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 each problem has to correspond to one single page, which has to include the problem statement on top and your solution below. 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

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

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

$$h_{\rm opt}(\cdot) = \operatorname*{argmin}_{h(\cdot)\in\mathcal{H}} \mathcal{E}(h(\cdot)|\mathbb{X}). \quad (2)$$

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

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

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.

We can try to estimate duration from 7:00am as a linear combination of the greeness of the picture and a constant term, *i.e.*, $y^i = w_1 x_g^i + w_0 = (x_g^i, 1)(w_1, w_0)^T$. For multiple measurements this can be written as

$$\begin{aligned} y_1 &= w_1 x_g^1 + w0 \\ y_2 &= w_1 x_g^2 + w0 \\ &\vdots \\ y_n &= w_1 x_g^n + w0, \end{aligned} \quad (4)$$

which can be written in matrix form as

$$\mathbf{y} = A\mathbf{w}, \quad (5)$$

where $\mathbf{y} = (y_1, y_2, y_3, \cdots, y_n)$, $A = (x_g^1, 1; x_g^2, 1; \cdots; x_g^n, 1)$ and $\mathbf{w} = (w_1, w_0)^T$. From equation 5 the optimal weight vector $\mathbf{w}_{\rm opt}$ can be found using the well known least squares solution

$$\mathbf{w}_{\rm opt} = (A^T A)^{-1} A^T \mathbf{y}. \quad (6)$$

We can reformulate this using $B = (\mathbf{x}\ \mathbf{c})$ ($\mathbf{c}$ is a $n \times 1$ column vector) which have a pseudo-inverse $B^+ = B^T(BB^T)^{-1}$. The inverse $(BB^T)^{-1}$ can be written as $(\mathbf{x}\mathbf{x}^T + cc^T)^{-1}$ which can be further written using Sherman-Morrison formula as

$$(\mathbf{x}\mathbf{x}^T + \mathbf{c}\mathbf{c}^T)^{-1} = (\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{c}\mathbf{u}), \quad (7)$$

where $\mathbf{u} = \frac{\mathbf{c}^T(\mathbf{x}\mathbf{x}^T)^{-1}}{1+\mathbf{v}^T(\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{v}}$. Inserting this into B^+ gives

$$\begin{aligned} \mathbf{w}_{\rm opt} &= \mathbf{B}^+\mathbf{y} \\ &= (\mathbf{x}\ \mathbf{c})^T(\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{v}\mathbf{u})\mathbf{y}. \end{aligned} \quad (8)$$

The measured labels, *i.e.*, duration since 7:00pm and the greeness of images are shown in figure 1 in the next section.

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.

The durations from 7:00am in minutes can be read from each picture and the picture data can be read using SciPy. The data points (x_g^i, y^i) as well as the predicted durations, obtaines using eq. (6), are shown in figure 1. The greeness of the image is calculated using equation $x_g = \frac{1}{N}\sum_{j=1}^{N}(g[j] - (1/2)(r[j] + b[j]))$ where N is the total number of pixels in picture i. According to figure 1 the time can be rather well estimated from the greenness of the image in the given time interval.

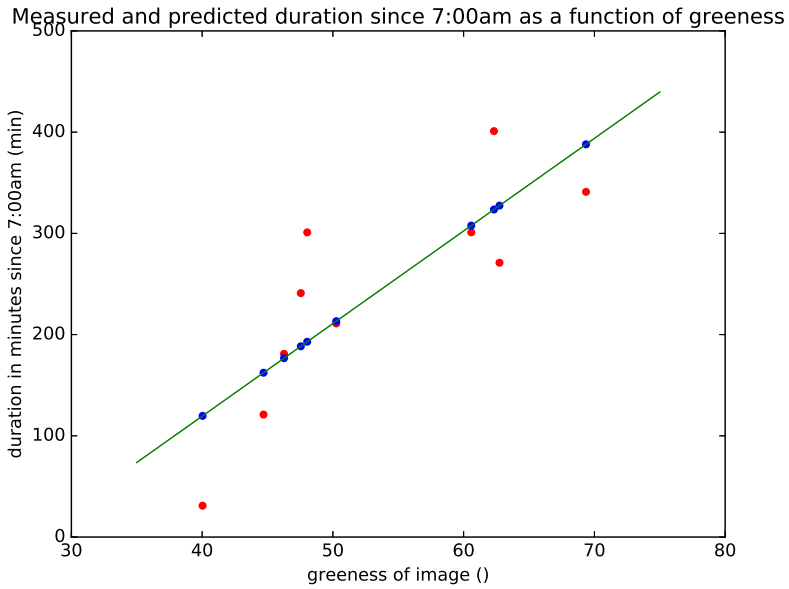


Figure 1: Measured (red dots) and predicted (blue dots) durations since 7:00am as a function of image greenness. The green line illustrates the optimal linear fit.

