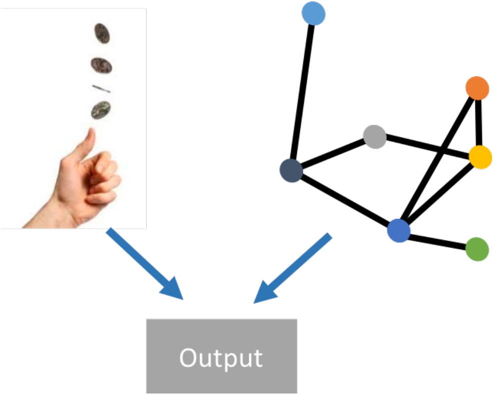
COMPSCI 614: Randomized Algorithms with Applications to Data Science

Prof. Cameron Musco

University of Massachusetts Amherst. Spring 2024.
Lecture 1

Motivation

Randomized algorithms take steps that depend on both their inputs and on the outcomes of random coin flips. I.e., they are algorithms that make random decisions during their execution.



Output

Motivation

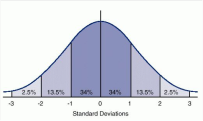
In many settings randomized algorithms give big advantages over deterministic ones:

- Can be much faster than the best known deterministic algorithms (polynomial identity testing, approximation algorithms, linear algebraic computation and data analysis)
- Often very simple and elegant (randomized quicksort, Karger's min–cut algorithm)
- In many cases, by using randomness, we can achieve things that are simply impossible for deterministic algorithms (sublinear time algorithms, efficient communication protocols, small-space streaming algorithms)

What We'll Cover

What We'll Cover

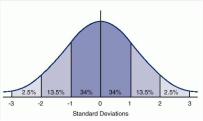
Section 1: Probability Foundations & Random Hashing



DOC
XLS
a4276cd
24e21f5
75282ca
4298375

What We'll Cover

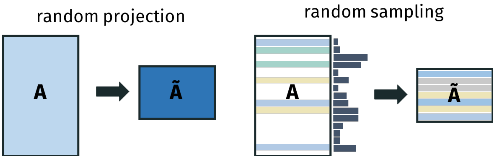
Section 1: Probability Foundations & Random Hashing



- Review of probability tools and concentration inequalities, that will be used throughout the course.
- Classic probability problems like coupon collector and balls into bins.
- Applications to the analysis of random hashing algorithms. E.g., analysis of chaining and linear probing.
- Hashing for efficient communication and lookup. Fingerprints, pattern matching, communication complexity.

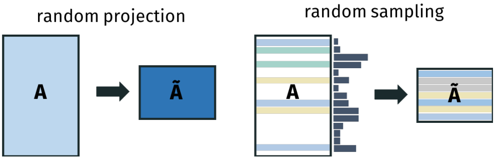
What We'll Cover

Section 2: Random Sketching and Randomized Numerical Linear Algebra (RandNLA)



What We'll Cover

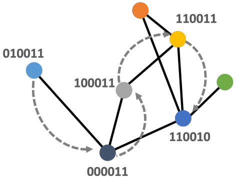
Section 2: Random Sketching and Randomized Numerical Linear Algebra (RandNLA)

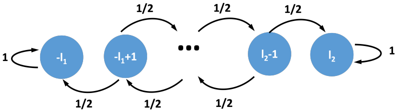


- Sketch based streaming algorithms, including frequency estimation and graph sketching.
- Sampling and sketching for approximate matrix multiplication, low-rank approximation, and trace estimation.
- Dimensionality reduction. The Johnson-Lindenstrauss lemma, subspace embedding, and sketching for linear regression.
- Importance sampling and leverage scores. Connections to matrix concentration bounds and spectral graph theory.

What We’ll Cover

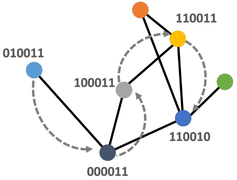
Section 3: Markov Chains

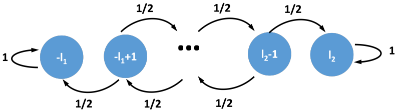




What We'll Cover

Section 3: Markov Chains





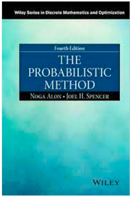
- Introduction to Markov chains and basic tools for their analysis.
- Analysis of Markov chain mixing time via coupling.
- Markov chain based methods for 2-SAT and 3-SAT

$2^n \quad \left(\frac{4}{3}\right)^n = 1.33^n$

- Markov Chain Monte Carlo methods (MCMC) for approximate sampling and counting.

