Homework #0

CSE 546: Machine Learning
Cassia Cai
October 5, 2022

Collaborators: William Kumler, Apoorva Kalaskar, Madeleine Grunde-McLaughlin, Andrey Risukhin, Abbas Idris, and Chloe Yang

Probability and Statistics

A1. *[2 points]* (From Murphy Exercise 2.4.) After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease, and that the test is 99% accurate (i.e., the probability of testing positive given that you have the disease is 0.99, as is the probability of testing negative given that you don't have the disease). The good news is that this is a rare disease, striking only one in 10,000 people. What are the chances that you actually have the disease?

If you text positive, the chances that you actually have the disease is approx. 0.98%.

From the problem statement, we have information about the test sensitivity and the prior probability of this rare disease, where $x = 1$ is the event that the test is positive, and $y = 1$ is the event that you have the disease.

$$p(x = 1|y = 1) = 0.99, p(x = 1|y = 0) = 0.01 \tag{1}$$

$$p(y = 1) = 0.0001, p(y = 0) = 0.9999 \tag{2}$$

Recall Bayes Theorem:

$$p(X = x|Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)} = \frac{p(X = x)p(Y = y|X = x}{\Sigma_{x'} p(X = x')p(Y = y|X = x')} \tag{3}$$

For our problem, we have:

$$p(y = 1|x = 1) = \frac{p(y = 1)p(x = 1|y = 1)}{p(y = 1)p(x = 1|y = 1) + p(y = 0)p(x = 1|y = 0)} \tag{4}$$

$$p(y = 1|x = 1) = \frac{0.0001 \times 0.99}{0.0001 \times 0.99 + 0.999 \times 0.01} = 0.0098 \tag{5}$$

If you text positive, you only have a 0.98% of actually having the disease!

A2. For any two random variables X, Y the *covariance* is defined as $\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$.

a. *[1 point]* If $\mathbb{E}[Y \mid X = x] = x$ show that $\text{Cov}(X, Y) = \mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big]$.

Starting from $\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$ and using linearity of expectation, we expand:

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \tag{6}$$

$$\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X\,\mathbb{E}[Y]] - \mathbb{E}[Y\,\mathbb{E}[X]] + \mathbb{E}[X]\,\mathbb{E}[Y] \tag{7}$$

$$\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y] \tag{8}$$

Let us now work with $\mathbb{E}[XY]$ and $\mathbb{E}[X]\,\mathbb{E}[Y]$:

By the law of total expectation:

$$\mathbb{E}[XY] = \sum_x \mathbb{P}(X=x)\,\mathbb{E}[XY \mid X=x] = \sum_x \mathbb{P}(X=x)x\,\mathbb{E}[Y \mid X=x] = \sum_x \mathbb{P}(X=x)x^2 = \mathbb{E}\big[X^2\big] \tag{9}$$

By the law of total expectation:

$$\mathbb{E}[Y] = \sum_x \mathbb{P}(X=x)\,\mathbb{E}[Y \mid X=x] = \sum_x \mathbb{P}(X=x)x = \mathbb{E}[X] \tag{10}$$

Now, we have $\mathbb{E}[XY]$ and $\mathbb{E}[X]\,\mathbb{E}[Y]$. Combining, we show that:

$$\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y] = \mathbb{E}\big[X^2\big] - (\mathbb{E}[X])^2 = \mathbb{E}\Big[(X-\mathbb{E}[X])^2\Big] \tag{11}$$

b. *[1 point]* If X, Y are independent show that $\mathrm{Cov}(X,Y) = 0$.

