CSE 446HW0

Rohan Mukherjee

April 2, 2024

1. By Bayes rule,

$$\mathbb{P}(\text{has disease} \mid \text{positive test}) = \frac{\mathbb{P}(\text{pos} \mid \text{disease}) \cdot \mathbb{P}(\text{disease})}{\mathbb{P}(\text{pos})} = \frac{0.99 \cdot 0.0001}{\mathbb{P}(\text{pos})}$$

   By the law of total probability, we have

$$\mathbb{P}(\text{pos}) = \mathbb{P}(\text{pos} \mid \text{disease}) \cdot \mathbb{P}(\text{disease}) + \mathbb{P}(\text{pos} \mid \text{doesn't}) \cdot \mathbb{P}(\text{doesn't})$$
$$0.99 \cdot 0.0001 + (1 - 0.99) \cdot 0.9999$$

   Since $\mathbb{P}(\text{not pos} \mid \text{doesn't}) = 0.99$ we have $\mathbb{P}(\text{pos} \mid \text{doesn't}) = 1 - 0.99$. Combining these results yields: $\mathbb{P}(\text{has disease} \mid \text{positive test}) = \frac{0.99 \cdot 0.0001}{0.99 \cdot 0.0001 + (1-0.99) \cdot 0.9999} \approx 0.009803 \approx .98\%$.

2. (a) Notice that

$$\begin{aligned}\mathrm{Cov}(X, Y) &= \mathbb{E}[XY - X\mathbb{E}[Y] - Y\mathbb{E}[X] - \mathbb{E}[X]\mathbb{E}[Y]] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y] + \mathbb{E}[X]\mathbb{E}[Y] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]\end{aligned}$$

   First, by the law of total expectation,

$$\begin{aligned}\mathbb{E}[Y] &= \int_{\mathbb{R}} \mathbb{E}[Y \mid X = x] p_X(x) dx \\ &= \int_{\mathbb{R}} x p_X(x) dx = \mathbb{E}[X]\end{aligned}$$

   By the same law,

$$\begin{aligned}\mathbb{E}[XY] &= \int_{\mathbb{R}} \mathbb{E}[XY \mid X = x] p_X(x) dx \\ &= \int_{\mathbb{R}} \mathbb{E}[x \cdot Y \mid X = x] p_X(x) dx \\ &= \int_{\mathbb{R}} x\mathbb{E}[Y \mid X = x] p_X(x) dx \quad \text{By lineararity of expectation} \\ &= \int_{\mathbb{R}} x^2 p_X(x) dx = \mathbb{E}\left[X^2\right]\end{aligned}$$

   Thus,

$$\mathrm{Cov}(X, Y) = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 = \mathrm{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$$

(b) In this case, notice that

$$\begin{aligned}
\mathbb{E}[XY] &= \iint_{\mathbb{R}^2} xyp_{X,Y}(x,y)dxdy = \iint_{\mathbb{R}^2} xyp_X(x)p_Y(y)dxdy \\
&= \int_{\mathbb{R}} xp_X(x)dx \cdot \int_{\mathbb{R}} yp_Y(y)dy \quad \text{By Fubini's theorem} \\
&= \mathbb{E}[X] \cdot \mathbb{E}[Y]
\end{aligned}$$

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

3. (a) We shall instead prove the problem in the case where X, Y are discrete random variables. By the law of total expectation,

$$\begin{aligned}
\mathbb{P}(X+Y=z) &= \sum_{x\in\Omega_x} \mathbb{P}(x+Y=z \mid X=x) \cdot \mathbb{P}(X=x) \\
&= \sum_{x\in\Omega_X} \mathbb{P}(Y=z-x \mid X=x) \cdot \mathbb{P}(X=x)
\end{aligned}$$

Since X, Y are independent, we have

$$\mathbb{P}(Y=z-x \mid X=x) = \frac{\mathbb{P}(Y=z-x, X=x)}{\mathbb{P}(X=x)} = \frac{\mathbb{P}(Y=z-x)\mathbb{P}(X=x)}{\mathbb{P}(x=x)} = \mathbb{P}(Y=z-x)$$

Thus,

$$\mathbb{P}(Z=z) = \sum_{x\in\Omega_X} \mathbb{P}(Y=z-x)\mathbb{P}(X=x)$$

Our expression is analogous to the continuous case because the integral in the discrete case becomes a sum and the pmfs in the continuous case become pdfs in the discrete case, which naturally would yield the above expression.

(b) By the formula we just proved (and a small change of variables), we calculate

$$h(z) = \int_0^1 f(x)g(z-x)dx = \int_0^1 g(z-x)dx = \int_{z-1}^{z} g(u)du$$

By definition, if $u < 0$ or $u > 1$ $g(u) \equiv 0$. So $h(z)$ is equivalently 0 for $z < 0$ and $z > 2$. Now we have two cases: if $0 \le z \le 1$,

$$\int_{z-1}^{z} g(u)du = \int_0^z 1du = z$$

And if $1 \le z \le 2$,

$$\int_{z-1}^{z} g(u)du = \int_{z-1}^{1} 1du = 1-(z-1) = 2-z$$

Thus,

$$h(z) = \begin{cases} 0 & \text{for } z < 0 \text{ or } z > 2 \\ z & \text{for } 0 \le z \le 1 \\ 2-z & \text{for } 1 \le z \le 2 \end{cases}$$

4. (a) We choose $a = \frac{1}{\sigma}$ and $b = -\frac{\mu}{\sigma}$. Then clearly,

$$\mathbb{E}\left[\frac{1}{\sigma}(X_1 - \mu)\right] = \frac{1}{\sigma}\mathbb{E}[X_1 - \mu] = 0$$

and

$$\mathrm{Var}\left(\frac{1}{\sigma}(X_1 - \mu)\right) = \frac{1}{\sigma^2}\mathrm{Var}(X_1 - \mu) = \frac{\sigma^2}{\sigma^2} = 1.$$

(b) By linearity of expectation,

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

And, since X_1 and X_2 are independent,

$$\mathrm{Var}(X_1 + 2X_2) = \mathrm{Var}(X_1) + \mathrm{Var}(2X_2) = \sigma^2 + 4\sigma^2 = 5\sigma^2$$

(c) We see that

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

Similarly,

$$\mathrm{Var}\left(\sqrt{n}(\hat{\mu}_n - \mu)\right) = \frac{1}{n}\mathrm{Var}\left(\sum_{i=1}^{n}(X_i - \mu)\right) = \frac{1}{n}\sum_{i=1}^{n}\sigma^2 = \sigma^2$$

5. (a) We use the following row operations to find the row reduced echelon form of A:

$$\begin{pmatrix}1 & 2 & 1\\ 1 & 0 & 3\\ 1 & 1 & 2\end{pmatrix} \to \begin{pmatrix}1 & 2 & 1\\ 0 & -2 & 2\\ 1 & 1 & 2\end{pmatrix} \to \begin{pmatrix}1 & 2 & 1\\ 0 & -2 & 2\\ 0 & -1 & 1\end{pmatrix} \to \begin{pmatrix}1 & 2 & 1\\ 0 & 1 & -1\\ 0 & -1 & 1\end{pmatrix} \to \begin{pmatrix}1 & 2 & 1\\ 0 & 1 & -1\\ 0 & 0 & 0\end{pmatrix}$$

This matrix has precisely 2 linearly independent columns so it has rank 2.
Next, we use these row operations to find the row reduced echelon form of B:

$$\begin{pmatrix}1 & 2 & 3\\ 1 & 0 & 1\\ 1 & 1 & 2\end{pmatrix} \to \begin{pmatrix}1 & 2 & 3\\ 0 & -2 & -2\\ 1 & 1 & 2\end{pmatrix} \to \begin{pmatrix}1 & 2 & 3\\ 0 & -2 & -2\\ 0 & -1 & -1\end{pmatrix} \to \begin{pmatrix}1 & 2 & 3\\ 0 & 1 & 1\\ 0 & -1 & -1\end{pmatrix} \to \begin{pmatrix}1 & 2 & 3\\ 0 & 1 & 1\\ 0 & 0 & 0\end{pmatrix} \to \begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 0 & 0 & 0\end{pmatrix}$$

Once again, since this matrix has precisely 2 non-zero diagonal entries it has rank 2.

(b) The row reduced echelon form calculations above tell us that the first 2 columns of each matrix will form a basis for the column space of each. That is, a basis for the column space of A is

$$\left\{ [1,1,1]^T, [2,0,1]^T \right\}$$

and a basis for the column space of B is

$$\left\{ [1,1,1]^T, [2,0,1]^T \right\}$$

6. (a) One gets

$$[0,2,3]^T + [2,4,3]^T + [4,2,1]^T = [6,8,7]^T$$

(b) Let $A = \begin{pmatrix} 0 & 2 & 4 \\ 2 & 4 & 2 \\ 3 & 3 & 1 \end{pmatrix}$, and the vector $b = [-2,-2,4]^T$. To solve $Ax = b$, we do the following row operations:

$$\left(\begin{array}{ccc|c} 0 & 2 & 4 & -2 \\ 2 & 4 & 2 & -2 \\ 3 & 3 & 1 & -4 \end{array}\right) \to \left(\begin{array}{ccc|c} 2 & 0 & -6 & 2 \\ 2 & 4 & 2 & -2 \\ 3 & 3 & 1 & -4 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & -3 & 1 \\ 2 & 4 & 2 & -2 \\ 3 & 3 & 1 & -4 \end{array}\right)$$
$$\to \left(\begin{array}{ccc|c} 1 & 0 & -3 & 1 \\ 0 & 4 & 8 & -4 \\ 0 & 3 & 10 & -7 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & -3 & 1 \\ 0 & 1 & 2 & -1 \\ 0 & 3 & 10 & -7 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & -3 & 1 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 4 & -4 \end{array}\right)$$
$$\to \left(\begin{array}{ccc|c} 1 & 0 & -3 & 1 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & -1 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & -1 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

Thus our answer is just $x = [-2,1,-1]^T$.

7. (a) We shall calculate first $y^T Ax$ for $A \in \mathbb{R}^{n\times n}$ and $x, y \in \mathbb{R}^n$. Writing

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

We see that

$$Ax = x_1 \begin{bmatrix} A_{11} \\ A_{21} \\ \vdots \\ A_{n1} \end{bmatrix} + x_2 \begin{bmatrix} A_{12} \\ A_{22} \\ \vdots \\ A_{n2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} A_{1n} \\ A_{2n} \\ \vdots \\ A_{nn} \end{bmatrix}$$

Notice that the index of x specifies the column of A. Next,

$$\begin{aligned} y^T Ax &= x_1(y_1 A_{11} + y_2 A_{21} + \cdots) + x_2(y_1 A_{12} + y_2 A_{22} + \cdots) + \cdots + x_n(y_1 A_{1n} + y_2 A_{2n} + \cdots) \\ &= \sum_{j=1}^{n} x_j \left(\sum_{i=1}^{n} y_i A_{ij} \right) = \sum_{i,j=1}^{n} y_i x_j A_{ij} \end{aligned}$$

It is now clear that

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

(b) Fix $1 \leq \ell \leq n$. We wish to calculate $\frac{\partial x^T Ax}{\partial x_\ell}$. Notice that the quadratic form, which we calculated above, will precisely be the sum of all the entries in this matrix:

$$\begin{bmatrix} x_1x_1A_{11} & x_1x_2A_{12} & \cdots & \cdots & x_1x_nA_{1n} \\ x_2x_1A_{21} & x_2x_2A_{22} & \cdots & \cdots & x_2x_nA_{2n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ x_nx_1A_{n1} & x_nx_2A_{n2} & \cdots & \cdots & x_nx_nA_{nn} \end{bmatrix}$$

The terms that contain x_ℓ lie precisely in either the ℓ'th row of the ℓ'th column of this matrix, and precisely one term contains x_ℓ^2. Any term not containing x_ℓ will be annihilated to 0, so we just have to consider those that do. If $i \neq \ell$, then $\frac{\partial}{\partial x_\ell} x_i x_\ell A_{i\ell} = x_i A_{i\ell}$, otherwise $\frac{\partial}{\partial x_\ell} x_\ell^2 A_{\ell\ell} = 2x_\ell A_{\ell\ell}$. Our partial derivative thus becomes (noticing that the $2x_\ell A_{\ell\ell}$ gets split up into the two different terms),

$$\begin{aligned} &x_1A_{\ell 1} + \cdots + x_nA_{\ell n} \\ +&x_1A_{1\ell} + \cdots + x_nA_{n\ell} \end{aligned}$$

Thus,

$$\nabla_x(x^TAx) = \begin{bmatrix} x_1A_{11} + \cdots + x_nA_{1n} \\ x_1A_{11} + \cdots + x_nA_{n1} \\ \vdots \\ x_1A_{n1} + \cdots + x_nA_{nn} \\ x_1A_{1n} + \cdots + x_nA_{nn} \end{bmatrix}$$

Then the first sum in the ℓ'th partial is the ℓ'th entry of the vector Ax since the ℓ is specifying the row of A to do the dot product with. Similarly, the second sum is the ℓ'th entry of the vector A^Tx, since now the ℓ is going to specify the column to dot with. Thus our final answer is $\nabla_x x^TAx = (A + A^T)x$. Through an extremely similar process, writing out the matrix, taking the partials, then thinking very deeply about if it is columns or rows, one can determine that $\nabla_x y^TBx = B^Ty$. I shall write out a different way of deriving it here:

$$\frac{\partial}{\partial x_j} \sum_{i,j=1}^{n} y_i x_j B_{ij} = \sum_{i=1}^{n} y_i B_{ij}$$

Thus the jth entry of the column vector $\nabla x y^T Bx$ is the jth column dotted with y. We can repersent this in matrix form as B^Ty.

Putting it all together, $\nabla_x f(x, y) = (A + A^T)x + B^Ty$ (obviously, the derivative of constants are 0).

(c) No variable of y even appears in the first expression x^TAx so applying the gradient to that will yield equivalently 0. Similarly the derivative of a constant is 0. We need only find $\nabla_y y^T Bx$. In this case, our final function is the sum of every entry of the following matrix:

$$\begin{bmatrix} y_1x_1B_{11} & y_1x_2B_{12} & \cdots & y_1x_nB_{1n} \\ y_2x_1B_{21} & y_2x_2B_{22} & \cdots & y_2x_nB_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_nx_1B_{n1} & y_nx_2B_{n2} & \cdots & y_nx_nB_{nn} \end{bmatrix}$$

When we differentiate w.r.t. y_ℓ, every entry not containing y_ℓ disappears. So the only remaining entries are the ones in the ℓ'th row. This yields that the partial derivative w.r.t. y_ℓ is just $\sum_{j=1}^n x_j B_{\ell j}$. This is precisely the ℓ'th row of B dotted with the variable x. The vector containing that as it's ℓ'th entry in general is just Bx, thus $\nabla_y y^T Bx = Bx$, and by our previous arguments $\nabla_y f(x, y) = Bx$.

8. (a) We claim that $g(x) = x^{-1}$ suffices. This is because:

$$\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix} \begin{bmatrix} w_1^{-1} & 0 & \cdots & 0 \\ 0 & w_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n^{-1} \end{bmatrix} = I_n$$

Since, for example, if the column $[0, \ldots, w_\ell^{-1}, \ldots, 0]^T$ is dotted with any row other than $[0, \ldots, w_\ell, \ldots, 0]$, the only non-zero entries will not line up and instead be each multiplied by 0, yielding 0 as the final answer. On the other hand if you do multiply it with that row you get precisely 1. So the product is exactly equal to the identity, verifying that g indeed works.

(b) We notice that $\|x\|_2^2 = x^T x$, since $x^T x = \sum_{i=1}^n x_i^2$. Then $\|Ax\|_2^2 = (Ax)^T(Ax) = x^T A^T A x = x^T x = \|x\|_2^2$.

(c) Notice that

$$I = BB^{-1} = B^T B^{-1}$$

and similarly that

$$I = B^{-1} B^T$$

Thus B^T has a two-sided inverse, which is also a matrix. We wish to show that $(B^{-1})^T = B^{-1}$. This can be done in the following way:

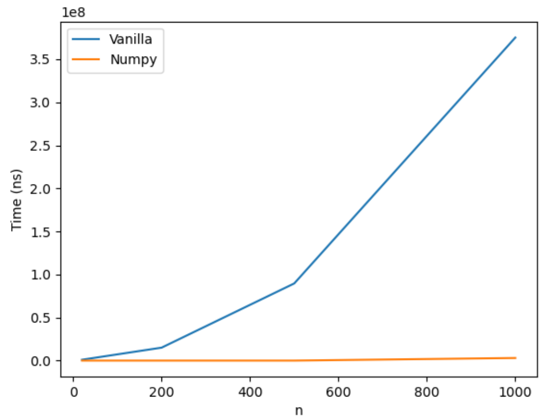
$$I^T = I = (B^{-1}B^T)^T = B(B^{-1})^T$$

Thus $B^{-1} = (B^T)^{-1} = (B^{-1})^T$, as desired.

(d) Let λ be any eigenvalue of C and x the corresponding eigenvector. If x does not have norm 1, normalize it to have norm one by replacing x with $x/\|x\|$. Then,

$$0 \leq x^T C x = x^T \lambda x = \lambda x^T x = \lambda.$$

9. The following plot illustrates the difference in wall-clock time:



   We can see that the numpy implementation stays nearly constant while the vanilla implementation scales linearly. We know that python arrays (even of ints) are arrays of pointers that point to ints in memory, while numpy arrays are blocks in memory (like in c) and that numpy operations can take advantage of parallelism. Since my CPU has many threads, this means that the efficient usage of more threads will cut the time down drastically, while the naive python implementation will only be able to make use of a single thread.

10. (a) The value the code gave me was $n = 40,000$. As seen in the below plot, the theoretical results of the law of large number hold up quite drastically. The empirical values of the CDF very, very quickly go to the normal distribution (even with a relatively small number of variables of $n = 512$).

    (b) Here is my plot: