Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning

Instructor: Purushottam Kar
Date: January 31, 2018
Total: 60 marks

ASSIGNMENT 1

Instructions

1. Only electronic submissions to Gradescope `https://gradescope.com` will be accepted. No hard copy or emailed submissions would be accepted.
2. Your submission should be a single PDF file compiled using the LaTeX style file `saltsubmit.sty`.
3. No zip/tar/png/jpg files or handwritten and then scanned submissions will be accepted.
4. Submissions will be accepted till February 14, 2018, 2359 hrs, IST.
5. Late submissions will be accepted till February 16, 2018, 2359 hrs, IST.
6. Late submissions will incur a penalty – the maximum marks for a late submission will be 80% of the total marks, even if solutions to all questions are absolutely correct.
7. We will be closely checking your submissions for instances of plagiarism.
8. You may be penalized if you do not follow the formatting instructions carefully.
9. Your answer to every question should begin on a new page. The style file is designed to do this automatically. However, if it fails to do so, use the `\clearpage` option in LaTeX before starting the new question, to enforce this.
10. An account has been created for you on Gradescope. Use your IITK CC ID (not GMail, CSE etc IDs) to login and use the "Forgot Password" option to set your password initially.
11. While submitting your assignment on this website, you will have to specify on which page(s) is question 1 answered, on which page(s) is question 2 answered etc. To do this properly, first ensure that the answer to each question starts on a different page.
12. Be careful to flush all your floats (figures, tables) corresponding to question n before starting the answer to question $n+1$ otherwise graders might miss your figures and award you less points. Remember, different people may grade question n and $n+1$.
13. Your solutions must appear in proper order in the PDF file i.e. your solution to question n must be complete in the PDF file (including all plots, tables, proofs etc) before you present a solution to question $n+1$.

Problem 1.1 (The search for fairness ends here). Alice needs fair coin tosses to complete a programming assignment in her randomized algorithms course. She has access to K coins $C_1, C_2, \ldots, C_K$ with biases $p_1, p_2, \ldots, p_K$. She has no idea about the biases of these coins but knows that at least one of these coins (maybe more) is fair i.e. for at least one $i \in [K], p_i = 0.5$. Can you help Alice obtain fair (or almost fair) coin tosses using these K coins?

More specifically, design an algorithm called SAMPLE. Whenever SAMPLE is invoked, it is allowed to toss any/all of the K coins any number of times, and use all their outcomes to produce a single heads or tails verdict. Analyze your algorithm to give a guarantee on $\mathbb{P}[\text{SAMPLE}() = H]$. Remember, what we desire is that $\mathbb{P}[\text{SAMPLE}() = H] \approx 0.5$.

Your can (also) have an algorithm called PREPROCESS using which Alice can play around with these K coins to test them a bit before SAMPLE starts getting invoked. Is it possible to ensure $\mathbb{P}[\text{SAMPLE} = H] = 0.5$? Solutions that are less expensive in terms of coin tosses and produce SAMPLE results that are more fair will get more credit. (10+10 marks)

Problem 1.2 (Why adding two random variables makes sense). Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Show that the set of all real-valued random variables X on this probability space that take finite values almost surely, forms a vector space (over the set $\mathbb{R}$ as scalars). Be careful to verify all properties that a random variable (see Lecture 1, Definition 1.6 and footnotes 1 an 2), and a vector space must satisfy while giving your solution. (10 marks)

Problem 1.3 (An overcomplicated definition of independence). Show that two events are independent if and only if the σ-subalgebras they generate are independent (see Lecture 1, Definition 1.2 and Remark 1.10). (10 marks)

Problem 1.4 (Social Network Analysis). A company that manages a large social messaging network of N users, wishes to find out how many messages do an average pair of users send each other. For any two users $i, j \in [N]$, the company has a count c_{ij} of how many messages have they sent each other. Assume $0 \leq c_{ij} \leq K$ for $i \neq j$ and $c_{ii} = 0$ for sake of simplicity. Thus, the company wishes to find out $\mu := \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j>i} c_{ij}$. Finding this quantity explicitly will take $\mathcal{O}\left(N^2\right)$ time which is prohibitive since N is large.

A SALT alumni who is employed in that company suggested that the company estimate μ by first sampling $n \ll N$ users with replacement, say $I_1, I_2, \ldots, I_n$, and then estimating μ using $\hat{\mu}_S := \frac{1}{n(n-1)} \sum_{j=1}^{n} \sum_{k>j} c_{I_j I_k}$, calculating which takes only $\mathcal{O}\left(n^2\right)$ time.

1. Would you instead like to sample S without replacement?
2. Would you like to stick with the given definition of $\hat{\mu}_S$ or would you like to tweak it a bit?
3. For the sampling strategy chosen by you and the definition of $\hat{\mu}_S$ chosen by you, show that $\hat{\mu}_S \approx \mu$ with high probability.

Give a proper quantified version of your result (in the form Chernoff, Hoeffding inequalities are presented). You may leave small constants such as $2, 32, 15$ etc unspecified but dependence on accuracy/tolerance and confidence parameters must be exactly presented.

While analyzing the above problem, be careful to note that the random variables $c_{I_1 I_2}$ and $c_{I_1 I_3}$ are not independent even if I_1, I_2, I_3 were independently sampled since they contain I_1 in common that couples them. (20 marks)