
Course Title

Probability for Computer Science

Course Description

This course introduces probability theory from a computational perspective. Students learn to model uncertainty, analyze randomized algorithms, reason about machine learning systems, and simulate stochastic processes. Emphasis is placed on algorithmic thinking, implementation, and real-world CS applications.

Target Audience

- CS majors (freshman–junior level)
- Students who completed:
    - Calculus I (basic derivatives + integrals)
    - Intro programming (Python or C++)

Learning Objectives

By the end of the course, students should be able to:

1. Model real-world uncertainty using probability spaces.
2. Analyze expected runtime of randomized algorithms.
3. Work with discrete and continuous random variables.
4. Use distributions common in CS (Bernoulli, Binomial, Geometric, Poisson, Gaussian).
5. Apply concentration bounds (Markov, Chebyshev, Chernoff).
6. Simulate probabilistic systems in code.
7. Understand probabilistic foundations of ML.
8. Implement Monte Carlo methods.
9. Reason about hashing, load balancing, and streaming via probability.
10. Build small probabilistic systems (e.g., spam filter, random graph simulator).

Course Structure

15 Weeks. 2 lectures + 1 lab per week.

Module Breakdown

Module 1: Foundations of Probability (Weeks 1–3)

Topics

- Sample spaces and events
- Axioms of probability
- Conditional probability
- Bayes' Theorem
- Independence
- Law of Total Probability

CS Applications

- Spam filtering via Naive Bayes
- Fault detection systems
- Probabilistic reasoning in distributed systems

Lab Ideas

- Implement a Naive Bayes classifier
- Simulate dice experiments and verify convergence

Module 2: Random Variables & Distributions (Weeks 4–6)

Topics

- Discrete random variables
- PMF, CDF
- Expectation and linearity of expectation

- Variance and standard deviation
- Bernoulli, Binomial, Geometric
- Poisson distribution

CS Applications

- Packet loss modeling
- Randomized quicksort analysis
- Retry algorithms and backoff strategies

Lab Ideas

- Simulate geometric retries in networking
- Empirically test quicksort expected runtime

Module 3: Continuous Probability (Weeks 7–8)

Topics

- Continuous random variables
- PDFs
- Normal distribution
- Exponential distribution
- Central Limit Theorem

CS Applications

- Latency modeling
- Gaussian noise in ML
- Queueing systems
- A/B testing

Lab Ideas

- Simulate CLT convergence
- Model server response times

Module 4: Randomized Algorithms (Weeks 9–10)

Topics

- Randomized vs deterministic algorithms
- Indicator random variables
- Hashing and universal hashing
- Balls and bins analysis
- Load balancing

CS Applications

- Hash tables
- Bloom filters
- Distributed systems load balancing

Lab Ideas

- Implement Bloom filter
- Simulate load distribution across servers

Module 5: Concentration Inequalities (Week 11)

Topics

- Markov’s inequality
- Chebyshev’s inequality
- Chernoff bounds
- High probability guarantees

CS Applications

- Streaming algorithms
- Random sampling
- Approximate counting
- ML confidence intervals

Lab Ideas

- Implement randomized streaming estimator
- Compare empirical error vs theoretical bounds

Module 6: Monte Carlo & Simulation (Week 12)

Topics

- Monte Carlo estimation
- Law of Large Numbers
- Random sampling techniques
- Variance reduction

CS Applications

- Reinforcement learning simulations
- Risk modeling
- Graphics rendering (ray tracing)

Lab Ideas

- Monte Carlo estimation of π
- Simulate reinforcement learning bandits

Module 7: Markov Chains & Stochastic Processes (Weeks 13–14)

Topics

- Markov chains
- Transition matrices
- Stationary distributions
- Random walks
- PageRank

CS Applications

- Google PageRank
- Recommendation systems
- MCMC in ML

Lab Ideas

- Implement PageRank
- Simulate random walk on graph

Module 8: Probability in Machine Learning (Week 15)

Topics

- Maximum likelihood estimation
- MAP estimation
- Overfitting and uncertainty
- Bias–variance tradeoff
- Probabilistic models overview

CS Applications

- Logistic regression
- Bayesian models
- Generative vs discriminative models

Lab Ideas

- Build simple logistic regression from scratch
- Compare deterministic vs probabilistic predictions

Assessments

Homework (Weekly)

- Mix of proofs + implementation
- 50% theoretical reasoning
- 50% coding

Midterm

- Conceptual + analytical problems
- Randomized algorithm analysis

Final Project (Very Important)

Students must build one of:

- A probabilistic spam filter
- A Monte Carlo simulation system
- A load-balancing simulator
- A Markov chain text generator
- A streaming algorithm with probabilistic guarantees

Graded on:

- Correctness
- Mathematical reasoning
- Code quality
- Interpretation of results

Suggested Tools

- Python (NumPy, matplotlib)
- Jupyter notebooks
- NetworkX (graphs)
- Optional: PyTorch for ML tie-ins

Textbook Options

1. “Probability and Computing” – Mitzenmacher & Upfal
2. “Introduction to Probability for Data Science” – Blitzstein
3. “Randomized Algorithms” – Motwani & Raghavan

Optional Advanced Topics (Honors Section)

- Martingales
- VC dimension
- PAC learning
- Information theory basics

- Entropy
- Mutual information
- Probabilistic graphical models

What Makes This CS-Focused Instead of Pure Math

- Every theorem connects to a system.
- Every concept is simulated.
- Students write code for probability.
- Proofs are used to analyze algorithms, not abstract measure theory.
- Heavy emphasis on “with high probability” reasoning.