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

$$h_{\rm opt,r}(\cdot) = \operatorname*{argmin}_{h(\cdot)\in\mathcal{H}} \mathcal{E}(h(\cdot)|\mathbb{X}) + \lambda\|\mathbf{w}\|^2. \tag{9}$$

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

$$\mathbf{w}_{\rm opt,r} = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^2} \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 + \lambda\|\mathbf{w}\|^2. \tag{10}$$

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: The solution for regularized problem can be written as

$$\begin{aligned} \mathbf{w}_{\rm opt} &= (A^TA + \lambda^2 I)^{-1}A^T y \\ &= (B^TB + \lambda^2 I)^{-1}B^T\mathbf{y}, \end{aligned} \tag{11}$$

where matrices A and B are defined in section Problem 1:. The inverse $(B^TB + \lambda^2 I)^{-1}$ can be written, using Woodbury matrix identity, in the form

$$(B^TB + \lambda^2 I)^{-1} = (BB^T)^{-1} - \lambda^2(BB^T)^{-1}(I + \lambda^2(BB^T)^{-1})^{-1}(BB^T)^{-1}. \tag{12}$$

Inserting (12) into eq. (11) and using identity $(BB^T)^{-1} = (\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{c}\mathbf{u})$ from section Problem 1: the solution for $\mathbf{w}_{\rm opt}$ can be finally written in the form

$$\mathbf{w}_{\rm opt} = \left((\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{c}\mathbf{u}) - \lambda^2(\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{c}\mathbf{u})(I - \lambda^2(\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{c}\mathbf{u}))^{-1}(\mathbf{x}\mathbf{x}^T)^{-1}(I - \mathbf{c}\mathbf{u})\right)(\mathbf{x}\,\mathbf{c})^T\mathbf{y}. \tag{13}$$

Problem 4: Regularized Linear Regression - Figure

Reconsider the setup of Problem 3 and generate a plot with horizontal (vertical) axis representing greenness x_{g} (label y) which contains the data points $(x_{\mathrm{g}}^{(i)}, y^{(i)})$, for $i = 1, \ldots, N$, and depicts the optimal predictor $h_{\mathrm{opt,r}}(\cdot)$ (cf. (9)) for the two particular choices $\lambda = 2$ and $\lambda = 5$. Which choice for λ seems to be better for the given task?

Answer:

The data points (x_g^i, y^i) as well as the predicted durations (blue dots corresponding to $\lambda = 2$, red dots corresponding ot $\lambda = 5$), obtained using eq. (11), are shown in figure 2. According to figure 2 clearly the best prediction is given by the unregularized solution. Both regularized solutions with $\lambda = 2$ and $\lambda = 5$ performs nearly identically, the $\lambda = 2$ solution being slightly better.

