Homework 0 - Machine Learning

Catherine Baker

January 2024

1 Preliminaries

Yes, I have read through the course website, noted important dates, and class policies.

2 Preliminaries

2.1 MATH 361 Mathematical Statistics I

Taken at Emory University in the Mathematics Department

2.2 MATH 221 Linear Algebra

Taken at Emory University in the Mathematics Department

2.3 MATH 346 Linear Optimization

Taken at Emory University in the Mathematics Department

2.4 None

No except for some machine learning topics in CS 325 Artificial Intelligence and some pattern recognition in CS 571 Natural Language Processing, both in the Computer Science Department at Emory University

3 Honor Code Acknowledgement

I hereby affirm by writing my name that I will abide by the honor code set forth in BMI534/CS534 - Catherine Baker

4 Linear Algebra

4.1 Find A such that $[v_1, v_2]A = [\hat{v}_1, \hat{v}_2]$ and $[v_1, v_2] = A^{-1}[\hat{v}_1, \hat{v}_2]$

See findingA.py

4.2 Inverse of a Skew Symmetric Matrix

If A is skew symmetric then $A^T = -A$. We want to prove that the inverse of a skew symmetric A is also skew symmetric, or that $(A^{-1})^T = -A^{-1}$.

First we recognize that $(A^{-1})^T = (A^T)^{-1}$. Since A is skew symmetric, we know that $(A^T)^{-1} = (-A)^{-1}$. From these knows, we can show the following:

$$(A^{-1})^T = (A^T)^{-1} = (-A)^{-1} = -A^{-1}$$

So, $(A^{-1})^T = -A^{-1}$, and the inverse of the skew-symmetric A is shown to also be skew symmetric.

4.3 Real Eigenvalues from a Real Symmetric Matrix

If A is a real, symmetric, nxn matrix then $A = A^T = A^*$, the conjugate transpose, and A must have n eigenvectors. These can be represented as:

$$Ax = \lambda x \tag{1}$$

where λ is the eigenvalue of eigenvector x. We can also represent this statement by taking the conjugate transpose of both sides

$$(Ax)^* = (\lambda x)^* \Rightarrow A^*x^* = \lambda^*x^*$$

Because A is symmetric and real, $A^* = A$, so:

$$A^*x^* = \lambda^*x^* \Rightarrow Ax^* = \lambda^*x^*$$

$$Ax^* = \lambda^*x^* \tag{2}$$

Now, because $(a + bi)(a - bi) = (a^2 + b^2)$, we can show that multiplying a vector by it's conjugate transpose produces a non-negative, real number:

$$\begin{bmatrix} (a_1 + b_1 i) \\ (a_2 + b_2 i) \\ ... \\ (a_n + b_n i) \end{bmatrix} \begin{bmatrix} (a_1 - b_1 i) & (a_2 - b_2 i) & ... & (a_n - b_n i) \end{bmatrix} = (a_1^2 + b_1^2) + (a_2^2 + b_2^2) + ... + (a_n^2 + b_n^2) \tag{3}$$

$(a^2 + b^2)$ will always be a non-negative, real number. Using this idea, let's multiply each of our equations 1 and 2, by the conjugate transpose of their eigenvector, x,:

$$xAx^* = x\lambda^* x^*$$

$$x^*Ax = x^*\lambda x$$

Now let's isolate λ, our eigenvalue, on both sides to show that it is a real number:

$$\frac{xAx^*}{xx^*} = \lambda^* \tag{4}$$

$$\frac{x^*Ax}{x^*x} = \lambda \tag{5}$$

From our work in equation 3 we can remember that a vector multiplied by its conjugate transpose results in a real, non-negative number. We also know that our matrix A has only real, non-negative entries. From these two knowns, we can determine with confidence that our λ values in equations 4 and 5 will be both real and non-negative even if the corresponding eigenvector has complex entries if A is a real, symmetric matrix.

5 Statistics and Probability

5.1 Maximizing Shannon Entropy

The Shannon Entropy is shown as a measure of the unpredictability of set of outcomes:

$$S(X) = -\sum_{i=1}^{n} p_i log(p_i) \tag{6}$$

This probability will be maximized when the uncertainty of the outcomes is maximized. This occurs when all probabilities $P = p_1, p_2, ..., p_n$ are equally likely or for all p_i in P, $1 \leq i \leq n$, $p_i = \frac{1}{n}$. With all outcomes equally likely there is no way to predict one outcome over another, maximizing entropy.

5.2 Proving the Maximizing of Shannon Entropy

If all outcomes are equally likely then $p_i = \frac{1}{n}$ for all i in $1 \leq i \leq n$. Plugging this into our summation equation we see:

$$S(X) = -\sum_{i=1}^{n} p_i log(p_i) \Rightarrow S(X) = -\sum_{i=1}^{n} \frac{1}{n} log(\frac{1}{n})$$

We can further simplify the right hand side of this equation by recognizing that both $\frac{1}{n}$ and $log(\frac{1}{n})$ are constants and can be pulled out of the summation leaving us with:

$$S(X) = -\frac{1}{n} log(\frac{1}{n}) \sum_{i=1}^{n} 1$$

$\sum_{i=1}^{n} 1 = n$ because i starts at 1. This turns our above equation into the following:

$$S(X) = -\frac{1}{n} log(\frac{1}{n})n \Rightarrow S(X) = -log(\frac{1}{n})$$

As the log() function approaches 0, it's y value increases at a faster rate towards $-\infty$. So, as our n approaches ∞, $log(\frac{1}{n})$ will approach $-\infty$ at an increasingly faster rate. And by the Shannon Entropy Equation (6), we see that this negative number ultimately gets multiplied by another negative, resulting in an entropy value that approaches ∞ faster than n as n approaches ∞. These findings are in agreement with our assumption in 5.1.

6 Covariance

6.1 Transforming Standard Normal to Normal Distribution

First let's identify our normally-distributed random variable:

$$X \sim \mathcal{N}(\mu, \Sigma), X \in \mathbb{R}^p$$

And our random variable distributed over the standard normal distribution:

$$Z \sim \mathcal{N}(0, I), Z \in \mathbb{R}^p$$

We want to show how we can transform samples from our variable Z's standard normal distribution to our variable X's normal distribution through use of the diagonalization of the covariance matrix of the normally-distributed X, $\Sigma = V\Lambda V^T$, where V is a matrix of the eigenvectors of Σ and Λ is the matrix of eigenvalues of Σ. To show this we must find a linear transformation for which both the covariance matrix and mean vector from Z can be coverted to that of X. This is possible because a linear transformation of a normally-distributed variable is also normally-distributed:

$$AZ + b \sim \mathcal{N}(A * 0 + b, AA^T I) \Rightarrow AZ + b \sim \mathcal{N}(b, AA^T)$$

In our problem, we need $b = \mu$ and $AA^T = \Sigma$. After diagonalizing Σ we see that we must find a matrix A such that:

$$AA^T = V\Lambda V^T$$

This can be achieved relatively easily by remembering the orthogonality of V and the diagonality of Λ. If we set $A = V\Lambda^{1/2}$, then $AA^T = V\Lambda^{1/2}V^T(\Lambda^{1/2})^T$. Because Λ is diagonal, $\Lambda = \Lambda^T$. And congruently, $\Lambda^{1/2} = (\Lambda^{1/2})^T$ because it is diagonal as well. So we can simplify our equation to the following:

$$AA^T = V\Lambda^{1/2}V^T(\Lambda^{1/2})^T$$

$$\Rightarrow AA^T = V\Lambda^{1/2}(\Lambda^{1/2})V^T$$

$$\Rightarrow AA^T = V\Lambda V^T = \Sigma$$

Through the above calculations we have shown that by setting $b = \mu$ and $A = V\Lambda^{1/2}$, we can perform the following linear transformation on Z to match the distribution of X:

$$X = V\Lambda^{1/2}Z + \mu$$

6.2 Plotting Normal Distribution with the PDF

See distribution.py

6.3 Plotting Normal Distribution with the Euclidian Distance

See distributionEuclid.py

6.4 Plotting Normal Distribution with the Mahalanobis Distance

See distributionMahala.py

7 Sample Covariance

From our code and its plot, we can see that the larger N gets, the smaller the frobenius norm of the error is. See errorPlot.py

8 Notes

I have included all code files as a .py file and a .ipynb file. I plotted my graphs using jupyter lab which generates .ipynb files. I converted these to .py files as submission dictates I should use python to code, but I kept the .ipynb files in, in case they were easier to view my plots on.