max planck institut informatik

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Antonios Antoniadis and Marvin Künnemann **Winter 2018/19**

Exercises for Randomized and Approximation Algorithms

www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter18/rand-apx-algo/

Project: Average-Case Analysis of Orthogonal Vectors

In lieu of lecture and exercise session on January 8th and 10th
Discussion takes place on **January 24th, 2018**

This project asks you to ponder about research-level questions and involves a certain amount of study on your own. Some questions may be challenging, but don't despair: We don't expect you to solve all questions. Rather, we invite you to attack interesting questions on your own, giving you the opportunity to earn bonus points, and more importantly, to impress us ;).

To obtain bonus points, please bring a write-up of your (partial) answers on January 24th and participate in the discussion.

The following problem is a fundamental polynomial-time problem.

Orthogonal Vectors problem (OV): Given are two sets of n d-dimensional 0-1-vectors, i.e., $A = \{a_1, \ldots, a_n\}, B = \{b_1, \ldots, b_n\} \subseteq \{0,1\}^d$. The task is to determine whether there exists an *orthogonal pair*, i.e., a pair $a_i \in A, b_j \in B$ with inner product 0. Note that the inner product $\sum_{k=1}^{d} a_i[k] \cdot b_j[k]$ is 0 if and only if for each coordinate $k \in [d]$, at least one of $a_i[k]$ and $b_j[k]$ is 0.

By checking all pairs, we obtain a simple $O(n^2 d)$-time algorithm. Also, there is a simple $O(2^d n)$-time algorithm (**warm-up question:** How does it work?). A fundamental open question is the following: If $d = \omega(\log n)$, do we need essentially quadratic time in n? More precisely, there is the hypothesis that no algorithm solves OV in time $O(n^{2-\varepsilon}\text{poly}(d))$ for some constant $\varepsilon > 0$. In this project, we want to show that *average* instances of OV are simpler, i.e., solvable in time $O(n^{2-\varepsilon})$ for some $\varepsilon > 0$.[1]

Exercise 1 *(15 Bonus Points)* Let $p = p(n)$ be a probability possibly depending on n. We generate an input randomly by choosing, *independently for each* $i, j \in [n]$, $k \in [d]$,

$$a_i[k] = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1-p, \end{cases} \qquad b_j[k] = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1-p. \end{cases}$$

a) *(Warm-up)* What is the expected number of orthogonal pairs?

[1] Throughout the project, we assume that $d = n^{o(1)}$, which means that the input has almost-linear size $O(nd) = n^{1+o(1)}$.

b) *(10 Bonus Points)* Give as tight as possible upper and lower bounds on the probability that at least one orthogonal pair exists (this value depends on n, p, d).
Self-study task: Read up on the so called first- and second-moment methods!

c) *(5 Bonus Points)* Let $0 \leq q \leq 1$. Consider the problem of choosing p such that the probability that at least one orthogonal pair exists is exactly q. Fix n and d to some values, e.g., $n = 20, d = 6$. Generate a graph that plots your upper and lower bounds on p following from b) against the true value (that you can compute numerically) for q ranging from 0 to 1.

Exercise 2 *(10 Bonus Points)* Let $0 < p < 1$ be a fixed constant. We call an OV algorithm an *Average-Case OV Solver* if it has the following guarantees: (1) the probability that it correctly decides whether the given instance has an orthogonal pair is $1 - o(1)$, and (2) its expected running time is $O(n^{2-\varepsilon})$ for some constant $\varepsilon > 0$. Here, the probability is taken over the instances generated randomly (as described in Exercise 1).

a) *(Warmup)* Show that if d is very large (e.g., $d = \Omega(\log^2 n)$), we can give an Average-Case OV Solver running in constant time, since with high probability no orthogonal pair exists. Give an analogous solver for small d. Try to optimize the bounds on d that you need!

b) *(10 Points)* Give an Average-Case OV Solver for general d that is as fast as possible.
A possible approach:

Step 1 Compute the expected number of ones of u and v *conditioned on* u *and* v *being orthogonal.*

Step 2 Show that you only need to regard vectors with sufficiently few ones to decide an Average-Case OV instance with high probability.