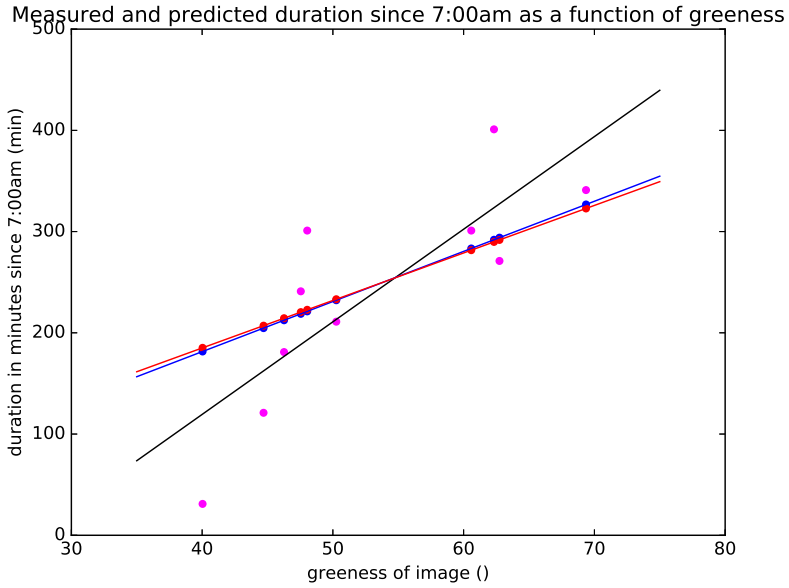


Figure 2: Regularized solutions. The blue dots corresponds to solution $\lambda = 2$ and the red dots corresponds to $\lambda = 5$. The black line represents the optimal prediction without regularization.

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 $x_{\mathrm{g}}^{(i)}$ 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):

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha \nabla f(\mathbf{w}) \tag{14}$$

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 (14), 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 (14). 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 (14). Can you state a few stopping criteria that indicate when it would be reasonable to stop iterating (14)?

Answer:

The 100×100 region can be stacked to 10000×1 feature vector and by taking into accout the constant term we get a total of $d = 10001$ elements in the feature vector $\mathbf{x}^{(i)}$. The error cost function $f(\mathbf{w})$ can be written equivalently in the form $f(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y^i - \mathbf{x}^{(i)T}\mathbf{w})$. It is good to notice that $\mathbf{x}_1^{(i)} = 1$. Taking partial derivative of $f(\mathbf{w})$ with respect to w_j gives

$$\begin{aligned}\frac{\partial f}{\partial w_j} &= -\frac{2}{N}\sum_{i=1}^{N} x_j^{(i)}(y^i - \mathbf{x}^{(i)T}) \\ &= -\frac{2}{N}X_j(\mathbf{y} - X^T\mathbf{w}),\end{aligned} \tag{15}$$

where $x_j^{(i)}$ denotes the j-th element of feature vector i, $X = (\mathbf{x}^{(1)}\,\mathbf{x}^{(2)}\ldots\mathbf{x}^{(N)})$ and X_j is the j-th row of X. The the total gradient vector then becomes $\nabla f(\mathbf{w}) = -\frac{2}{N}X(\mathbf{y} - X^T\mathbf{w})$. Figure 3 shows convergence of the gradient descent method with different alphas. Possible stopping crieteria for the iterationa are difference in error functions between iterations, relative difference between erro functions between iterations and the maximum number of iteration (used here).

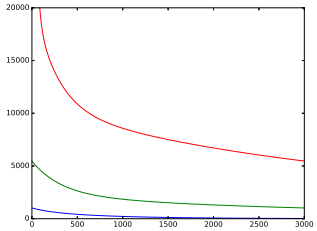


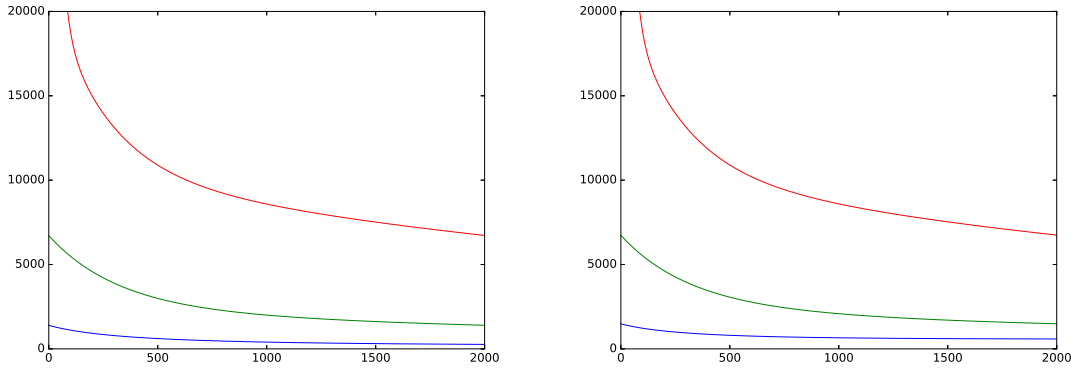
Figure 3: Empirical risk as a function of iteration number. The red curve corresponds to $\alpha = 0.000001$, the green curve corresponds to $\alpha = 0.00001$ and the blue curve corresponds to $\alpha = 0.0001$

