LECTURE 3 - REVIEW OF LINEAR ALGEBRA & THE CURSE OF DIMENSIONALITY

1. Review the Polynomial Curve Fitting Example

- Suppose we are given a training set comprising of N observations of x, $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$, along with desired outputs $\mathbf{t} = (t_1, t_2, \ldots, t_N)^T$
- Now we must assume a model. Lets assume a polynomial function as our model:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1}$$

- Now we must *train* this model by estimating the unknown parameters ($\mathbf{w}$) that maps the training data, $\mathbf{x}$, to their desired values, $\mathbf{t}$, given some assumed value for M
- So, we have N discrete points from which to estimate $\mathbf{w}$. We can minimize the squared error to estimate the parameters:

$$\arg\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \left(y(x_n, \mathbf{w}) - t_n\right)^2 = \frac{1}{2}\sum_{n=1}^{N} \left(\sum_{j=0}^{M} w_j x_n^j - t_n\right)^2 \tag{2}$$

- Consider the following illustration of the error function:
- We can write the error function compactly in matrix/vector form:

$$E(\mathbf{w}) = \frac{1}{2}\left([w_0, w_1, \ldots, w_M]\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^M & x_2^M & \ldots & x_n^M \end{bmatrix} - [t_1, t_2, \ldots, t_N]\right)$$

$$\left([w_0, w_1, \ldots, w_M]\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^M & x_2^M & \ldots & x_n^M \end{bmatrix} - [t_1, t_2, \ldots, t_N]\right)^T \tag{3}$$

$$= \frac{1}{2}\left(\mathbf{w}^T\mathbf{X}^T - \mathbf{t}^T\right)\left(\mathbf{w}^T\mathbf{X}^T - \mathbf{t}^T\right)^T \tag{4}$$

$$= \frac{1}{2}\left\|\mathbf{w}^T\mathbf{X}^T - \mathbf{t}^T\right\|_2^2 \tag{5}$$

- To solve for $\mathbf{w}$, We can take the derivative of the error function, set it to zero, and solve for the parameters.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right)^2 \tag{6}$$

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \left[\frac{\partial E(\mathbf{w})}{\partial w_0}, \frac{\partial E(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial E(\mathbf{w})}{\partial w_M}\right]^T \tag{7}$$

$$= \left[\sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right) x_n^0, \sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right) x_n^1, \ldots, \right. \tag{8}$$
$$\left. \sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right) x_n^M\right]^T$$

$$= \left[\sum_{n=1}^{N}\left(\mathbf{w}^T\mathbf{x}_n^T - t_n\right) x_n^0, \sum_{n=1}^{N}\left(\mathbf{w}^T\mathbf{x}_n^T - t_n\right) x_n^1, \ldots, \sum_{n=1}^{N}\left(\mathbf{w}^T\mathbf{x}_n^T - t_n\right) x_n^M\right]^T \tag{9}$$

where $\mathbf{x}_n = \left[1, x_n^1, x_n^2, \ldots, x_n^M\right]$ (i.e., the n^{th} row of $\mathbf{X}$)

$$= \mathbf{X}^T\left(\mathbf{w}^T\mathbf{X}^T - \mathbf{t}^T\right)^T \tag{10}$$

where $\mathbf{X}^T = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ x_1^2 & x_2^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^M & x_2^M & \ldots & x_n^M \end{bmatrix}$.

- Then, we can set the derivative to zero and solve:

$$0 = \mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{t} \tag{11}$$

$$\mathbf{X}^t\mathbf{t} = \mathbf{X}^T\mathbf{X}\mathbf{w} \tag{12}$$

$$\mathbf{w} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{t} \tag{13}$$

- The $\mathbf{X}^T\mathbf{X}$ term is called the auto-correlation matrix (computes how correlated the data is with itself). The $\mathbf{X}^T\mathbf{t}$ is called the cross-correlation term (computes how correlated the data is with the desired output values).
- *Let us run the beer foam example in today's jupyter notebook and review the implementation of the above polynomial curve fitting example.*

2. Curse of Dimensionality

- *With what parameter settings is the polynomial curve fitting code likely to overfit? Why? What can you do to try to prevent overfitting?*

- *When we have a large model order and a small number of data points, what happens to the* $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ *term in our code?*
- Recall: Given a matrix $\mathbf{A}$, almost all vectors change direction when they are multiplied by $\mathbf{A}$. The only vectors that do not change direction when they are multiplied by $\mathbf{A}$ are the *eigenvectors* of $\mathbf{A}$. Each eigenvector is paired with an *eigenvalue*. When you multiply the eigenvector of $\mathbf{A}$ by $\mathbf{A}$, the result is a vector λ (the eigenvalue) times the original vector.