From above, $Cov(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]$. If X, Y are independent:

$$\mathbb{E}[XY] = \sum_{x,y} \mathbb{E}[XY \mid X=x, Y=y]\mathbb{P}(X=x, Y=y) = \sum_{x,y} xy\mathbb{P}(X=x, Y=y) \tag{12}$$

$$\mathbb{E}[XY] = \sum_x x\mathbb{P}(X=x) \sum_y y\mathbb{P}(Y=y) = \mathbb{E}[X]\,\mathbb{E}[Y] \tag{13}$$

So $Cov(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y] = 0$

A3. Let X and Y be independent random variables with PDFs given by f and g, respectively. Let h be the PDF of the random variable $Z = X + Y$.

a. *[1 point]* Show that $h(z) = \int_{-\infty}^{\infty} f(x)g(z-x)\,\mathrm{d}x$. (If you are more comfortable with discrete probabilities, you can instead derive an analogous expression for the discrete case, and then you should give a one sentence explanation as to why your expression is analogous to the continuous case.).

$$H(Z) = \mathbb{P}(Z \le z) = \mathbb{P}(X + Y \le z) \tag{14}$$

The joint PDF of X,Y is $f_{X,Y} = f(x)g(y)$.

$$H(Z) = \int\int_{x+y\le z} f_{X,Y}(x,y)dxdy = \int\int_{x+y\le z} f(x)g(y)dxdy = \int_{-\infty}^{\infty} f(x)\left(\int_{-\infty}^{z-x} g(y)dy\right)dx \tag{15}$$

$$H(Z) = \int_{-\infty}^{\infty} G(z-x)f(x)dx \tag{16}$$

$$h(z) = \frac{d}{dx}H(Z) = \frac{d}{dx}\int_{-\infty}^{\infty} G(z-x)f(x)dx = \int_{-\infty}^{\infty} f(x)g(z-x)dx \tag{17}$$

b. *[1 point]* If X and Y are both independent and uniformly distributed on $[0,1]$ (i.e. $f(x) = g(x) = 1$ for $x \in [0,1]$ and 0 otherwise) what is h, the PDF of $Z = X + Y$?

$$h(z) = \int_{-\infty}^{\infty} f(x)g(z-x)dx = \int_0^1 g(z-x)dx \tag{18}$$

We have $0 \le z - x \le 1$ and $z - 1 \le x \le z$. For $0 \le z \le 1$: $h(z) = \int_0^z dx = z$. For $1 < z \le 2$: $h(z) = \int_{z-1}^1 dx = 2 - z$. If not, $h(z) = 0$.

$$h(z) = \begin{cases} z & 0 \le z \le 1 \\ 2-z & 1 < z \le 2 \\ 0 & otherwise \end{cases}$$

A4. Let $X_1, X_2, ..., X_n \sim \mathcal{N}(\mu, \sigma^2)$ be i.i.d random variables. Compute the following:

a. *[1 point]* $a \in \mathbb{R}, b \in \mathbb{R}$ such that $aX_1 + b \sim \mathcal{N}(0, 1)$.

We have RV $X \sim \mathcal{N}(\mu, \sigma^2)$ which is Gaussian distributed with mean μ and variance σ^2. $Y = aX + b$ is also Gaussian where $Y \sim \mathcal{N}(0, 1)$

$$\mathbb{E}[Y] = a\,\mathbb{E}[X] + b = 0 \rightarrow a\mu + b = 0 \tag{19}$$

$$Var[Y] = a^2 Var[X] = 1 \rightarrow a^2\sigma^2 = 1 \tag{20}$$

We thus have

$$a = \frac{1}{\sigma}, b = \frac{-\mu}{\sigma}$$

b. *[1 point]* $\mathbb{E}[X_1 + 2X_2], \mathrm{Var}[X_1 + 2X_2]$.

$$\mathbb{E}[X_1 + 2X_2] = 3\mu$$

The expected value for X is μ so here, we have three of them. We can add them.

$$\mathrm{Var}[X_1 + 2X_2] = 5\sigma^2$$

This is from $\mathrm{Var}[X_1 + 2X_2] = \mathrm{Var}[X_1] + \mathrm{Var}[2X_2] = \sigma^2 + 4\sigma^2 = 5\sigma^2$

c. *[2 points]* Setting $\widehat{\mu}_n = \frac{1}{n}\sum_{i=1}^n X_i$, the mean and variance of $\sqrt{n}(\widehat{\mu}_n - \mu)$.

