Institut for Matematik og Datalogi
Syddansk Universitet, Odense

12 November 2015
JBJ

DM551 – 2. Exam assignment

Hand in by Friday December 4 at 14:15.

Rules

This is the second of two sets of problems which together with the oral exam in January constitute the exam in DM551. This second set of problems may be solved in groups of up to three persons. Any collaboration between different groups will be considered exam fraud. Thus you are not allowed to show your solutions to fellow students other than those in your group and you may not discuss the solutions with other groups. On the other hand, you can learn a lot from discussing the problems with each other so you may do this to some extend, such as which methods can be used or similar problems from the book or exercise classes.

Remember that this counts as part of your exam, so do a good job and try to answer all questions carefully. It is important that you **argue so that the reader can follow your calculations and explanations**.

How to hand in your report

Your report must be handed in by Friday December 4 at 14.15 **both**

- as a **paper** copy to the instructor **and**
- as a **pdf** file via Blackboard. Formats other than PDF will not be accepted.

On the first page you must write your **name** and **CPR-number**.
Handin via Blackboard is done as follows:

- Choose “DM551, Algoritmer og Sandsynlighed, efterår 15”.
- click on “SDU Assignment” (in the left menu).
- Fill in the form and attach your PDF file. Finish by clicking ’submit’
- Blackboard will send you a receipt by email.

Problems

Solve the following problems and **Remember to justify all answers:**

Problem 1 (25 points)

In Kleinberg and Tardos section 13.4 you saw a randomized approximation algorithm $\mathcal{B}$ for MAX-3-SAT whose expected number of satisfied clauses is within a factor $\frac{7}{8}$ of optimal. In fact we showed that if there are m clauses in the input, then the expected number of clauses that will be satisfied by $\mathcal{B}$ is $\frac{7 \cdot m}{8}$ and from this we proved (using the probabilistic method) that every instance of 3-SAT on m clauses has a truth assignment which satisfies at least $\frac{7 \cdot m}{8}$ clauses. We also saw that the expected number of times we need to guess a truth assignment (run $\mathcal{B}$) before obtaining a truth assignment which satisfies at least $\frac{7 \cdot m}{8}$ clauses is $8m$ times.

Question a:

Suppose we want to use the same idea to find a randomized approximation algorithm for MAX-k-SAT, where $k \geq 3$.

(a1) What is the expected number R of clauses we will satisfy if there are m clauses?

(a2) How many times should be expect to run our algorithm before we obtain a truth assignment which satisfies R clauses?

Question b:

Implement the algorithm $\mathcal{B}$. Explain briefly how you obtain the randomness in the truth assignment.

Question c:

Implement an algorithm $\mathcal{C}$ which given integers n, m with $n < m$ generates a "random-like" instance of MAX-3-SAT. You should explain how you achieve this.

Question d:

Perform experiments with your implementation of $\mathcal{B}$ on data from your algorithm $\mathcal{C}$ in order to estimate the number of times you need to run $\mathcal{B}$ before you obtain a truth assignment which satisfies at least $\frac{7 \cdot m}{8}$ clauses. Of course you should repeat this experiment many times for the same input and on many different instances. Besides testing on the random-like instances, you should also test on instances of the following kind:

- Instances with $n+2$ variables $x_1, x_2, \ldots, x_n, a, b$ and $4n$ clauses defined as follows: for each $i = 1, 2, \ldots, n$ we have the 4 clauses $(x_i \vee a \vee b), (x_i \vee \bar{a} \vee b), (x_i \vee a \vee \bar{b}), (x_i \vee \bar{a} \vee \bar{b})$
- Instances with $3n$ variables $x_1, \ldots, x_n, y_1, \ldots, y_n, z_1, \ldots, z_n$ and exactly n^3 clauses of the form $(x_i \vee y_j \vee z_k)$.

Note that for both of these kinds of instances you can test the algorithms performance without actually generating the set of clauses. Explain how!

Of course you should discuss the results that you obtain.

Problem 2 (25 points)

Let $G = (V, E)$ be an undirected graph with no parallel edges. An **orientation** of G is a digraph $D = (V, A)$ which we can obtain from G by replacing each edge $uv \in E$ by one of the arcs $u \to v$, $v \to u$.

Let $D = (V, A)$ be an orientation of an undirected graph $G = (V, E)$. A ordered triple (x, y, z) of distinct vertices of D is **in-bad** if A contains the arcs $x \to y$, $z \to y$ but there is no arc between x and z. Similarly, an ordered triple $x, y, z \in V$ of distinct vertices of D is **out-bad** if A contains the arcs $y \to x$, $y \to z$ but there is no arc between x and z.

For a given undirected graph $G = (V, E)$ we say that it has an **in-good** respectively an **out-good** orientation if there exists an orientation $D = (V, A)$ of G which no triple of vertices is in-bad, respectively is out-bad. Finally, we say that G has a **good** orientation if there is an orientation $D = (V, A)$ of G which is both in-good and out-good.

For a given input graph $G = (V, E)$ we denote by $\mathcal{E}(G)$ be the set of all ordered triples (x, y, z) of vertices from G with the property that $xy, yz \in E$ and $xz \notin E$. For example, if G is just a 4-cycle $v_1v_2v_3v_4v_1$, then $\mathcal{E}(G)$ contains the 4 triples $(v_1, v_2, v_3), (v_2, v_3, v_4), (v_3, v_4, v_1), (v_4, v_1, v_3)$. Note also that this graph does have an in-good orientation, namely $v_1 \to v_2 \to v_3 \to v_4 \to v_1$ and this orientation is also out-good.

Consider the following randomized algorithm $\mathcal{A}$ for constructing an in-good orientation of a given graph $G = (V, E)$ with m edges (the value of α is to be determined later):

1. Randomly orient every edge $uv \in E$.
2. Repeat up to $2\alpha m^2$ times, terminating if there is no in-bad triple left:
   2.1 Chose an arbitrary in-bad triple $(x, y, z) \in \mathcal{E}(G)$
   2.2 randomly pick one of the two edges of the triple and reorient it (e.g. change $x \to y$ to $y \to x$).
3. If an in-good orientation has been found, return it.
4. Otherwise, return that G has no in-good orientation.

Question a:

Suppose the input graph G with m edges does have an in-good orientation. Prove that the expected number of reorientation steps until the algorithm finds an in-good orientation is at most $O(m^2)$. Hint: consider some in-good orientation $D^* = (V, A^*)$ and measure how far the current orientation $D' = (V, A')$ is from agreeing with D^*.

Question b:

Explain why the algorithm $\mathcal{A}$ always return the correct answer if G does not have an in-good orientation.

Question c:

Which value of α should we choose if we want the error probability of $\mathcal{A}$ to be at most 2^{-10}? You must prove your claim.

Question d:

Recall that an orientation is good if it is both in-good and out-good. Explain how to modify the algorithm $\mathcal{A}$ to an algorithm $\mathcal{A}'$ which with high probability will return a good orientation of the given input graph G provided it has such an orientation. What does the modification imply for your analysis of the expected running time? Does it get tighther or looser?

Problem 3 (15 points)

This problem considers robustness of a score (grade) given to students based on a multiple choice test. The problem set-up is based on the paper Frandsen and Schwartzbach, *A singular choice for multiple choice,* `[http://dl.acm.org/citation.cfm?id=1189164]`.

Assume a multiple choice test consists of n questions, each having 4 choices. For each question precisely one choice is correct. Students are allowed to make 0 or 1 "check" (cross) for each question. The score for a question is 1 if the student has checked the correct choice, $-\frac{1}{3}$ if the student has checked a wrong choice and 0 if no choices are checked. The score for the test is computed as the sum of the scores for all questions. The maximal score is therefore n. We assume that the test is used only to decide pass/fail, and the threshold for passing is a 50% score, i.e. a score $\geq \frac{n}{2}$.