What We’ll Cover

Section 4: Other Topics



Wiley Series in Discrete Mathematics and Optimization

Fourth Edition

THE PROBABILISTIC METHOD

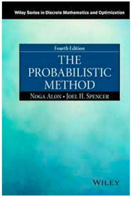
NOGA ALON • JOEL H. SPENCER

WILEY

?

What We’ll Cover

Section 4: Other Topics


Wiley Series in Discrete Mathematics and Optimization
Fourth Edition
THE PROBABILISTIC METHOD
Noga Alon • Joel H. Spencer
Wiley

?

- Convex relaxation and randomized rounding for approximating combinatorially hard problems.
- Mathematical proofs via randomized algorithms: the probabilistic method and the Lovász Local Lemma.
- Other topics you would like to see covered – let me know.

Style of the Course

This is a Ph.D. level theory course.

Style of the Course

This is a Ph.D. level theory course.

- Assignments will emphasize algorithm design, correctness proofs, and asymptotic analysis (minimal coding).
- The homework will be challenging. You will design and analyze algorithms using the tools taught in class, but the solutions will require significant thought and novel ideas.
- A strong algorithms and mathematical background **are required** (particularly in linear algebra and probability).
- If you are an undergraduate or Master's student I would not recommend taking this course unless you have previously taken either 514 or 611 and done very well – the course will be a significant step up in difficulty from 514.

Who Should Take this Course?

- You are a Ph.D. student that wants to do research in algorithms/theory.
- You are a Ph.D. student in another area that wants to apply randomized algorithms or probabilistic analysis in your work.
- You are an undergrad/Master's student excited about algorithms, who potentially wants to pursue a Ph.D.
- If you are unsure if you are prepared for the course, or if it will fulfill your goals for taking it, shoot me an email or Piazza message.

Course Logistics

See course webpage for logistics, course schedule, policies, lecture notes, assignments, etc.

http://people.cs.umass.edu/~cmusco/CS614S24/

Visit my homepage for this link if you lose it.

Personnel

Professor: Cameron Musco

- Email: cmusco@cs.umass.edu
- Office Hours: Thursday, 11:30am-12:30pm (directly after class). CS 234. Starting next week.
- I encourage you to come as regularly as possible to ask questions/work together on practice problems.
- If you need to chat individually, please email to set up a time.

TA: Weronika Nguyen

- Email: thuytrangngu@cs.umass.edu
- Office Hours: Tuesday, 3:00pm-4:00pm. CS 207.

Piazza and Participation

We will use Piazza for class discussion and questions.

- See website for link to sign up.

Piazza and Participation

We will use Piazza for class discussion and questions.

- See website for link to sign up.

You may earn up to 5% extra credit for participation.

- Asking good clarifying questions and answering questions during the lecture or on Piazza.
- Answering other students' on Piazza.
- Posting helpful/interesting links on Piazza, e.g., resources that cover class material, research articles related to the topics covered in class, etc.

Textbooks and Materials

I will post optional readings and references for each class. A lot of the content is covered in the following two textbooks which may be helpful for reference:

- *Probability and Computing*, by Mitzenmacher and Upfal
- *Randomized Algorithms*, by Motwani and Raghavan

Lecture notes will be posted before each class, and annotated notes posted after class. Recordings should be available via Echo360 – may take a couple of classes to work out any kinks and there may be occasional technical issues.

Homework

We will have 5 problem sets, which you may complete in **groups of up to 3 students**.

Homework

We will have 5 problem sets, which you may complete in **groups of up to 3 students**.

- We strongly encourage working in groups, as it will make completing the problem sets much easier/more educational.
- We strongly recommend that you do not simply split up the problem set and complete different parts independently.
- Collaboration with students outside your group is limited to discussion at a high level. You may not work through problems in detail or write up solutions together.
- See Piazza for a thread to help you organize groups.

Homework

We will have 5 problem sets, which you may complete in **groups of up to 3 students**.

- We strongly encourage working in groups, as it will make completing the problem sets much easier/more educational.
- We strongly recommend that you do not simply split up the problem set and complete different parts independently.
- Collaboration with students outside your group is limited to discussion at a high level. You may not work through problems in detail or write up solutions together.
- See Piazza for a thread to help you organize groups.

Problem set submissions will be via Gradescope.

- See website for a link to join and entry code.

Weekly Quizzes

I will release a multiple choice quiz in Moodle each Thursday after lecture, due the next Monday at 8pm.

Weekly Quizzes

I will release a multiple choice quiz in Moodle each Thursday after lecture, due the next Monday at 8pm.

- Designed as a check-in that you are following the material, and to help me make adjustments as needed.
- Will take around 15-30 minutes per week, open notes.
- Will also include free response check-in questions to get your feedback on how the course is going, what material from the past week you find most confusing, interesting, etc.

Grading

Grade Breakdown:

- Problem Sets (5 total): 40%, weighted equally.
- Weekly Quizzes: 10%, weighted equally, lowest score dropped.
- Midterm (March 18th, in class): 20%.
- Final OR Final Project: 30%.
- Extra Credit: Up to 5% for participation, and potentially more available on problem sets, for challenge questions asked in class, etc.

Final project

Optionally, instead of taking the final exam, you can complete a final project.

- Identify a topic of current research, formulate a research problem, and make an effort to tackle that problem.
- Submit a $\sim$ 10 page final report.
- Completed in groups of two – if you would like to work alone, please email the instructor to request permission.
- See details and project suggestions in Moodle. Deadlines are 3/12 for a 1 page proposal, 4/16 for a 2 page progress report, 5/10 for the final report.
- I encourage discussing ideas you have with me early in the semester to get some feedback.
- Much more work than if you just took the final, but valuable if you are interested in doing research in the area.

Academic Honesty

- A first violation cheating on a homework, quiz, or other assignment will result in a 0 on that assignment.
- A second violation, or a violation on an exam/project will result in failing the class.
- For fairness, I adhere very strictly to these policies.
- All students in a problem set group are responsible for violations, even those that they were not aware of. So make sure you actually work on the problems together and don't just split up the work.

Disability Services and Accommodations

UMass Amherst is committed to making reasonable, effective, and appropriate accommodations to meet the needs to students with disabilities.

- If you have a documented disability **on file with Disability Services**, you may be eligible for reasonable accommodations in this course.
- If your disability requires an accommodation, please email me by next Friday 2/8 so that we can make arrangements.

Disability Services and Accommodations

UMass Amherst is committed to making reasonable, effective, and appropriate accommodations to meet the needs to students with disabilities.

- If you have a documented disability on file with Disability Services, you may be eligible for reasonable accommodations in this course.
- If your disability requires an accommodation, please email me by next Friday 2/8 so that we can make arrangements.

I understand that people have different learning needs, home situations, etc. If something isn't working for you in the class, please reach out and let's try to work it out.

Questions?

Background on Randomized Algorithms

Types of Randomized Algorithms

Las-Vegas: Always correct, but the runtime is a random variable (if you get unlucky, the algorithm might be slow).

Monte-Carlo: Runtime is bounded deterministically, but the algorithm may be incorrect with small probability.

Complexity Theory

- $\boxed{\text{P}}$: Decidable by a deterministic polynomial time algorithm.
- Las Vegas ZPP: Decidable by a randomized algorithm which is always correct and runs in polynomial time in expectation.
- Monte carlo PP: Decidable by a polynomial time randomized algorithm that is correct with probability $> 1/2$ on both 'yes' and 'no' instances.
  $(\frac{1}{2} + \frac{1}{2^{2^n}}$
- $\boxed{\text{BPP}}$: Same as PP but must be correct with probability $\underline{\geq 2/3}$.
- RP: Decidable by a polynomial time randomized algorithm that is correct with probability 1 on 'no' instances, and $\underline{\geq 1/2 \text{ on}}$ 'yes' instances (one-sided error).

Complexity Theory

- P: Decidable by a deterministic polynomial time algorithm.
- ZPP: Decidable by a randomized algorithm which is always correct and runs in polynomial time in expectation.
- PP: Decidable by a polynomial time randomized algorithm that is correct with probability $> 1/2$ on both 'yes' and 'no' instances.
- BPP: Same as PP but must be correct with probability $\geq 2/3$.
- RP: Decidable by a polynomial time randomized algorithm that is correct with probability 1 on 'no' instances, and $\geq 1/2$ on 'yes' instances (one-sided error).

Think-Pair-Share 1: Why for BPP is the probability of success stated as 2/3, rather than say 1/4 or 9/10? Why for RP is it stated as 1/2 for ~~'no'~~ yes instances rather than say 1/4 or 2/3?

Think-Pair-Share 2: How do these complexity classes compare? Which is the biggest? Which is the smallest?

$PP \supseteq BPP$

Think-Pair-Share 1: Why for BPP is the probability of success stated as 2/3, rather than say 1/4 or 9/10? Why for RP is it stated as 1/2 for 'yes' instances rather than say 1/4 or 2/3?

For BPP any constant success prob. $> \frac{1}{2}$ should work.

└ boost success probability w/ repetition.

51% → run the algo 10000 times expect 5100 correct answer

- take majority

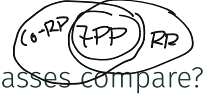
$\frac{1}{2} + \frac{1}{2^n}$

RP success 1/4

No instance: NO, NO, NO . . . NO

Yes instance: NO, NO, NO, YES, . . .

$\frac{1}{2^n}$



Think-Pair-Share 2: How do these complexity classes compare? Which is the biggest? Which is the smallest?

P

BPP

ZPP

RP

PP

take ZPP algo
turn it into
an RR

stop early.

if terminates
output answer
else output
"no"

$ZPP \subseteq BPP$

in?

$NP \subseteq$

$ZPP \subseteq RP \subseteq BPP \subseteq PP$

yes instance 100%
no instance 2/3

If $E[R] \leq T$

$Pr(R \geq 3T) \leq \frac{1}{3}$

2/3 time get right answer
1/3 time make guess

Complexity Theory

Major Open Question: $P \subseteq ZPP \subseteq RP \subseteq BPP \subseteq PP$ but does $P = BPP$?

$$\boxed{\underline{\text{co-RP}} \cap \underline{\text{RP}} = \text{ZPP}}$$

RP

NO INT	NO	NO	NO	NO . . .
YES INT	NO	NO	NO	<u>YES</u>

co-RP :

NO INT	YES	YES . . .	
YES INT	YES	YES	<u>NO</u>

Complexity Theory

Major Open Question: $P \subseteq ZPP \subseteq RP \subseteq BPP \subseteq PP$ but does $P = BPP$?

- Most think yes, that sufficiently strong pseudorandom number generators will eventually show that $P = BPP$.

Complexity Theory

Major Open Question: $\underbrace{P \underbrace{\subseteq ZPP \subseteq RP \subseteq}_{} BPP}_{} \subseteq PP$ but does $P = BPP$?

- Most think yes, that sufficiently strong pseudorandom number generators will eventually show that $\underline{P = BPP}$.

Are there natural problems in BPP **that we don't know are in** P**?**

Complexity Theory

Major Open Question: $P \subseteq ZPP \subseteq RP \subseteq BPP \subseteq PP$ but does $P = BPP$?

- Most think yes, that sufficiently strong pseudorandom number generators will eventually show that $P = BPP$.

Are there natural problems in BPP that we don't know are in P?

- For a long time primality testing was one such problem.
- Randomized algorithms for primality testing (Miller-Rabin '76, '80 and Solovay–Strassen '77) were very important. They made RSA encryption possible – central in the rise of randomized algorithms
- In 2002, Agrawal–Kayal–Saxena finally gave a polynomial time deterministic test. In practice, randomized tests are still used.

Complexity Theory

Major Open Question: $P \subseteq ZPP \subseteq RP \subseteq BPP \subseteq PP$ but does $P = BPP$?

- Most think yes, that sufficiently strong pseudorandom number generators will eventually show that $P = BPP$.

Are there natural problems in *BPP* that we don't know are in *P*?

- For a long time primality testing was one such problem.
- Randomized algorithms for primality testing (Miller-Rabin '76, '80 and Solovay–Strassen '77) were very important. They made RSA encryption possible – central in the rise of randomized algorithms
- In 2002, Agrawal–Kayal–Saxena finally gave a polynomial time deterministic test. In practice, randomized tests are still used.
- Currently, polynomial identity testing is probably the most important problem known to be in BPP but not P.