$$\mathbb{E}\big[\sqrt{n}(\widehat{\mu}_n - \mu)\big] = \sqrt{n}\,\mathbb{E}[(\widehat{\mu}_n] - \mathbb{E}[\mu)] = \sqrt{n}\,\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] - \mathbb{E}[\mu] = \sqrt{n}\,\mathbb{E}\left[\frac{1}{n}(n\mu)\right] - \mathbb{E}[\mu] = 0 \tag{21}$$

$$\mathrm{Var}\big[\sqrt{n}(\widehat{\mu}_n - \mu\big] = n\,\mathrm{Var}[\widehat{\mu}_n] + n\,\mathrm{Var}[\mu] = n\,\mathrm{Var}[\widehat{\mu}_n] = n\,\mathrm{Var}\left[\frac{1}{n}\sum_{i=1}^n X_i\right] = n\frac{\sigma^2}{n} = \sigma^2 \tag{22}$$

Linear Algebra and Vector Calculus

A5. Let $A = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 3 \\ 1 & 1 & 2 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}$. For each matrix A and B:

a. *[2 points]* What is its rank?
The rank of a matrix is the number of non-zero rows in the RREF of the matrix.

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 3 \\ 1 & 1 & 2 \end{bmatrix} \xrightarrow{(1)} \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 3 \\ 1 & 1 & 2 \end{bmatrix} \xrightarrow{(2)} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \xrightarrow{(3)} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{23}$$

We made matrix A into RREF by (1) row 1 - row 3 and swap row 2 and row 1, (2) row 3 - row 1, and (3) row 3 - row 2. The rank of matrix A, or $rank(A) = 2$.

$$B = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix} \xrightarrow{(1)} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 3 \end{bmatrix} \xrightarrow{(2)} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \xrightarrow{(3)} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 2 & 2 \end{bmatrix} \xrightarrow{(4)} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{24}$$

We made matrix B into RREF by (1) swapping the rows, (2) row 2 - row 1, (3) row 3 - row 1, and (4) row 3 - 2 times row 2. The rank of matrix B, or $rank(B) = 2$.

b. *[2 points]* What is a (minimal size) basis for its column span?

We identify and count the number of linearly independent columns. The basis for its column span can be found from the RREF matrix by finding the columns that do not pivot. Using this information, we find that the minimal size basis for A and B's column space:

$$\left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \right\}$$

A6. Let $A = \begin{bmatrix} 0 & 2 & 4 \\ 2 & 4 & 2 \\ 3 & 3 & 1 \end{bmatrix}$, $b = \begin{bmatrix} -2 & -2 & -4 \end{bmatrix}^\top$, and $c = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$.

a. *[1 point]* What is Ac?

$$Ac = \begin{bmatrix} 0 & 2 & 4 \\ 2 & 4 & 2 \\ 3 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0+2+4 \\ 2+4+2 \\ 3+3+1 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 7 \end{bmatrix} \tag{25}$$

b. *[2 points]* What is the solution to the linear system $Ax = b$?

$$\left[\begin{array}{ccc|c} 0 & 2 & 4 & -2 \\ 2 & 4 & 2 & -2 \\ 3 & 3 & 1 & -4 \end{array}\right] \xrightarrow{(1)} \left[\begin{array}{ccc|c} 0 & 1 & 2 & -1 \\ 1 & 2 & 1 & -1 \\ 3 & 3 & 1 & -4 \end{array}\right] \xrightarrow{(2)} \left[\begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & 1 & 2 & -1 \\ 3 & 3 & 1 & -4 \end{array}\right] \xrightarrow{(3)} \left[\begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & 1 & 2 & -1 \\ 0 & -3 & -2 & -1 \end{array}\right] \xrightarrow{(4)} \tag{26}$$