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{14}$$

- The *condition number* of a matrix can be computed as the ratio of its largest to smallest eigenvalues. A matrix with a small condition number (close to 1) is said to be *well conditioned*. A matrix with a very large condition number (or even infinite, which happens when an eigenvalue is 0) is said to be *ill conditioned*.
- The principal axes of the surfaces of equal error for $\mathbf{E}(w)$ correspond to the eigenvectors of the input correlation matrix ($\mathbf{X}^T\mathbf{X}$) and the rate of change of the gradient along the principal axes of the error surface contour correspond to the eigenvalues.
- Consider the following two system of equations and their solutions:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} 1.001 & 1 \\ 1 & 1.001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \tag{16}$$

- A small change in value, results in a large change in the parameter estimation. This is indicative of an *ill conditioned* matrix.
- Think about data with noise. Given an ill-conditioned system, small changes in value due to noise in the data may result in a very large change in parameter estimation. Another way to think about it is that ill-conditioned systems can magnify noise in a data set.
- The number of points needed to densely populate a space grows quickly with dimensionality. If you add informative, discriminating features - more features can be helpful - but you will need exponentially more points to fully understand the space.
- Things do not always behave as you would expect in high dimensions - this is part of the Curse of Dimensionality.
- A couple of illustrations to see this:
  - Radius needed to cover the same percentage volume with growing dimensionality:
    - Volume of unit line, square, cube, hyper-cube: $s^D = 1^D$
    - Side of a cube covering some percentage of the area: say, 10% would be $r^D = 1/10$, $r = (1/10)^{(1/D)}$
    - What happens as D increases?
  - Unit Porcupine: The unit hyper-sphere inscribed within the unit hyper-cube.

* Consider a sphere with radius r in D dimensions

$$S = \left\{ \mathbf{x} \middle| \sum_{i=1}^{D} x_i^2 \leq r^2 \right\}$$

It's volume is:

$$v_D(r) = \frac{r^D \pi^{\frac{D}{2}}}{\Gamma(\frac{d}{2}+1)}$$

where $\Gamma(n) = \int_0^\infty e^{-x} x^{n-1} dx$.
So, for $D = 1$: $v_1(r) = \frac{r\pi^{1/2}}{\Gamma(1/2+1)} = 2r$
$D = 2$: $v_2(r) = \frac{r^2\pi}{\Gamma(2)} = \pi r^2$
$D = 3$: $v_3(r) = \frac{r^2\pi^{3/2}}{\Gamma(3/2+1)} = \frac{4}{3}\pi r^3$

* Consider a hypercube with radius r. It's volume is $(2r)^D$.
So, for $D = 1$: $v_{1,c} = 2r$
$D = 2$: $v_{2,c} = 4r^2$
$D = 3$: $v_{3,c} = 8r^3$
* Take the case where the hyper-sphere is inscribed within the unit hypercube. What happens to the relative volume of the sphere and cube as D increases?

$$\begin{aligned} \frac{Vol(Sphere)}{Vol(Cube)} &= \frac{\frac{r^D \pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)}}{(2r)^D} \\ &= \frac{r^D \pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)(2r)^D} \\ &= \frac{\pi^{\frac{D}{2}}}{2^D \Gamma(\frac{D}{2}+1)} \end{aligned}$$

Note: The r dropped out, relative volume depends only on dimension.
– Volume of space between two spheres with slightly different radii in high dimensions

* $Vol_{crust} = Vol_{S_1} - Vol_{S_2}$ where radius of S_1 is greater than the radius of S_2

$$\begin{aligned} Vol_{crust} &= Vol_{S_1} - Vol_{S_2} \\ &= \left[1 - \frac{Vol_{S_2}}{Vol_{S_1}}\right] Vol_{S_1} \\ &= \left[1 - \frac{\frac{(a-\epsilon)^D \pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)}}{\frac{a^D \pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2}+1)}}\right] Vol_{S_1} \\ &= \left[1 - \frac{a^D(1-\frac{\epsilon}{a})^D}{a^D}\right] Vol_{S_1} \\ &= \left[1 - \left(1 - \frac{\epsilon}{a}\right)^D\right] Vol_{S_1} \end{aligned}$$

* What happens as D increases?