1 Fun with least squares

In ordinary least squares we learn to predict a *target* scalar $y \in \mathbb{R}l$ given a *feature* vector $\vec{x} \in \mathbb{R}^d$. Each element of $\vec{x}$ is called a feature, which could correspond to a scientific *measurement*. For example, the i-th element of $\vec{x}$, denoted by $(\vec{x})_i$, could correspond to the velocity of a car at time i. y could represent the final location (say just in one direction) of the car.

For the purpose of predicting y from $\vec{x}$ we are given n samples $(\vec{x}_i, y_i)$ with $i = 1, \ldots, n$ (where feature vectors and target scalars are observed in pairs), which we also call the training set. In this problem we want to predict the unobserved target y corresponding to a new $\vec{x}$ (not in the training set) by some linear prediction $\hat{y} = \vec{x}^\top \widehat{\vec{w}}$ where the *weight* $\widehat{\vec{w}} \in \mathbb{R}^d$ minimizes the least-squares training cost

$$\sum_{i=1}^{n} (\vec{x}_i^\top \vec{w} - y_i)^2 = \|\mathbf{X}\vec{w} - \vec{y}\|_2^2$$

where in the matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, the transposed sample feature vectors $\vec{x}_i^\top$ constitute the d-dimensional row vectors, and the n-dimensional vectors of training measurements $\vec{x}^j = ((\vec{x}_1)_j, \ldots, (\vec{x}_n)_j)^\top$ for $j = 1, \ldots, d$ correspond to the column vectors and $\vec{y} = (y_1, \ldots, y_n)^\top$.

Let us actually build on the example mentioned above and view the measurements $(\vec{x}_i)j$ of each sample $\vec{x}_i$ as a sequence of measurements, e.g. velocities of car i, over time $j = 1, \ldots, d$.

(a) Is this problem in a supervised or unsupervised learning setting? **Please explain.**

(b) Suppose that we want to learn (from our training set) to predict the location y from only the first t measurements. Denoting the prediction of y at time t by $\hat{y}^t$, we thus want to use $(\vec{x})_j, j = 1, \ldots, t$ to predict y. If we now learn how to obtain $\hat{y}^t$ for each $t = 1, \ldots, d$, we end up with a sequence of estimators $\hat{y}^1, \ldots, \hat{y}^d$ for each car.

Provide a method to obtain $\hat{y}^t$ for each t. Note that we will obtain a different model for each t.

(c) Someone suggests that maybe the measurements themselves are partially predictable from the previous measurements, which suggests employing a two stage strategy to solve the original prediction problem: First we predict the t-th measurement $(\vec{x})_t$ based on the previous measurements $(\mathbf{x})_1, \ldots, (\mathbf{x})_{t-1}$. Then we look at the differences (sometimes deemed the "innovation") between the actual t-th measurement we obtained and our prediction for it, i.e. $(\Delta\vec{x})_t := (\vec{x})_t - (\widehat{\vec{x}})_t$. Finally, we use $(\Delta\vec{x})_1, \ldots, (\Delta\vec{x})_t$ to obtain a prediction $\tilde{y}^t$.

In order to learn the maps which allow us to (1) take $(\mathbf{x})_1, \ldots, (\mathbf{x})_{t-1}$ to obtain $(\Delta\vec{x})_1, \ldots, (\Delta\vec{x})_t$ and (2) take $(\Delta\vec{x})_1, \ldots, (\Delta\vec{x})_t$ to predict $\tilde{y}^t$, we again use our training set. Specifically for each t, in stage (1), we fit the vectors of training measurements $\vec{x}^1, \ldots, \vec{x}^{t-1}$ linearly to $\vec{x}^t$ using least squares for each t. In stage (2), we use the innovation vectors $(\Delta\vec{x}^1, \ldots, \Delta\vec{x}^t)$ to predict $\vec{y}^t$ again using least squares. Let's define the matrix $\tilde{\mathbf{X}}^t := (\Delta\vec{x}^1, \ldots, \Delta\vec{x}^t)$ and $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}^d$.

Show how we can learn the best linear predictions $\widehat{\vec{x}}^t$ from $\vec{x}^1, \ldots, \vec{x}^{t-1}$. Then **provide an expression** for $\tilde{\vec{y}}^t$ depending on the innovations $\Delta\vec{x}^1, \ldots, \Delta\vec{x}^t$.

When presented with a new feature vector $\vec{x}$, are the sequence of final predictions of the one-stage training $\hat{y}^t$ in (a) and two-stage training $\tilde{y}^t$ in (b) the same? **Explain your reasoning.**

(d) **Which well-known procedure do the steps to obtain $\tilde{\mathbf{X}}$ from $\mathbf{X}$ remind you of**? (**HINT:** Think about how the column vectors in $\tilde{\mathbf{X}}$ are geometrically related.)

Is there an efficient way to update the weight vector $\widehat{\vec{w}}^t$ from $\widehat{\vec{w}}^{t-1}$ when computing the sequence of predictions $\tilde{y}^t$?

(e) Now let's consider the more general setting where we now want to predict a target vector $\vec{y} \in \mathbb{R}^k$ from a feature vector $\vec{x} \in \mathbb{R}^d$, thus having a training set consisting of observations $(\vec{x}_i, \vec{y}_i)$ for $i = 1, \ldots, n$.

Instead of learning a weight vector $\vec{w} \in \mathbb{R}^d$, we now want a linear estimate $\widehat{\vec{y}} = \hat{\mathbf{W}}\vec{x}$ with a weight matrix $\hat{\mathbf{W}} \in \mathbb{R}^{k \times d}$ instead. From our samples, we obtain wide matrices $\mathbf{Y} \in \mathbb{R}^{k \times n}$ with columns $\vec{y}_1, \ldots, \vec{y}_n$ and $\mathbf{X} \in \mathbb{R}^{d \times n}$ with columns $\vec{x}_1, \ldots, \vec{x}_n$. In order to learn $\hat{\mathbf{W}}$ we now want to minimize $\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2$ where $\|\cdot\|_F$ denotes the Frobenius norm of matrices, i.e. $\|\mathbf{L}\|_F^2 = \text{trace}(\mathbf{L}^\top \mathbf{L})$.

Show how to find $\hat{\mathbf{W}} = \arg\min_{\mathbf{W} \in \mathbb{R}^{d \times d}} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2$ using vector calculus.

(f) In the setting of problem (e), **argue why** the computation of the best linear prediction $\widehat{\vec{y}}$ of a target vector $\vec{y}$ using a feature vector $\vec{x}$ can be solved by separately finding the best linear prediction for each measurement $(\mathbf{y})_j$ of the target vector $\vec{y}$.