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & -1 \end{array}\right] \xrightarrow{(5)} \left[\begin{array}{ccc|c} 1 & 2 & 1 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right] \xrightarrow{(6)} \left[\begin{array}{ccc|c} 1 & 0 & 1 & -3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right] \xrightarrow{(7)} \left[\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right] \tag{27}$$

We thus find $x = \begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix}$

We found this by doing the following steps: (1) Dividing rows 1 and 2 by 2, (2) Swapping rows 1 and 2, (3) row 3 - 3 times row 1, (4) row 3 + 3 times row 2 and then divide row 3 by 2, (5) row 2 - 2 times row 3, (6) row 1 - 2 times row 2, and (7) row 1 - row 3.

A7. For possibly non-symmetric $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ and $c \in \mathbb{R}$, let $f(x, y) = x^\top \mathbf{A} x + y^\top \mathbf{B} x + c$. Define

$$\nabla_z f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial z_1}(x, y) & \frac{\partial f}{\partial z_2}(x, y) & \dots & \frac{\partial f}{\partial z_n}(x, y) \end{bmatrix}^\top \in \mathbb{R}^n \, .$$

a. *[2 points]* Explicitly write out the function $f(x, y)$ in terms of the components $A_{i,j}$ and $B_{i,j}$ using appropriate summations over the indices.

$$f(x, y) = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \begin{bmatrix} B_{11} & \dots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{n1} & \dots & B_{nn} \end{bmatrix} \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$$

We recall that the definition of matrix multiplication is $c_{ij} = \sum_{k=1}^{n} a_{ij} b_{kj}$. Expressing as summations over indices, we get:

$$f(x, y) = \sum_{i=1}^{n} x_i \sum_{j=1}^{n} A_{ij} x_j + \sum_{i=1}^{n} y_i \sum_{j=1}^{n} B_{ij} x_j + c$$

b. *[2 points]* What is $\nabla_x f(x, y)$ in terms of the summations over indices *and* vector notation?

Let us expand along the first element of x^T, the first row of **A** and x. We thus have

$$x_1 A_{11} x_1 + x_1 A_{12} x_2 + x_1 A_{13} x_3 + ... + x_1 A_{1n} x_n$$

$$\frac{\partial}{\partial x_1} = 2x_1 A_{11} + A_{12} x_2 + A_{13} x_3 + ... + A_{1n} x_n$$

$$\frac{\partial}{\partial x_k} = 2x_k A_{kk} + \sum_{i=1.j \neq k} A_{ki} x_i + \sum_{j=1.i \neq k} A_{jk} x_j$$

We also need to find the partial derivative of $\sum_{i=1}^n y_i \sum_{j=1}^n B_{ij} x_j$

We have:

$$\frac{\partial}{\partial x_k} = y_k B_{kk} + y_2 B_{2k} + ... + y_n B_{nk} = \sum_{i=1} y_i B_{ik}$$

Combining them, we get:

$$\frac{\partial}{\partial x_k} = 2x_k A_{kk} + \sum_{i=1.j \neq k} A_{ki} x_i + \sum_{j=1.i \neq k} A_{jk} x_j + \sum_{i=1} y_i B_{ik}$$

We can further simplify because $\sum_{j=1}^n A_{jk} x_j = x_k A_{kk} + \sum_{i=1,j \neq k}^n A_{jk} x_j$ and $\sum_{i=1}^n A_{ki} x_i = x_k A_{kk} + \sum_{j=1,i \neq k}^n A_{ki} x_i$

$$\frac{\partial}{\partial x_k} = \sum_{j=1}^n A_{jk} x_j + \sum_{i=1}^n A_{ki} x_i + \sum_{i=1}^n y_i B_{ik}$$

$$\frac{\partial}{\partial x_k} = \sum_{i=1}^n (A_{ik} x_i + A_{ki} x_i + y_i B_{ik})$$

$$\nabla_x f(x, y) = A^T x + Ax + B^T y$$

c. *[2 points]* What is $\nabla_y f(x, y)$ in terms of the summations over indices *and* vector notation?

