Assignment 2 — *Applied Algorithms, T. II/2022–23*
(due: mon mar 27 @ 11:59pm)

Ground Rules: Do all problems below. Solve them either by yourself or in teams. For written questions, please typeset your answers. Everything will be handed in on Canvas. You can look up things on the Internet; refrain from copying solutions straight-up.

Most of the problems weren't designed with gradeability in mind. They are generally open-ended. Their main function is for you to have fun and think more deeply about the topic. Therefore, you are encouraged to go above and beyond what is required.

Problem 1. *Equivalent Formulation.* The input to SVM consists of n data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Each $\mathbf{x}_i$ comes with a label $y_i \in \{-1, +1\}$. In class (long time ago), we began deriving SVM by phrasing it as finding the hyperplane that has the largest margin

> Find $(\mathbf{w}, b)$ with $\|\mathbf{w}\| = 1$ to maximize $\gamma := \min_{i=1}^{n} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$ subject to $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$ for all $i = 1, 2, \ldots, n$.

Here the γ term is known as the margin.

Show that the above formulation yields the same hyperplane as the following formulation:

> Find $(\mathbf{w}, b)$ to minimize $\|w\|^2$ subject to $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ for all $i = 1, 2, \ldots, n$.

(*Hint:* First, show that the optimal hyperplane found in the former formulation can be adapted for use in the latter formulation, i.e., feasible—meeting the conditions but not necessarily optimal. Furthermore, if there is a "better" solution in the latter formulation, this can be adapted to an even better solution for the former formulation.)

Problem 2. *Chernoff-Hoeffding With Bounds.* Consider the following variant of Chernoff-Hoeffding bounds (quite similar to what we saw in class):

> **Theorem:** Let $X = X_1 + X_2 + \cdots + X_n$, where X_i's are independently distributed in the range $[0, 1]$. If $\mu = \mathbf{E}[X]$, then
>
> - For all $t > 0$,
>
> $$\mathbf{Pr}[X > \mu + t] \quad \text{and} \quad \mathbf{Pr}[X < \mu - t] \leq e^{-2t^2/n}.$$
>
> - For $\varepsilon > 0$,
>
> $$\mathbf{Pr}[X > (1 + \varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{3}\mu\right)$$
>
> and
>
> $$\mathbf{Pr}[X < (1 - \varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{2}\mu\right)$$

Suppose instead of the true μ, we know upper- and lower- bounds $\mu_L \leq \mu \leq \mu_H$. Prove that the theorem above implies

- For all $t > 0$,

$$\mathbf{Pr}[X > \mu_H + t] \quad \text{and} \quad \mathbf{Pr}[X < \mu_L - t] \leq e^{-2t^2/n}.$$

- For $\varepsilon > 0$,
$$\mathbf{Pr}[X > (1+\varepsilon)\mu_H] \le \exp\left(-\frac{\varepsilon^2}{3}\mu_H\right)$$
and
$$\mathbf{Pr}[X < (1-\varepsilon)\mu_L] \le \exp\left(-\frac{\varepsilon^2}{2}\mu_L\right)$$

Problem 3. *Rescaling Trick..* The bounds above require that each X_i lies within $[0,1]$. We'll generalize this. Let $X_1, X_2, \dots, X_n$ be independent random variables like before, except this time, assume that each $X_i \in [a,b]$ for $a \le b \in \mathbb{R}$. Derive Chernoff-Hoeffding bounds for this setting by defining Y_i that rescales X_i so that $Y_i \in [0,1]$.

Problem 4. *χ^2 With n Degrees of Freedom.* Let $X_1, \dots, X_n \sim N(0,1)$ be independent, identically and normally distributed random variables, with mean 0 and variance 1. Let
$$\chi_n^2 = \sum_{i=1}^{n} X_i^2.$$
Show the following:

(i) $\mathbf{E}\left[\chi_n^2\right] = n$.

(ii) If $X = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}^\top$ and $0 \le \delta < 1$, then
$$\mathbf{Pr}\left[\left|\|X\|_2^2 - n\right| \ge \delta n\right] \le 2e^{-\delta^2 n}.$$

Problem 5. *Simple Samplers.* Suppose X is a random variable that takes on values in the interval $[0,1]$. Let $\mathbf{E}[X] = c$. The problem is, initially, you don't know anything about c, or about the probability distribution of X. However, you are given a black-box that every time you query it, it gives you an independent random sample drawn according to X. You want to estimate c.

As an example, imagine that you are given a coin with unknown bias (i.e., probability of heads) c. Then, every time you flip it you get heads with probability c, and tails with probability $1-c$ (and this is independent over different flips). If X denotes the outcome of a coin toss (1 for heads and 0 for tails), then $\mathbf{E}[X] = c$. You want to estimate the coin bias c.

Here is a natural scheme to estimate c: Sample from the black-box N times. Let's call these samples $X_1, X_2, \dots, X_N$—and return the *empirical mean* $\hat{c} := \frac{1}{N}\sum_{i=1}^{N} X_i$. In this problem, you'll explore, just how big the number of samples N has to be so that for $\delta, \varepsilon > 0$,
$$\mathbf{Pr}[|\hat{c} - c| \le \varepsilon] \le 1 - \delta.$$
(In statistics/machine learning speak, how many samples do we need to be within error ε with confidence $1-\delta$?)

Problem 6. *Dual Binary Search and Better Merge Sort.* Consider the implementation of `kth(A, B, k)` in the given handout. We will analyze its cost and use it to build merge sort with better span.

(i) Show that `kth(A, B, k)` runs in $O(\log|A| + \log|B|)$ work and span. (*Hint:* Can we guarantee that something goes down in half in each recursive call?)

(ii) We'll start by improving the span bound of `merge`. To merge two sorted sequences, we'll split them into $\sqrt{n}$ equal-sized pieces (previously, we split them into 2 equal-sized pieces). Describe how you would use the `kth` routine to accomplish this. Give a (pseudocode) implementation of the upgraded `merge`.

(iii) What is the work recurrence for our upgraded `merge`? How about span? (*Hint:* You should get $W(n) = \sqrt{n}W(\sqrt{n}) + \ldots$ and $S(n) = S(\sqrt{n}) + \ldots$.)

(iv) Let $f(n)$ be a nondecreasing function such that $0 < \frac{f(n)}{\sqrt{n}f(\sqrt{n})} < 1$ is always true for sufficiently large n. Solve the following recurrence: $W(n) = \sqrt{n}W(\sqrt{n}) + f(n)$. (*Hint:* $\Theta(n)$)

While the above is true, if you find this problem too technical, feel free to solve the following version instead: $W(n) = \sqrt{n}W(\sqrt{n}) + n^{1-\beta}$, where $\beta \in (0, 1)$.

Bonus: For those who took analysis/advanced calc already, prove/disprove: if $f(n) = o(n)$—read little-O of n—is a monotonically increasing function, then for sufficiently large n, $0 < \frac{f(n)}{\sqrt{n}f(\sqrt{n})} < 1$ always holds.

(v) Now that we know how to solve the recurrences, what are the work and span bounds for our upgraded `merge`?

(vi) If merge sort now uses the improved `merge` routine, what is its overall work and span?

Problem 7. *String Comparison.* Derive a parallel algorithm `compare(X, Y)` that lexicographically compares two given strings $X = x_1x_2\ldots x_n$ and $Y = y_1y_2\ldots y_m$ and returns the following result

$$\texttt{compare}(X,Y) = \begin{cases} -1 & \text{if } X < Y \\ 0 & \text{if } X = Y \\ +1 & \text{if } X > Y \end{cases}.$$

Your algorithm should run in $O(\min(n, m))$ work and $O(\log \min(n, m))$ span.

Problem 8. *Median of Means.* Median of means (MoM) is a popular trick in amplifying the sharpness of an estimate. Suppose you wish to estimate a quantity τ and you have come up with an algorithm A that returns T such that $\mathbf{E}[T] = \tau$ and $\mathbf{E}\left[(T - \tau)^2\right] = \beta$.

To use MoM, run K independent copies of algorithm A, which will result in estimates $T_1, T_2, \ldots, T_K$. By definition, each of these estimates has mean τ—i.e., $\mathbf{E}[T_i] = \tau$. Also, $\mathbf{E}\left[(T_i - \tau)^2\right] = \beta$. Then, the median of means estimate is simply

$$\widehat{T} := \mathsf{median}(T_1, T_2, \ldots, T_K)$$

This estimate $\widehat{T}$ is hopefully much sharper than T. How many copies of A do we need so that we can guarantee $\mathbf{Pr}\left[|\widehat{T} - \tau| < \varepsilon\right] \geq 1 - \delta$? (*Hint:* **(i)** Chebyshev's inequality can tell us how likely an estimate from A deviates from τ by more than ε. **(ii)** Chernoff-Hoeffding can then be applied: the median will be off by more than ε if at least $K/2$ copies are off.)

Problem 9. *Skip List Space Bounds..* You showed earlier that a skip list with n keys uses $\Theta(n)$ space in expectation. Let n_i be the number of keys that remain in level i, so $n_0 = n$ (the bottom level has all the keys).

(i) Show that $\mathbf{E}[n_t] \leq \frac{n}{2^t}$.

(ii) Show that for $k \leq \ln\left(\frac{n}{\ln n}\right)$, the probability

$$\mathbf{Pr}\left[n_k \geq (1+\lambda)\frac{n}{2^t}\right] \leq \frac{1}{n^c},$$

where $c = \Theta(\lambda^2)$.

(iii) Use this to conclude that the total space requirement of a skip list with n keys is $\Theta(n)$ with probability at least $1 - 1/n^2$. (*Hint*: When the number of keys first drops below $\ln n$, every subsequent level at least one fewer key than the level before.)