IFT6390 Fundamentals of Machine Learning

Professor: Ioannis Mitliagkas

Solution for Homework 2

- This homework must be done in groups of 2. Make sure to write the name of both team members on top of your report, and as a comment, on top of every file you submit.
- We ask you to submit a report as a pdf file. You should also submit every source code file you made or adapted. The practical part should be coded in python (with the numpy and matplotlib libraries). You are of course encouraged to draw inspiration from what was done in lab sessions.
- You can submit you python code as a Jupyter notebook (.ipynb). To write math in your report, you may use softwares such as LaTeX; LyX; Word; or even write the equations directly in the notebook with the MathJax syntax. In any case, you should export your report to a pdf file that you will submit.
- You should hand in your report via StudiUM. Only one of the team-mates should submit the report. If you have to submit lots of files, you may also compress them into a .zip or .tar.gz archive and upload this file.

1 Linear and non-linear regularized regression (50 pts)

1.1 Linear Regression

Let's consider a regression problem for which we have a training dataset D_n with n samples (input, target):

$$D_n = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)}\}$$

with $\mathbf{x}^{(i)} \in \mathbb{R}^d$, and $t^{(i)} \in \mathbb{R}$.

The linear regression assumes a parametrized form for the function f which predicts the value of the target from a new data point $\mathbf{x}$. (More precisely, it seeks to predict the expectation of the target variable conditioned on the input variable $f(\mathbf{x}) \simeq \mathbb{E}[t|\mathbf{x}]$.)

The parametrization is a linear transformation of the input, or more precisely an *affine* transformation.

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

1. Undergraduates 5 pts Graduates 2.5 pts
   Question. Precise this model's set of parameters θ, as well as the nature and dimensionality of each of them.
   Answer. The parameters are $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

2. Undergraduates 5 pts Graduates 2.5 pts
   Question. The loss function typically used for linear regression is the quadratic loss:
   $$L((\mathbf{x}, t),\ f) = (f(\mathbf{x}) - t)^2$$
   We are now defining the **empirical risk** $\hat{R}$ on the set D_n as the **sum** of the losses on this set (instead of the average of the losses as it is sometimes defined). Give the precise mathematical formula of this risk.
   Answer. $\hat{R} = \sum_{i=1}^{n}(f(\mathbf{x}^{(i)}) - t^{(i)})^2$

3. Undergraduates 5 pts Graduates 2.5 pts
   Question. Following the principle of Empirical Risk Minimization (ERM), we are going to seek the parameters which yield the smallest quadratic loss. Write a mathematical formulation of this minimization problem.
   Answer. $\arg\min_{\mathbf{w},b} \hat{R}(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}^{(i)} + b - t^{(i)})^2$

4. Undergraduates 10 pts Graduates 5 pts
   Question. A general algorithm for solving this optimization problem is gradient descent. Give a formula for the gradient of the empirical risk with respect to each parameter.
   Answer.
   $$\nabla_{\mathbf{w}}\hat{R} = 2\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b - t^{(i)}\right)\mathbf{x}^{(i)}$$

(the answer using partial derivatives instead of vectorized form is ok as well)

$$\frac{\partial \hat{R}}{\partial b} = 2\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b - t^{(i)}\right)$$

5. Undergraduates bonus: 5 pts Graduates 5 pts
   Question. Define the error of the model on a single point $(\mathbf{x}, t)$ by $f(\mathbf{x}) - t$. Explain in English the relationship between the empirical risk gradient and the errors on the training set.
   Answer. The gradient with respect to the bias is the sum of the errors while the gradient with respect to the weights is the weighted sum of the data points where the weights are given by the errors.

1.2 Ridge Regression

Instead of $\hat{R}$, we will now consider a **regularized empirical risk:** $\tilde{R} = \hat{R} + \lambda\mathcal{L}(\theta)$. Here $\mathcal{L}$ takes the parameters θ and returns a scalar penalty. This penalty is smaller for parameters for which we have an a priori preference. The scalar $\lambda \geq 0$ is an **hyperparameter** that controls how much we favor minimizing the empirical risk versus this penalty. Note that we find the unregularized empirical risk when $\lambda = 0$.

We will consider a regularization called *Ridge*, or *weight decay* that penalizes the squared norm (ℓ^2 norm) of the weights (but not the bias): $\mathcal{L}(\theta) = \|\mathbf{w}\|^2 = \sum_{k=1}^{d} \mathbf{w}_k^2$. We want to minimize $\tilde{R}$ rather than $\hat{R}$.

1. Undergraduates 7.5 pts Graduates 5 pts
   Question. Express the gradient of $\tilde{R}$. How does it differ from the unregularized empirical risk gradient?
   Answer.

$$\nabla_{\mathbf{w}}\tilde{R} = \nabla_{\mathbf{w}}\hat{R} + \lambda\nabla\mathcal{L}(\theta) = 2\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b - t^{(i)}\right)\mathbf{x}^{(i)} + 2\lambda\mathbf{w}$$

   (the answer using partial derivatives instead of vectorized form is ok as well)

$$\frac{\partial \tilde{R}}{\partial b} = \frac{\partial \hat{R}}{\partial b} = 2\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b - t^{(i)}\right)$$

The regularization only affect the gradient w.r.t. $\mathbf{w}$, where it simply differs by an additional term corresponding to the gradient of the regularizer. (extra) After each gradient descent step, the vector $\mathbf{w}$ is pushed towards 0 by subtracting a small portion of the old values from the new ones for each components of $\mathbf{w}$.

2. Undergraduates 7.5 pts Graduates 5 pts
   Question. Write down a detailed pseudocode for the training algorithms that finds the optimal parameters minimizing $\tilde{R}$ by gradient descent. To keep it simple, use a constant step-size η.
   Answer.

   - initialize $\mathbf{w}$, b randomly
   - Until some stopping criterion, repeat

     $\mathbf{w} \leftarrow \mathbf{w} - 2\eta \left(\sum_{i=1}^{n} \left(\mathbf{w}^T \mathbf{x}^{(i)} + b - t^{(i)} \right) \mathbf{x}^{(i)} + \lambda \mathbf{w} \right)$

     $b \leftarrow b - 2\eta \sum_{i=1}^{n} \left(\mathbf{w}^T \mathbf{x}^{(i)} + b - t^{(i)} \right)$

   Any reasonable stopping criterion and initialization is ok to get the points. However, usually when we want to minimize error on unseen data, we use a validation set to do early stopping (another form of regularization!). Expressing the algorithm in a more abstract form using a variable θ to gather the parameters $\mathbf{w}, b$ is ok as well.

3. Undergraduates bonus: 7.5 pts Graduates 5 pts
   Question. There happens to be an analytical solution to the minimization problem coming from linear regression (regularized or not). <u>Assuming no bias (meaning $b = 0$)</u>, find a matrix formulation for the empirical risk and its gradient, with the matrix $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{(1)} & \dots & \mathbf{x}_d^{(1)} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_1^{(n)} & \dots & \mathbf{x}_d^{(n)} \end{pmatrix}$ and the vector $\mathbf{t} = \begin{pmatrix} t^{(1)} \\ \vdots \\ t^{(n)} \end{pmatrix}$.
   Answer. We start with the empirical risk in matrix form:

$$R = ||Xw - t||_2^2 + \lambda ||w||_2^2 = (Xw - t)^T (Xw - t) + \lambda w^T w$$

and differentiate it

$$\begin{aligned}\nabla_w R &= \nabla_w[((Xw - t)^T(Xw - t) + \lambda w^T w] \\ &= \nabla_w[w^T X^T X w - 2w^T X^T t - t^T t + \lambda w^T w] \\ &= 2X^T X w - 2X^T t + 2\lambda w \\ &= 2(X^T X + \lambda I)w - 2X^T t\end{aligned}$$

Any response that gets to the right answer with some reasonable amount of justification got full marks. Wrong answers with an attempt to derive got partial marks.

4. Undergraduates bonus: 7.5 pts Graduates 5 pts
Question. Derive a matrix formulation of the analytical solution to the ridge regression minimization problem by expressing that the gradient is null at the optimum. What happens when $N < d$ and $\lambda = 0$?
Answer. Setting the above gradient to 0:

$$\begin{aligned}2(X^T X + \lambda I)w - 2X^T t = 0 \\ \implies w = (X^T X + \lambda I)^{-1} X^T t\end{aligned}$$

Regarding the last part of the question acceptable answers include

- there is an infinite number of minimizers / the linear system is indeterminate
- we need to use the pseudo-inverse instead

The main thing is that you notice that $X^T X$ is not invertible in this case.

1.3 Regression with a fixed non-linear pre-processing

We can make a non-linear regression algorithm by first passing the data through a fixed non-linear filter: a function $\phi(\mathbf{x})$ that maps $\mathbf{x}$ non-linearly to a higher dimensional $\tilde{\mathbf{x}}$.

For instance, if $x \in \mathbb{R}$ is one dimensional, we can use the polynomial transformation:

$$\tilde{x} = \phi_{\text{poly}^k}(x) = \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^k \end{pmatrix}$$

We can then train a regression, not on the $(x^{(i)}, t^{(i)})$ from the initial training set D_n, but on the transformed data $(\phi(x^{(i)}), t^{(i)})$. This training finds the parameters of an affine transformation f

To predict the target for a new training point x, you won't use $f(x)$ but $\tilde{f}(x) = f(\phi(x))$.

1. Undergraduates 5 pts Graduates 2.5 pts
   Question. Write the detailed expression of $\tilde{f}(x)$ when x is one-dimensional (univariate) and we use $\phi = \phi_{\text{poly}^k}$.
   Answer. $\tilde{f}(x) = \sum_{i=1}^{k} w_i x^i + b$

2. Undergraduates 5 pts Graduates 2.5 pts
   Question. Give a detailed explanation of the parameters and their dimensions.
   Answer. weight vector $\mathbf{w} \in \mathbb{R}^k$ and bias term $b \in \mathbb{R}$

3. Undergraduates bonus: 2.5 pts Graduates 2.5 pts
   Question. In dimension $d \geq 2$, a polynomial transformation should include not only the individual variable exponents x_i^j, for powers $j \leq k$, and variables $i \leq d$, but also all the interaction terms of order k and less between several variables (e.g. terms like $x_i^{j_1} x_l^{j_2}$, for $j_1 + j_2 \leq k$ and variables $i, l \leq d$). For $d = 2$, write down as a function of each of the 2 components of x the transformations $\phi_{\text{poly}^1}(x)$, $\phi_{\text{poly}^2}(x)$, and $\phi_{\text{poly}^3}(x)$.
   Answer.

   $\phi_{\text{poly}^1}(x) = (x_1, x_2)$
   $\phi_{\text{poly}^2}(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$
   $\phi_{\text{poly}^3}(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^3, x_2^3, x_1 x_2^2, x_1^2 x_2)$

4. Undergraduates bonus: 7.5 pts Graduates 5 pts
   Question. What is the dimensionality of $\phi_{\text{poly}^k}(x)$, as a function of d and k?

Answer. The answer is $\binom{d+k}{d} - 1$ or equivalently $\binom{d+k}{k} - 1$ which can be proven using the *stars and bars* technique. See this blog article. If k is small w.r.t. d (i.e. k is a constant) this grows at a rate of $\mathcal{O}(d^k)$. If k is of the same order as d this grows at a rate of $\mathcal{O}(4^d/\sqrt{d})$!

2 Practical Part (50 pts)

You should include all the python files you used to get your results. It should have a main file (which can be a notebook) that produces the required plots, one after another. Your results should be reproducible! Briefly explain how to run your code in the report.

A sample solution is provided in this Colab notebook.

1. Undergraduates 15 pts Graduates 10 pts
   Question. Implement in python the ridge regression with gradient descent. We will call this algorithm `regression_gradient`. Note that we now have parameters $\mathbf{w}$ and b we want to learn on the training set, as well an *hyper*-parameter to control the capacity of our model: λ. There are also hyper-parameters for the optimization: the step-size η, and potentially the number of steps.

2. Undergraduates 5 pts Graduates 5 pts
   Question. Consider the function $h(x) = \sin(x) + 0.3x - 1$. Draw a dataset D_n of pairs $(x, h(x))$ with $n = 15$ points where x is drawn uniformly at random in the interval $[-5, 5]$. Make sure to use the **same** set D_n for **all** the plots below.

3. Undergraduates 10 pts Graduates 5 pts
   Question. With $\lambda = 0$, train your model on D_n with the algorithm `regression_gradient`). Then plot on the interval $[-10, 10]$: the points from the training set D_n, the curve $h(x)$, and the curve of the function learned by your model using gradient descent. Make a clean legend. **Remark:** The solution you found with gradient descent should converge to the straight line that is closer from the n points (and also to the analytical solution). Be ready to adjust your step-size (small enough) and number of iterations (large enough) to reach this result.

4. Undergraduates 10 pts Graduates 5 pts
Question. on the same graph, add the predictions you get for intermediate value of λ, and for a large value of λ. Your plot should include the value of λ in the legend. It should illustrate qualitatively what happens when λ increases.

5. Undergraduates 10 pts Graduates 10 pts
Question. Draw another dataset D_{test} of 100 points by following the same procedure as D_n. Train your linear model on D_n for λ taking values in $[0.0001, 0.001, 0.01, 0.1, 1, 10, 100]$. For each value of λ, measure the average quadratic loss on D_{test}. Report these values on a graph with λ on the x-axis and the loss value on the y-axis.

6. Undergraduates bonus: 10 pts Graduates 10 pts
Question. Use the technique studied in problem 1.3 above to learn a non-linear function of x. Specifically, use Ridge regression with the fixed pre-processing ϕ_{poly^l} described above to get a polynomial regression of order l. Apply this technique with $\lambda = 0.01$ and different values of l. Plot a graph similar to question 2.2 with all the prediction functions you got. Don't plot too many functions to keep it readable and precise the value of l in the legend.

7. Undergraduates bonus: 5 pts Graduates 5 pts
Question. Comment on what happens when l increases. What happens to the empirical risk (loss on D_n), and to the true risk (loss on D_{test})?