February 26, 2026

Problem Set 2

Exercise 1: Approximate Matrix Multiplication. A fundamental problem in computing is to find the product of two matrices: that is, given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times p}$, compute $\mathbf{AB}$. In this exercise, we will study a randomized technique for fast approximate matrix multiplication.

1. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times p}$. The product matrix $\mathbf{AB}$ is defined entrywise by

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{n} \mathbf{A}_{ik}\mathbf{B}_{kj}.$$

   From this viewpoint, how many arithmetic operations does it take to compute $\mathbf{AB}$? Suppose addition and multiplication each take one arithmetic operation.

2. An alternative way to view matrix multiplication is as a sum of rank-one outer products. In particular,

$$\mathbf{AB} = \sum_{k=1}^{n} \mathrm{col}_k(\mathbf{A})\,\mathrm{row}_k(\mathbf{B})$$

   where $\mathrm{col}_k(\mathbf{A})$ is the kth column vector of $\mathbf{A}$ and $\mathrm{row}_k(\mathbf{B})$ is the kth row vector of $\mathbf{B}$. Verify that this formulation is equivalent to entrywise matrix multiplication.

Building on this viewpoint, we can estimate $\mathbf{AB}$ by picking a random subset of outer products to form an approximation. Let's analyze a couple variants of this idea.

3. First, suppose we draw an index X uniformly at random: $X \sim U(n)$. Consider the approximate matrix

$$\widehat{\mathbf{AB}} := n \cdot \mathrm{col}_X(\mathbf{A})\,\mathrm{row}_X(\mathbf{B}).$$

   Verify that $\mathbb{E}[\widehat{\mathbf{AB}}] = \mathbf{AB}$.

4. Now, let's generalize this to multiple indices. Let $r \ll n$ be a fixed constant and $X_1, \ldots, X_r$ be i.i.d from $U(n)$. Consider

$$\widehat{\mathbf{AB}} := \frac{n}{r}\sum_{k=1}^{r} \mathrm{col}_{X_k}(\mathbf{A})\,\mathrm{row}_{X_k}(\mathbf{B}).$$

   How many arithmetic operations are needed to compute $\widehat{\mathbf{AB}}$? Assume $X_1, \ldots, X_r$ are sampled beforehand.

5. Compute the expectation and variance of $\widehat{\mathbf{AB}}$. What is the dependence on r?

6. Notice that the variance also depends on the entries of $\mathbf{A}$ and $\mathbf{B}$. Show how to construct matrices $\mathbf{A}$ and $\mathbf{B}$ where the variance of $\widehat{\mathbf{AB}}$ is arbitrarily large.

7. To address the limitations of uniform sampling, we can sample indices *nonuniformly*. As before, let $r \ll n$ be a fixed constant, and fix sampling probabilities $p_1, \ldots, p_n$. Let $X_1, \ldots, X_n$ be i.i.d. random variables drawn with $\Pr[X_i = j] = p_j$. Construct the approximation

$$\widehat{\mathbf{AB}} := \frac{1}{r} \sum_{k=1}^{r} \frac{1}{p_{X_k}} \operatorname{col}_{X_k}(\mathbf{A}) \operatorname{row}_{X_k}(\mathbf{B}).$$

Compute the expectation and variance of $\widehat{\mathbf{AB}}$.

To ensure our approximation $\widehat{\mathbf{AB}}$ is close to the actual matrix product $\mathbf{AB}$, we need to study the *residual*

$$\mathbf{AB} - \widehat{\mathbf{AB}}.$$

In a following problem set, we will show how to pick $p_1, \ldots, p_n$ in order to minimize

$$\|\mathbf{AB} - \widehat{\mathbf{AB}}\|_F^2.$$

This will effectively give a tight bound on how well we can perform approximate matrix multiplication under this sampling scheme. Stay tuned!

Exercise 2: Randomized Turing Machines. Now we are looking at adding randomization to Turing Machines.
Informally, this basically allows a TM to flip a coin if needed to determine its transition (so this only allows 2 possible transitions based on the result of the coin flip). Also note that the TM need not flip coins for every transition.
Consider a deterministic Turing Machine $M = (Q, \Gamma, b, \Sigma, \delta, q_0, F)$; however, now with the modification that $M = (Q, \Gamma, b, \Sigma, \delta_0, \delta_1, q_0, F)$, which is called a PTM. This essentially means that at every turn a coin is flipped. If the answer is 0, then δ_0 is applied, otherwise δ_1 is applied as the transition function instead. Notice that if the outputs of both of functions are the same the coin flip does not matter (which is equivalent to not flipping the coin in the first place).

1. ($\star\star$) Show that any coin with $Pr[Heads] = \rho$ can be simulated by a PTM in expected $O(1)$ time, given that bit i of ρ can be computed in $poly(i)$ time. (Simulated means that the PTM will essentially have the result of a coin flip with $Pr[Heads] = \rho$ at the end of the procedure).

Currently we have not defined the acceptance of this model. This will depend on the complexity class itself.
Let $M(w) = 1$ if the machine accepts and 0 otherwise.
Let $L(w) = 1$ if $w \in L$ and 0 otherwise.
Let **BPP** (Bounded-error polynomial time) be the class that includes all languages L such that there is a poly-time PTM M, where $Pr[M(w) = L(w)] \geq 2/3$.

2. ($\star\star$) Let PIT be the language: Given a polynomial (in the form of an arithmetic formula), determine if it is the zero polynomial. Show that $PIT \in$ **BPP**.

3. ($\star\star$) Show that, this error can be made to be exponentially close to 1, that is $Pr[M(w) = L(w)] \geq 1 - e^{-p(n)}$ for some polynomial $p(n)$.

Let **RP** (Randomized polynomial time) be the class that includes all languages L such that there is a poly-time PTM M, where $w \in L \rightarrow Pr[M(w) = 1] \geq 1/2$ and $w \notin L \rightarrow Pr[M(w) = 1] = 0$. (This is the one-sided error version of **BPP**). Define **coRP** in the a similar manner but $w \in L \rightarrow Pr[M(w) = 0] \leq 1$ and $w \notin L \rightarrow Pr[M(w) = 1] \leq 1/2$.

4. ($\star$) Show that **RP** $\subseteq$ **BPP** and **coRP** $\subseteq$ **BPP**.

5. ($\star\star$) Show that **RP** $\subseteq$ **NP**.

6. ($\star\star$) Show that **coRP** $\subseteq$ **NP**.

7. ($\star\star$) Show that, this one-sided error can be made to be exponentially close to 1, that is $Pr[M(w) = L(w)] \geq 1 - e^{-p(n)}$ for some polynomial $p(n)$.

Let **ZPP** (Zero-error probabilistic poly-time) be the class that includes all languages L such that there is an *expected* poly-time PTM M, where $Pr[M(w) = L(w)] = 1$.

8. ($\star$) Show that if the PTMs were poly-time instead of expected poly-time, then **ZPP** $=$ **P**.

9. ($\star$) Show that **P** $\subseteq$ **ZPP**.

10. ($\star\star$) Show that **ZPP** $=$ **RP** $\cap$ **coRP**.

Let **PP** (Probabilistic poly-time) be the class that includes all languages L such that there is a poly-time PTM M such that $Pr[M(w) = L(w)] \geq 1/2$. (So the error probability can be arbitrarily close to 1/2)

11. ($\star$) Show that **BPP** $\subseteq$ **PP**.

12. ($\star$) Show that **NP** $\subseteq$ **PP**.

13. ($\star\star$) Show that **PP** $\subseteq$ **PSPACE**.

Exercise 3: Connecting Our Complexity Zoo. This uses definitions used in Question 2.

1. ($\star\star$) Show that $\mathbf{BPP}^{\mathbf{BPP}} = \mathbf{BPP}$ and conclude that if $\mathbf{BPP} = \mathbf{NP}$, then $\mathbf{PH} \subseteq \mathbf{BPP}$.
2. ($\star\star\star$) Show that $\mathbf{RP} \subseteq \mathbf{P/poly}$.
3. ($\star\star\star$) Show that $\mathbf{BPP} \subseteq \Sigma_2 \cap \Pi_2$.
4. ($\star\star\star\star$) Show that $\mathbf{BPP} \subseteq \mathbf{P/poly}$.