Problem 6: Gradient Descent for Regularized Linear Regression

Redo Problem 5 for regularized linear regression (Problem 3) instead of linear regression (Problem 1).
Answer:
When the regularized regression is used the total gradient can be written in the form

$$\nabla f(\mathbf{w}) = -\frac{2}{N} X(\mathbf{y} - X^T\mathbf{w}) + 2\lambda\mathbf{w}. \tag{16}$$

Figure 4 shows the convergence of gradient-descent as a function of iteration number for lambda values $\lambda = 2$ and $\lambda = 5$.



(a) $\lambda = 2$ (b) $\lambda = 5$

Figure 4: Empirical risks as a function of iteration number for two different values of λ (left $\lambda = 2$, right $\lambda = 5$). The red curve corresponds to $\alpha = 0.000001$, the green curve corresponds to $\alpha = 0.00001$ and the blue curve corresponds to $\alpha = 0.0001$

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

$$\mathcal{H} = \left\{ h^{(\sigma)}(\cdot) : \mathbb{R} \rightarrow \mathbb{R} : h^{(\sigma)}(x) = \sum_{i=1}^{N} y^{(i)} \frac{K_\sigma(x, x^{(i)})}{\sum_{l=1}^{N} K_\sigma(x, x^{(l)})} \right\} \tag{17}$$

with the "kernel"

$$K_\sigma(x, x^{(i)}) = \exp\left(-\frac{1}{2} \frac{(x - x^{(i)})^2}{\sigma^2} \right). \tag{18}$$

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:

The predictor $h^{(\sigma)}$ for σ values $\{1, 5, 10\}$ is shown in figure 5. Clearly the lowest mean squared error is obtained with $\sigma = 1$.

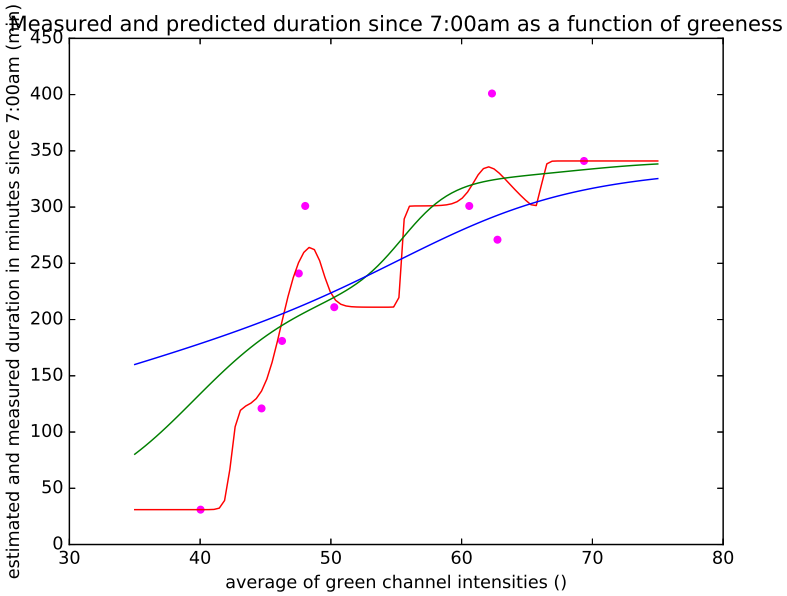


Figure 5: The predictor $h^{(\sigma)}$ for different values of σ. The red curve corresponds to $\sigma = 1$, the green curve corresponds to $\sigma = 5$ and the blue curve corresponds to $\sigma = 10$

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

$$h^*(x) = 0 \text{ for any } x \notin [0, 10], \text{ and } |h^*(x') - h^*(x'')| \leq 10^{-3}|x' - x''| \text{ for any } x', x'' \in [0, 10]. \tag{19}$$

We apply then a feature map $\boldsymbol{\phi} : \mathbb{R} \to \mathbb{R}^n$, 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 $\boldsymbol{\phi}$ such that for any hypothesis $h^*(\cdot)$, which satisfies (19), there is always a 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: