$\mathcal{H} = \left\{ h^{(\sigma)}(\cdot) : \mathbb{R} \rightarrow \mathbb{R} : h^{(\sigma)}(x) = \sum_{i=1}^{N} y^{(i)} \frac{K_{\sigma}(x, x^{(i)})}{\sum_{l=1}^{N} K_{\sigma}(x, x^{(l)})} \right\} \tag{7}$$

with the "kernel"

$$K_{\sigma}(x, x^{(i)}) = \exp\left(- \frac{1}{2} \frac{(x - x^{(i)})^2}{\sigma^2} \right). \tag{8}$$

Try out predicting the daytime $y^{(i)}$ using the greenness $x_{\rm g}^{(i)}$ using a predictor $h^{(\sigma)}(\cdot) \in \mathcal{H}$ using the choices $\sigma \in \{1, 5, 10\}$. Generate a plot with horizontal (vertical) axis representing greenness $x_{\rm g}$ (label y), which depicts the predictor $h^{(\sigma)}(\cdot)$ for $\sigma \in \{1, 5, 10\}$ and also contains the data points $(x_{\rm g}^{(i)}, y^{(i)})$. Which choice for σ achieves the lowest mean squared error $\mathcal{E}(h^{(\sigma)} | \mathbb{X})$ (cf. (1)) ?

Answer:

Problem 8: Linear Regression using Feature Maps

Consider a regression problem, where we aim at predicting the value of a real-valued label or target or output variable $y \in \mathbb{R}$ of a data point based on a single feature $x \in \mathbb{R}$ of this data point. We assume that there is some true underlying functional relationship between feature x and output y, i.e., $y = h^{*}(x)$ with some unknown function (hypothesis). All we know about this true underlying functional relationship is that

$$h^{*}(x) = 0 \text{ for any } x \notin [0,10], \text{ and } |h^{*}(x') - h^{*}(x'')| \leq 10^{-3}|x' - x''| \text{ for any } x', x'' \in [0,10]. \tag{9}$$

We apply then a feature map $\boldsymbol{\phi} : \mathbb{R} \rightarrow \mathbb{R}^{n}$, with some suitable chosen dimension n, which transforms the original feature x into a modified feature vector $\boldsymbol{\phi}(x) = (\phi_1(x), \ldots, \phi_n(x))^{T}$. We use the transformed features $\boldsymbol{\phi}(x)$ to predict the label y using the predictor $h^{(\mathbf{w})}(x) = \mathbf{w}^{T}\boldsymbol{\phi}(x)$ with some weight vector $\mathbf{w} \in \mathbb{R}^{n}$. Note that the so defined predictor $h^{(\mathbf{w})}$ is linear only w.r.t. the high-dimensional features $\boldsymbol{\phi}(x)$, but typically a non-linear function of the original feature x. Is there a feature map $\boldsymbol{\phi}$ which allows to approximate the true hypothesis $h^{*}(\cdot)$ (which satisfies (9)) by some predictor $h^{(\mathbf{w}_0)}(x) = \mathbf{w}_0^{T}\boldsymbol{\phi}(x)$ with a suitably chosen weight $\mathbf{w}_0$? In particular, is there a feature map $\boldsymbol{\phi}$ and weight vector $\mathbf{w}_0 \in \mathbb{R}^{n}$ such that $|h^{(\mathbf{w}_0)}(x) - h^{*}(x)| \leq 10^{-3}$ for all $x \in \mathbb{R}$?

Answer: