Homework 1, CS 4220, Spring 2021
Instructor: Austin R. Benson
Due Friday, February 26, 2021 at 3:44pm ET on CMS (before lecture)

Policies

Submission. Submit your write-up as a single PDF on CMS: https://cmsx.cs.cornell.edu.
Coding questions. You can use any programming language for the coding parts of the assignment. Include your code in your write-up.
Typesetting. Your write-up must be typeset with LaTeX— no handwritten homeworks.
Collaboration. Please discuss and collaborate on the homework, but you have to write your own solutions and code.
Resources and attribution. Feel free to use any resources that might be helpful, and provide attribution for any key ideas. We only ask that you work on the problems in earnest. Please do not hunt for solutions with a search engine.

Problems

1. *Fast matrix operations.*
   Let $A, B \in \mathbb{R}^{n\times n}$ be general square matrices and $u, v \in \mathbb{R}^n$ be vectors. Let $D \in \mathbb{R}^{n\times n}$ be a diagonal matrix ($D_{ij} = 0$ if $i \neq j$). Write code snippets that implement the following computations:
   (a) $(I + uu^T)v$ in $O(n)$ time
   (b) $u^T ABABv$ in $O(n^2)$ time
   (c) $\text{trace}(DAD)$ in $O(n)$ time (recall that the trace of a matrix is the sum of its diagonal entries)

   For example, here is a Julia code snippet to compute Du in $O(n)$ time:

```
function Du(D, u)
    n = length(u)
    y = zeros(n)
    for i = 1:n
        y[i] = D[i, i] * u[i]
    end
    return y
end
```

2. *SVD and rank.*
   Let $A \in \mathbb{R}^{m\times n}$ and $k \leq n \leq m$.
   (a) Use the SVD to show that if $\text{rank}(A) = k$, then we can write $A = XY^T$, where $X \in \mathbb{R}^{m\times k}$, $Y \in \mathbb{R}^{n\times k}$, and $\text{rank}(X) = \text{rank}(Y) = k$.
   (b) Show that if $A = XY^T$, where $X \in \mathbb{R}^{m\times k}$, $Y \in \mathbb{R}^{n\times k}$, then $\text{rank}(A) \leq k$.

3. *Norm equivalences and structure.*
   (a) Show that $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$ for any vector $x \in \mathbb{R}^n$.
   (b) Show that $\|A\|_\infty \leq \sqrt{n}\|A\|_2 \leq \sqrt{nm}\|A\|_\infty$ for any matrix $A \in \mathbb{R}^{m\times n}$.
   (c) Let $L = xy^T$ for $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$. Show that $\|L\|_2 = \|L\|_F = \|x\|_2 \cdot \|y\|_2$.
   (d) Let $L = xy^T$ for $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$. Show how to compute $\|L\|_1$ in $O(m + n)$ time.

4. *Permutation matrices.*
   Let $P \in \mathbb{R}^{n \times n}$ be a permutation matrix.
   (a) Show that P^T is also a permutation matrix.
   (b) Show that $P^T = P^{-1}$, i.e., P is an orthogonal matrix.

5. *Gauss transforms.*
   When discussing Gaussian elimination and then LU decompositions in class, we considered matrices $G^{(k)}$ (Gauss transforms) that when applied to a matrix A introduced zeros in the k^{th} column below the diagonal. Assuming all of the matrices involved in this problem are $n \times n$, we may write these matrices compactly as

$$G^{(k)} = I - \ell^{(k)} e_k^T,$$

   where $\ell^{(k)} = [0, \ldots, 0, \ell_{k+1,k}, \ldots \ell_{n,k}]^T$ is zero in the first k entries. We will now prove two properties of these matrices outlined in class.

   (a) Show that $\left(G^{(k)}\right)^{-1} = I + \ell^{(k)} e_k^T$.
   (b) Show that

$$L \equiv \left(G^{(1)}\right)^{-1} \left(G^{(2)}\right)^{-1} \cdots \left(G^{(n-1)}\right)^{-1} = I + \sum_{k=1}^{n-1} \ell^{(k)} e_k^T.$$

   Note that this matrix is lower triangular with ones on the diagonal. The non-zero entries in the kth column of L are simply the non-zero entries in $\ell^{(k)}$.

6. *Floating point error in matmul.*
   For this question, you can assume that there is no underflow or overflow and that the input data is exactly representable in floating point.
   (a) Let $x, y \in \mathbb{R}^n$. Show that $\mathrm{fl}(\sum_{i=1}^n x_i y_i) = \sum_{i=1}^n x_i y_i (1 + \delta_i)$, where $|\delta_i| \leq n\epsilon$.
   (b) Let $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}$ and suppose that we compute the multiplication AB using the inner product algorithm discussed in class. Using part (a), show that $\mathrm{fl}(AB) = AB + E$, where $|E| \leq n\epsilon \cdot |A||B|$ (here, the inequality is entrywise and $|M|$ denotes the matrix with entries equal to the absolute values of the entries of M).

7. *exp*
   Implement a function that approximately computes e^x using the Taylor expansion. Plot the relative error between your computation of e^{-20} and the built-in library implementation, as a function of the number of expansion terms. Use your function in two ways: (i) evaluate the Taylor expansion at $x = -20$ and (ii) use the fact that $e^{-x} = 1/e^x$ and evaluate at $x = 20$. Your plot should include results for both methods.

   Which approach is more accurate? Why?

8. *Matmul implementation.*
   For this question, we will examine the performance of algorithms for computing $C = AB$, where $A, B, C \in \mathbb{R}^{n \times n}$.

   (a) Implement the inner product algorithm manually. Here's a Julia code snippet to start.

```
for i = 1:n
    for j = 1:n
        for k = 1:n
            @inbounds C[i, j] += A[i, k] * B[k, j]
        end
    end
end
```

(b) Implement the row-by-row algorithm using mat-vecs. Here's a Julia code snippet to start.

```
for i = 1:n
    @inbounds C[i:i, :] += A[i:i, :] * B
end
```

(c) Implement the column-by-column algorithm using mat-vecs. Here's a Julia code snippet to start.

```
for j = 1:n
    @inbounds C[:, j] += A * B[:, j]
end
```

(d) Compare the performance of the algorithms from parts (a)–(c) and make a plot to show that they all scale as $O(n^3)$. Also include the performance of the built-in matrix multiplication library function. Explain any performance differences. (In Julia, you can use the `@time` macro to measure running time.)