Define a **challenged** student to be a student that knows the answers to at most 40% of the questions.

Let us assume that a challenged student leaves no questions unanswered. Then clearly he has nothing to lose by guessing the answers to the questions he does not know. So assume that he accordingly puts down checks at uniformly random choices (one per question).

Define a multiple choice test to be **good**, if the probability that a challenged student passes is at most 5%.

A teacher has to make a test, and naturally he wants it to be good. He suspects that if he has enough questions in the test then it will be good. This is indeed correct as we shall see below.

Consider a challenged student and assume that he guesses uniformly at random one of the 4 possible answers to each of the $m = \frac{3}{5}n$ questions to which he does not know the answer. Define

$$X_i = \begin{cases} 1, & \text{if the } i\text{th guess is correct} \\ 0, & \text{otherwise} \end{cases}$$

Define $X = \sum_{i=1}^{m} X_i$

Question a:

Determine $E[X]$.

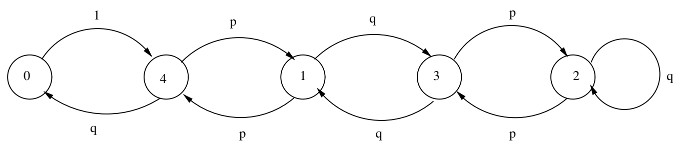
Question b:

Show that the challenged student only passes if $X \geq \frac{3}{2}E[X]$.

Question c:

Using the Chernoff bound technique, determine a size n of the test for which the challenged student only passes with probability at most .05.

Problem 4 (20 points)



Figur 1: A Markov chain. The states are drawn in the order shown to make the drawing look nicer. The numbers $1, p, q$ are transition probabilities so $p + q = 1$.

Jørgen has 4 umbrellas, some at home, some in the office. He keeps moving between home and office. He only takes an umbrella with him if it rains. If it does not rain he leaves the umbrellas behind (at home or in the office). It may happen that all umbrellas are in one place, Jørgen is at the other, it starts raining and he must leave, so he gets wet. Assume now that Jørgen has been using this strategy (take an umbrella if it rains, otherwise not) for a long time. Jørgen wants answers to the following questions.

1. If the probability of rain is p, show that the probability that Jørgen gets wet is given by $\frac{p(1-p)}{5-p}$.
2. Suppose the probability of rain is $p = 0.6$. What is the probability that Jørgen will get wet if he has 4 umbrellas?
3. With the probability of rain being $p = 0.6$ how many umbrellas should Jørgen have so that, if he follows the strategy above, the probability he gets wet is less than 0.01?

To answer the questions/claims, consider a Markov chain taking values in the set $S = \{i : i = 0, 1, 2, 3, 4\}$, where $i \in S$ represents the number of umbrellas in the place where Jørgen is currently at (home or office).

Question a:

Explain why the digraph in Figure 1 represents a Markov chain model for the problem.

Question b:

Find the stationary distribution $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$ and use this to answer the first and second question.

Question c:

Generalize the Markov chain above to states $0, 1, 2, \ldots, N$, write up the stationary distribution and show that the probability of getting wet when Jørgen has N umbrellas is given by $\frac{p(1-p)}{N+1-p}$. Use this to answer the third question.

Question d:

Explain what happens if we take $p = 1$ and why that is not an error of the model.

Problem 5 (15 points)

This is about solving linear recurrence equations.

Question a:

Solve the linear recurrence equation $a_n = 4a_{n-1} - 4a_{n-2}$ with initial conditions $a_0 = 1$, $a_1 = 4$.

Question b:

Solve the linear recurrence equation $a_n = 4a_{n-1} - 4a_{n-2} + n^2$ with initial conditions $a_0 = 22$ and $a_1 = 43$.