1 Supervised learning and linear regression background

- We define $X \in \mathbb{R}^{N\times(d+1)}$ as a matrix whose rows are the data vectors: $(1, x_1^i, x_1^i, \ldots, x_d^i)$ for $i = 1, 2, \ldots, N$. With this notation, an affine function $f : \mathbb{R}^{d+1} \to \mathbb{R}$ acting on each data point gives rise to $\hat{y}^i = w^T \bar{x}^i$, $i = 1, \ldots, N$. If stack the $\hat{y}^i \in \mathbb{R}$ together and write it as a vector $\hat{y} = (\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^N)$, we can then write $\hat{y} = Xw$. Thus, if the true labels are $y \in \mathbb{R}^N$, verify that we can then write the error vector in prediction using a linear model as a vector $e = (e^1, e^2, \ldots, e^N)$ with $e^i = \hat{y}^i - y^i$ as $e = Xw - y$.
- In case each label y is a vector in $\mathbb{R}^m$, we would have a set of m weights for each element of y, since we learn an affine functions for each element of y. We use the notation $w_{k,j}$ to refer to the j-th element of the k-th weight vector for $j = 0, 1, \ldots, d$ and $k = 1, 2, \ldots, m$. It follows that $y_l^i = w_l^T x^i$, for $l \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, N\}$.

Our goal in supervised learning is to learn a function $f : \mathbb{R}^d \to \mathbb{R}^m$, called a predictor, such that for a given independent variable x we can predict the corresponding dependent variable y as $\hat{y} := f(x) \approx y$. We often fix a model class (e.g. linear) and parameterize f with a set of parameters $w \in \mathbb{R}^p$ (e.g. coefficients in the linear regression).

1.1 Loss or cost function

Given a pair (x, y), the error in our prediction depends on how close our prediction $\hat{y}$ is to the true label y. Since $\hat{y} = f(x)$ and f is parameterized by w, the error becomes a function of w. Based on the error, we define a loss function $L : \mathbb{R}^p \to \mathbb{R}$. We then aim to learn w so as to minimize the loss $L(w)$. Note: the loss function is also referred to as a cost function, and sometimes also as the performance metric (though in some cases the performance metric might be different than the loss function). Furthermore, this function is sometimes written as $J(.)$ instead of $L(.)$.

Gradient, hessian, partial derivative

- The gradient of a function $L : \mathbb{R}^p \to \mathbb{R}$ is denoted by $\nabla L(w) \in \mathbb{R}^p$. It is formed by stacking together the partial derivatives $\frac{\partial L}{\partial w_i}$, $i = 1, 2, \ldots, p$ in a vector.
- For an affine function $f : \mathbb{R}^p \to \mathbb{R}$, represented as $f(w) = a^T w + b$, with $a \in \mathbb{R}^p, b \in \mathbb{R}$, verify that the gradient is $\nabla f(w) = a$. The gradient of several loss functions we have encountered can be derived using this fact, the chain rule and product rule from calculus.
- The problem of finding minimum of a function $L : \mathbb{R}^p \to \mathbb{R}$ is written as $\min_w L(w)$ and the optimizer of the problem is denoted by $w^* = \arg\min L(w)$.
- Recall that if $w^* \in \mathbb{R}^p$ is a minimum of a differentiable function $L : \mathbb{R}^p \to \mathbb{R}$, then $\nabla L(w^*) = 0$. In other words, for an unconstrained optimization, the gradient of the function has to vanish at an optimum.

Exercise 1. Go through the notes on computing gradient and Hessian of linear and quadratic functions here.

- Derive the gradient and Hessian of the function $f : \mathbb{R}^p \to \mathbb{R}$, given by $f(w) = w^T a + b$ with respect to $w \in \mathbb{R}^p, b \in \mathbb{R}$.
- Derive the gradient and Hessian of the function $f : \mathbb{R}^p \to \mathbb{R}$, given by $f(w) = w^T Aw + w^T a + b$, with respect to $w \in \mathbb{R}^p, b \in \mathbb{R}$.

Exercise 2. Consider $M \in \mathbb{R}^{p \times p}$.

- Under which conditions is M invertible? you may state as many conditions as you know.
- Now, suppose $M = X^T X$. Under which conditions on $X \in \mathbb{R}^{N \times p}$ is M invertible? What would this imply regarding your data matrix X?