If we do the matrix multiplication (expanding it out), we have $y_1 B_{11} x_1 + y_2 B_{21} x_1 + ... + y_n B_{n1} x_1$. We can see then that:

$$\frac{\partial}{\partial y_k} = \sum_{i=1}^n B_{ki} x_i$$

In vector form, we have:

$$\nabla_y f(x, y) = Bx$$

A8. Show the following:

a. *[2 points]* Let $g : \mathbb{R} \to \mathbb{R}$ and $v, w \in \mathbb{R}^n$ such that $g(v_i) = w_i$. Find an expression for g such that $\text{diag}(v)^{-1} = \text{diag}(w)$.

For example, we can start with square matrices **W** and **V**:

$$\mathbf{W} = \mathbf{D} = \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix}$$

$$\mathbf{V} = \mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{w_1} & 0 \\ 0 & \frac{1}{w_2} \end{bmatrix}$$

Then we can see that a general expression for g would be:

$$g(v_i) = \frac{1}{w_i}$$

b. *[2 points]* Let $\mathbf{A} \in R^{n\times n}$ be orthonormal and $x \in \mathbb{R}^n$. An orthonormal matrix is a square martix whose columns and rows are orthonormal vectors, such that $\mathbf{A}\mathbf{A}^\top = \mathbf{A}^\top\mathbf{A} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Show that $||\mathbf{A}x||_2^2 = ||x||_2^2$.

$$||\mathbf{A}x||_2^2 = (\mathbf{A}x)^T(\mathbf{A}x) = x^T\mathbf{A}^T\mathbf{A}x = x^T(\mathbf{I})x = x^Tx = ||x||_2^2$$

We have thus shown that $||\mathbf{A}x||_2^2 = ||x||_2^2$.

c. *[2 points]* Let $\mathbf{B} \in R^{n\times n}$ be invertible and symmetric. A symmetric matrix is a square matrix satisfying $\mathbf{B} = \mathbf{B}^\top$. Show that $\mathbf{B}^{-1}$ is also symmetric.

From the problem statement, we have that $\mathbf{B}^T = \mathbf{B}$. We already know that $\mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$.

$$(\mathbf{B}^{-1})^T = (\mathbf{B}^{-1})^T\mathbf{B}^{-1}\mathbf{B} = (\mathbf{B}^{-1})^T\mathbf{B}^T(\mathbf{B}^T)^{-1} = (\mathbf{B}\mathbf{B}^{-1})^T(\mathbf{B}^T)^{-1} = (\mathbf{B}^T)^{-1} = \mathbf{B}^{-1}$$

We have thus shown that $(\mathbf{B}^{-1})^T = \mathbf{B}^{-1}$ and this is the definition of symmetry.

d. *[2 points]* Let $\mathbf{C} \in R^{n\times n}$ be positive semi-definite (PSD). A positive semi-definite matrix is a symmetric matrix satisfying $x^\top\mathbf{C}x \geq 0$ for any vector $x \in \mathbb{R}^n$. Show that its eigenvalues are non-negative.

We start with the definition of the eigenvalue.

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

If we set $\mathbf{C} = \lambda$ and work with the inequality given in the question prompt, we have:

$$x^\top\lambda x \geq 0$$

$$x^\top x\lambda \geq 0$$

We also know that:

$$x^\top x = ||x||_2^2$$

And we know that $||x||_2^2$ is non-negative. Working from our inequality, we have $||x||_2^2$ multiplied by λ which is greater than or equal to 0. This means that λ or the eigenvalues are non-negative.

