LECTURE 5 - THE CURSE OF DIMENSIONALITY & PRINCIPAL COMPONENTS ANALYSIS

1. 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 (M) and a small number of data points (N), what happens to the* $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ *term in our code?*
  - When $M > N$, the autocorrelation matrix will have linearly dependent columns meaning it is not full rank and is not invertible. If it is not invertible, then you cannot compute $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{d}$.
  - One method to try to address this issue is to *diagonally load* (or *regularize*) the autocorrelation matrix:

$$\mathbf{X}^T\mathbf{X} + \mathbf{I}\epsilon \tag{1}$$

  - You can also just use more data.
- 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 more and 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:
  1. **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?
  2. **Unit Porcupine**: The unit hyper-sphere inscribed within the unit hyper-cube.
     - Consider a sphere with radius r in D dimensions

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

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.

(3) **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?

2. Principal Components Analysis

- So, a major take away from the Curse of Dimensionality discussion is that when we are in high dimensional spaces, much of the space is empty and the data lives at the surface. Given this, it makes sense to use a lower-dimensional *manifold* representation of the data.
- A very common approach (and on of the simplest approaches) to dimensionality reduction is Principal Components Analysis (PCA). PCA takes data from sensor coordinates to data centric coordinates using linear projections (i.e., it is assuming that the informative components of the data lies on a linear manifold.)
- PCA uses a linear transformation to minimize the redundancy of the resulting transformed data (by ending up with data that is uncorrelated).
- PCA finds the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one.
- Without loss of generality, let's assume the input data has zero mean.

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{2}$$

The correlation matrix of $\mathbf{y}$ is:

$$\begin{aligned} R_y &= E[\mathbf{y}\mathbf{y}^T] \tag{3}\\ &= E[\mathbf{A}\mathbf{x}\mathbf{x}^T\mathbf{A}^T] \tag{4}\\ &= \mathbf{A}R_x\mathbf{A}^T \tag{5}\end{aligned}$$

If we are given N data vectors, $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, we can estimate R_x as

$$R_x \approx \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k\mathbf{x}_k^T \tag{6}$$

- This is a symmetric matrix, so, it's eigenvectors are mutually orthogonal.
- So, if we choose $\mathbf{A}$ to have columns equal to the orthonomal eigenvectors of R_x, the R_y is diagonal.

$$\mathbf{R}_y = \begin{bmatrix} \mathbf{e_1}^T \\ \mathbf{e_2}^T \\ \vdots \\ \mathbf{e_D}^T \end{bmatrix} \mathbf{R}_x\left[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_D\right] = \begin{bmatrix} \mathbf{e}_1^T\mathbf{R}_x\mathbf{e}_1 & \mathbf{e}_1^T\mathbf{R}_x\mathbf{e}_2 & \ldots & \mathbf{e}_1^T\mathbf{R}_x\mathbf{e}_D \\ \mathbf{e}_2^T\mathbf{R}_x\mathbf{e}_1 & \mathbf{e}_2^T\mathbf{R}_x\mathbf{e}_2 & \ldots & \mathbf{e}_2^T\mathbf{R}_x\mathbf{e}_D \\ \vdots & & \ddots & \vdots \\ \mathbf{e}_D^T\mathbf{R}_x\mathbf{e}_1 & \mathbf{e}_D^T\mathbf{R}_x\mathbf{e}_D & \ldots & \mathbf{e}_D^T\mathbf{R}_x\mathbf{e}_D \end{bmatrix} \tag{7}$$

$$= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \lambda_D \end{bmatrix} \tag{8}$$

where $\mathbf{e}_i \in \mathbb{R}^{D\times 1}$ and $\mathbf{R}_x \in \mathbb{R}^{D\times D}$.

  - Note: Given that $\mathbf{e}_i$ is an eigenvector of $\mathbf{R}_x$, we know that $\lambda_i \mathbf{e}_i = \mathbf{R}_x \mathbf{e}_i$. So, $\mathbf{e}_i^T \mathbf{R}_x \mathbf{e}_i = \mathbf{e}_i^T (\lambda_i \mathbf{e}_i) = \lambda_i$ using the fact that $\mathbf{e}_i$ is normalized (i.e., $\|\mathbf{e}_i\|_2^2 = 1$)
  - Similarly, Given that $\mathbf{e}_i$ and $\mathbf{e}_j$ are orthogonal eigenvectors of $\mathbf{R}_x$, we know that $\lambda_i \mathbf{e}_i = \mathbf{R}_x \mathbf{e}_i$. So, $\mathbf{e}_j^T \mathbf{R}_x \mathbf{e}_i = \mathbf{e}_j^T (\lambda_i \mathbf{e}_i) = 0$

2.1. **Given a symmetric matrix, the eigenvectors of distinct eigenvalues are orthogonal:** Let $\mathbf{A}$ be an $l \times l$ symmetric matrix, $\mathbf{A}^T = \mathbf{A}$. Then the eigenvectors corresponding to distinct eigenvalues are orthogonal. Let $\lambda_i \neq \lambda_j$ be two such eigenvalues. The the definitions we have:

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{9}$$

$$\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j \tag{10}$$

By multiplying the first eigenvalue equation on the left by $\mathbf{v}_j^T$ and by the transpose of the second equation on the right by $\mathbf{v}_i$, we get:

$$\mathbf{v}_j^T \mathbf{A}\mathbf{v}_i = \mathbf{v}_j^T \lambda_i \mathbf{v}_i \tag{11}$$

$$(\mathbf{A}\mathbf{v}_j)^T \mathbf{v}_i = (\lambda_j \mathbf{v}_j)^T \mathbf{v}_i \tag{12}$$

$$\mathbf{v}_j^T \mathbf{A}\mathbf{v}_i = \lambda_j \mathbf{v}_j^T \mathbf{v}_i \tag{13}$$

$$\mathbf{v}_j^T \mathbf{A}\mathbf{v}_i - \mathbf{v}_j^T \mathbf{A}\mathbf{v}_i = 0 = (\lambda_i - \lambda_j)\, \mathbf{v}_j^T \mathbf{v}_i \tag{14}$$

Thus, $\mathbf{v}_j^T \mathbf{v}_i = 0$.

- The eigenvecotrs can be interpreted as an orthogonal axis defined by the data. It makes sense to use these to look for data-centric coordinates/subspaces if the data projects differently to each axis. Since all data is noisy, we can concentrate on the axes corresponding to the largest eigenvectors if you are interested in preserving the variance of the data. However, when using PCA within a classification problem, it is much more difficult because we are interested in discriminability (not necessarily variance).