Lecture notes (linear regression)

1 Linear Regression

Given training data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \mathbb{R}$, linear regression learns a model vector $\boldsymbol{w} \in \mathbb{R}^D$ by minimizing the square error defined on training data:

$$\begin{aligned} \boldsymbol{w}^* &= \operatorname*{argmin}_{\boldsymbol{w}} \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{w}^T\boldsymbol{x}_n - y_n)^2 && (1) \\ &= \operatorname*{argmin}_{\boldsymbol{w}} \underbrace{\frac{1}{2}\|X\boldsymbol{w} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{w})}, && (2) \end{aligned}$$

where $X \in \mathbb{R}^{N\times D}$ is the data matrix; each row of X corresponds to the feature vector of a training sample; $\boldsymbol{y} \in \mathbb{R}^N$ is the label vector with $\boldsymbol{y} = [y_1, \ldots, y_N]$. In the prediction phase, given a test sample $\tilde{\boldsymbol{x}}$, the model computes the prediction by $\boldsymbol{w}^T\tilde{\boldsymbol{x}}$.

Next we discuss how to to solve the minimization problem in (2):

1. The gradient of (2) can be written as

$$\nabla f(\boldsymbol{w}) = X^T X\boldsymbol{w} - X^T\boldsymbol{y}.$$

   Note that $\nabla f(\boldsymbol{w})$ is a D-dimensional vector, each element $(\nabla f(\boldsymbol{w}))_i = \frac{\partial f(\boldsymbol{w})}{\partial w_i}$.

2. The function $f(\boldsymbol{w})$ is a convex function. To see this, you can first assume $D = 1, N = 1$, then the function becomes single-variate $f(w) = (xw - y)^2$ which is clearly convex. In general, we can verify the convexity of a function from its second order derivative. In linear regression case

$$\nabla^2 f(\boldsymbol{w}) = X^T X.$$

   Here $X^T X$ is a semi-positive definite matrix (all the eigenvalues are non-negative), which implies $f(\boldsymbol{w})$ is convex. We will give more details in the next lecture when we talk about optimization.

For a convex and continuously differentiable convex function, we know

$$\boldsymbol{w}^* \text{ is a global minimum of } f(\boldsymbol{w}) \text{ if and only if } \nabla f(\boldsymbol{w}^*) = 0.$$

So, to solve (2), it's equivalent to finding a $\boldsymbol{w}^*$ such that $\nabla f(\boldsymbol{w}^*) = 0$. This means $\boldsymbol{w}^*$ is minimum if and only if it satisfies the following equation:

$$\textbf{normal equation: } \; X^T X \boldsymbol{w}^* = X^T \boldsymbol{y}. \tag{3}$$

This is called "normal equation" for linear regression. To solve (3), we consider the following two cases:

When X^TX is invertible, eq (3) directly implies $\boldsymbol{w}^* = (X^TX)^{-1}X^T\boldsymbol{y}$ is the **unique** solution of linear regression. This often happens when we face an over-determined system—number of samples is much larger than number of variables ($N \gg D$). An intuitive way to see this: When $N \gg D$, we have many training samples to fit but don't have enough degree of freedom (number of variables), so it's unlikely to fit data very well and the minimizer can be uniquely determined.

When X^TX is not invertible, equation (3) does not have a unique solution. In fact, (3) will have **infinite** number of solutions in this case, and so does our linear regression problem—there will be infiinite number of solutions $\boldsymbol{w}^*$ that achieves the same minimal square error on the training data.

X^TX will not be invertible when $N < D$. To illustrate why we have infinite number of solutions, consider in a two-dimensional problem ($D = 2$) we have only one training sample $\boldsymbol{x}_1 = [1, -1], y_1 = 1$. We can see $\boldsymbol{w} = [a+1, a]$ for any $a \in \mathbb{R}$ will get 0 training error:

$$\boldsymbol{w}^T\boldsymbol{x}_1 = a + 1 - a = 1 = y_1.$$

This is true for any problem with $N < D$—in this case, you can always find a vector in the null space of X (a vector such that $X\boldsymbol{v} = 0$), and then for a solution $\boldsymbol{w}^*$, any vector with $\boldsymbol{w}^* + a\boldsymbol{v}$ with $a \in \mathbb{R}$ will get the same square error with $\boldsymbol{w}^*$. This case ($N < D$) is also called the **under-determined** problem, since you have too many degree of freedom in your problem and don't have enough constraints (data).

So how to find a solution in the under-determined case? Indeed, we could use the following approach to find the **minimum-norm solution $\boldsymbol{w}^+$**: Let $\mathcal{W} = \operatorname{argmin}_{\boldsymbol{w}} \|X\boldsymbol{w} - \boldsymbol{y}\|^2$ denote the set of solutions, we aim to find the minimum norm solution that

$$\boldsymbol{w}^+ = \operatorname*{argmin}_{\boldsymbol{w} \in \mathcal{W}} \|\boldsymbol{w}\|_2. \tag{4}$$

Theorem 1 (Singular Value Decomposition (SVD))**.** *Given an m-by-n real matrix A, it can be decomposed as*

$$A = U\Sigma V^T,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are unitiry ($U^TU = UU^T = I$, $V^TV = VV^T = I$) and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix with non-negative real numbers in the diagonal. We denote $\Sigma = diag[\sigma_1, \ldots, \sigma_r, 0, \ldots, 0]$, each $\sigma_i > 0$ and r is the rank of A.

Definition 1 (Pseudo Inverse)**.** *If $A = U\Sigma V^T$ is the SVD of A, the pseudo inverse of A can be defined by $A^+ = V\Sigma^+U^T$. Note that A^+ is an n-by-m matrix, and Σ^+ is an n-by-m diagonal matrix with $\Sigma^+ = diag[\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}, 0, \dots, 0]$.*

If A is inevitable, we must have $m = n = r$, and in this case we can easily verify that pseudo inverse will be the same with the normal inverse matrix:

$$A^{-1} = (V^T)^{-1}\Sigma^{-1}U^{-1} = V\Sigma^+U^T = A^+.$$

We get this equation because $V^TV = I$ (V is the inverse of V^T) and $U^TU = I$ (U^T is the inverse of U). If A is not invertible, the definition of pseudo inverse can be understood by: we inverse all the positive singular values, and keep the zero singular values remain zero.

You can also verify that A^+A is an n-by-n diagonal matrix with the first r diagonal entries to be 1, others to be 0; AA^+ will be a similar diagonal matrix with size m-by-m. So this is a more general notion of matrix inversion.

Now let's show the closed form solution of the minimum norm solution of linear regression (4) can be obtained by pseudo inverse:

Theorem 2. *The minimum norm solution of $\|X\boldsymbol{w} - \boldsymbol{y}\|_2^2$ is given by*

$$\boldsymbol{w}^+ = X^+\boldsymbol{y}.$$

Therefore, if $X = U\Sigma V^T$ is the SVD of X, then $\boldsymbol{w}^+ = V\Sigma^+U^T\boldsymbol{y}$.

Proof. First, we rewrite the linear regression objective:

$$\|X\boldsymbol{w} - \boldsymbol{y}\|_2 = \|U\Sigma V^T\boldsymbol{w} - \boldsymbol{y}\|_2 = \|\Sigma V^T\boldsymbol{w} - U^T\boldsymbol{y}\|_2.$$

Note that the second equation comes from the fact that U is unitary ($\|U\boldsymbol{v}\| = \|\boldsymbol{v}\|$ for any $\boldsymbol{v}$ if $U^TU = UU^T = I$).

Letting $\boldsymbol{z} = V^T\boldsymbol{w}$, we have $\|\boldsymbol{z}\| = \|\boldsymbol{w}\|$ (since V is unitary). Therefore, the least norm solution of $\|\Sigma V^T\boldsymbol{w} - U^T\boldsymbol{y}\|$ is equivalent to finding the least norm solution of

$$\min_{\boldsymbol{z}} \|\Sigma\boldsymbol{z} - U^T\boldsymbol{y}\|^2.$$

For this new problem, since Σ is diagonal, it's obvious that the minimum-norm solution is

$$\boldsymbol{z}^+ = \Sigma^+U^T\boldsymbol{y}.$$

Therefore the minimum norm solution of the original system is

$$\boldsymbol{w}^+ = V\boldsymbol{z}^+ = V\Sigma^+U^T\boldsymbol{y}.$$

□

1.1 Computational time

To compute the closed form solution of linear regression, we can:

1. Compute $X^T X$, which costs $O(nd^2)$ time and d^2 memory.
2. Inverse $X^T X$, which costs $O(d^3)$ time.
3. Compute $X^T \boldsymbol{y}$, which costs $O(nd)$ time.
4. Compute $\{(X^T X)^{-1}\}\{X^T \boldsymbol{y}\}$, which costs $O(nd)$ time.

So the total time in this case is $O(nd^2 + d^3)$. In practice, one can replace these steps by Gaussian elimination, which can reduce the time to $O(nd^2)$.

1. When d is small, nd^2 is not too expensive, so a closed form solution can be easily computed for linear regression.
2. When d is large, nd^2 is usually too large and we need to use other iterative algorithms to solve linear regression (next lecture).