Programming

These problems are available in a .zip file, with some starter code. All coding questions in this class will have starter code. **Before attempting these problems, you will need to set up a Conda environment that you will use for every assignment in the course. Unzip the HW0-A.zip file and read the instructions in the README file to get started.**

A9. For $\nabla_x f(x, y)$ as solved for in Problem 7:

a. *[1 point]* Using native Python, implement the summation form.

```
def vanilla_solution(x: Vector, y: Vector, A: Matrix, B: Matrix) -> Vector:
    """Calculates gradient of f(x, y) with respect to x using vanilla python lists.
    Where $$f(x, y) = x^T A x + y^T B x + c$$

    Args:
        x (Vector): a (n,) vector.
        y (Vector): a (n,) vector.
        A (Matrix): a (n, n) matrix.
        B (Matrix): a (n, n) matrix.

    Returns:
        Vector: a resulting (n,) vector.
    """
    first_list = vanilla_matmul(vanilla_transpose(A),x)
    second_list = vanilla_matmul(A,x)
    third_list = vanilla_matmul(vanilla_transpose(B),y)

    temp_list = [sum(value) for value in zip(first_list, second_list)]
    final_list = [sum(value) for value in zip(temp_list, third_list)]
    return final_list
```

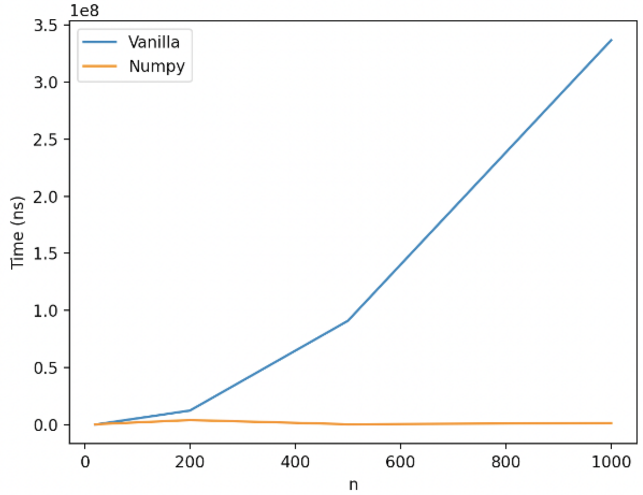
b. *[1 point]* Using NumPy, implement the vector form.

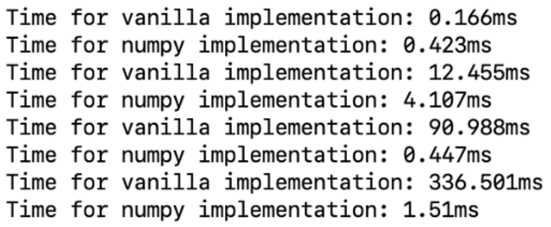
```
def numpy_solution(
    x: np.ndarray, y: np.ndarray, A: np.ndarray, B: np.ndarray
) -> np.ndarray:
    """Calculates gradient of f(x, y) with respect to x using numpy arrays.
    Where $$f(x, y) = x^T A x + y^T B x + c$$

    Args:
        x (np.ndarray): a (n,) numpy array.
        y (np.ndarray): a (n,) numpy array.
        A (np.ndarray): a (n, n) numpy array.
        B (np.ndarray): a (n, n) numpy array.

    Returns:
        np.ndarray: a resulting (n, ) numpy array.
    """
    return A.transpose() @ x + A @ x + B.transpose() @ y
```

c. *[1 point]* Report the difference in wall-clock time for parts a-b, and discuss reasons for the observed difference.





```
Time for vanilla implementation: 0.166ms
Time for numpy implementation: 0.423ms
Time for vanilla implementation: 12.455ms
Time for numpy implementation: 4.107ms
Time for vanilla implementation: 90.988ms
Time for numpy implementation: 0.447ms
Time for vanilla implementation: 336.501ms
Time for numpy implementation: 1.51ms
```

Figure 1: Report of time differences

At n = 1000, we see that the time difference between Vanilla implementation and NumPy implementation is 334.991 ms. NumPy implementation is significantly faster than Vanilla implementation. From the figure above, the differences between vanilla implementation and NumPy implementation are [-0.257 ms, 8.348 ms, 90.541 ms, 334.991 ms]. One reason for this observed difference is that with NumPy, we are using NumPy arrays, which "is a collection of homogeneous data-types that are stored in contiguous memory locations...a list in Python is a collection of heterogenous data types stored in non-contiguous memory locations" (source is Geeks for Geeks article). NumPy also uses parallelism. With our Vanilla solution, we could have made it even slower by using a for loop.

A10. Two random variables X and Y have equal distributions if their CDFs, F_X and F_Y, respectively, are equal, i.e. for all x, $|F_X(x) - F_Y(x)| = 0$. The central limit theorem says that the sum of k independent, zero-mean, variance $1/k$ random variables converges to a (standard) Normal distribution as k tends to infinity. We will study this phenomenon empirically (you will use the Python packages Numpy and Matplotlib). Each of the following subproblems includes a description of how the plots were generated; these have been coded for you. The code is available in the .zip file. In this problem, you will add to our implementation to explore **matplotlib** library, and how the solution depends on n and k.

a. *[2 points]* For $i = 1, \dots, n$ let $Z_i \sim \mathcal{N}(0, 1)$. Let $F(x)$ denote the true CDF from which each Z_i is drawn (i.e., Gaussian). Define $\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{Z_i \le x\}$ and we will choose n large enough such that, for all $x \in \mathbb{R}$,

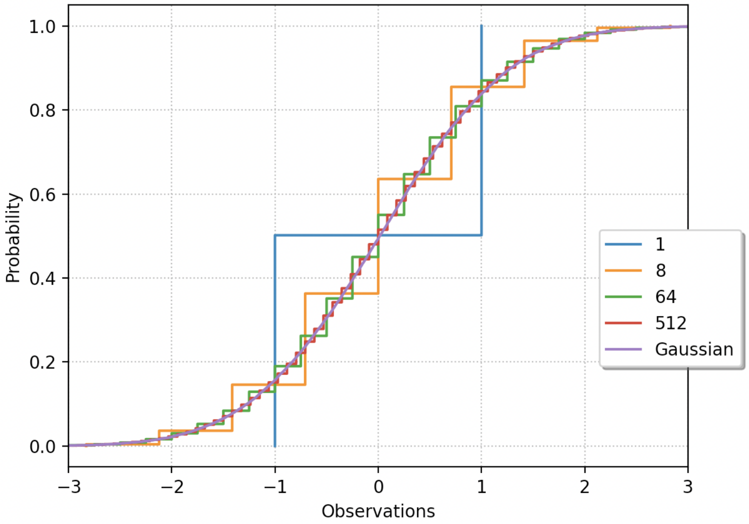
$$\sqrt{\mathbb{E}\left[\left(\widehat{F}_n(x) - F(x)\right)^2\right]} \le 0.0025 \ .$$

Plot $\widehat{F}_n(x)$ from -3 to 3.

$$n = 40000$$

The empirical CDF changes with k according to the central limit theorem (as expected) where as k tends to infinity, we should see convergence to a (standard) normal distribution. From the plot, we indeed see that this is the case.

b. *[2 points]* Define $Y^{(k)} = \frac{1}{\sqrt{k}}\sum_{i=1}^{k} B_i$ where each B_i is equal to -1 and 1 with equal probability and the B_i's are independent. We know that $\frac{1}{\sqrt{k}}B_i$ is zero-mean and has variance $1/k$. For each $k \in \{1, 8, 64, 512\}$ we will generate n (same as in part a) independent copies $Y^{(k)}$ and plot their empirical CDF on the same plot as part a.


Probability
Observations
Gaussian