UC Berkeley
Department of Electrical Engineering and Computer Sciences

EECS 126: Probability and Random Processes

Final Study Guide

Spring 2021

This is a brief summary of the topics we covered that will be in scope for the final. However, the scope of the exam will encompass all of the homeworks, discussions, lab, and lecture material; if something is not in this document, it is not necessarily out of scope. Students are expected to understand topics in more depth than they are discussed here.

1 Probability Fundamentals

1. Kolmogorov's axioms: probabilities are nonnegative, probability of at least one possible outcome is 1, for disjoint events A, B, $\Pr(A \cup B) = \Pr(A) + \Pr(B)$
2. Conditional probability: $\Pr(A, B) = \Pr(A|B)\Pr(B)$
   (a) Law of total probability: for a partition $B_1, .., , B_n$ of Ω, $\Pr(A) = \sum_{i=1}^{n} \Pr(A|B_i)\Pr(B_i)$
   (b) Bayes' rule: $\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$
3. Independence implies uncorrelated; reverse not necessarily true
   (a) X, Y are independent if for all A, B: $\Pr(X \in A, Y \in B) = \Pr(X \in A)\Pr(Y \in B)$
   (b) Covariance: $\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$
   (c) Correlation: $\mathrm{Corr}(X, Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma(X)\sigma(Y)}$ ($-1 \leq \mathrm{Corr}(X, Y) \leq 1$ by Cauchy-Schwarz)

2 Common Tools

1. Inclusion-exclusion: $\Pr(\bigcup_{i=1}^{n} A_i) = \sum_{k=1}^{n}(-1)^{k+1} \sum_{1 \leq i_1 < \cdots < i_k \leq n} \Pr(A_{i_1} \cap \cdots \cap A_{i_k})$
2. Iterated expectation/tower rule: $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]]$
3. Law of total variance: $\mathrm{Var}(Y) = \mathbb{E}[\mathrm{Var}(Y|X)] + \mathrm{Var}(\mathbb{E}[Y|X])$
4. Convolution: we can solve for the pdf of the sum of independent random variables $Z = X + Y$ as $f_Z(z) = \int_{t=-\infty}^{\infty} f_X(z-t) f_Y(t) dt$ (or analogous summation for discrete)
5. Tail sum: $\mathbb{E}[X] = \sum_{x=0}^{\infty} P(X > x)$ for discrete, $\int_{x=0}^{\infty} P(X > x)$ for continuous
6. Order statistics: for i.i.d. $X_1, \ldots, X_n$, denote $X^{(i)}$ as the i-th smallest value
   (a) For continuous X_i, $f_{X^{(i)}}(y) = n\binom{n-1}{i-1} F_X(y)^{i-1}(1 - F_X(y))^{n-i} f_X(y)$

3 Problem Solving Techniques

1. Be able to use counting to calculate probabilities (combinations, stars and bars, etc.)
2. Be familiar with indicator variables and using them to calculate expectations/variance
3. Be comfortable with using symmetry to simplify calculations
4. Derived distributions: know how to get distributions for functions of random variables
5. Graphical density: reading a pdf from a graph and performing calculations with it

4 Common Distributions

1. Bernoulli: is 1 w.p. p and 0 otherwise
2. Binomial: sum of i.i.d. Bernoullis
3. Geometric/exponential: exhibit unique memoryless property
   (a) Min of exponentials is exponential with rate $\sum_{j=1}^{n} \lambda_j$; $\mathbb{P}(X_k = \min_i X_i) = \frac{\lambda_k}{\sum_{j=1}^{n} \lambda_j}$
4. Poisson(λ) is the limit of a binomial as $n \to \infty$ and $p \to 0$ and $np \to \lambda$
   (a) Poisson merging: for independent $X \sim \text{Pois}(\lambda), Y \sim \text{Pois}(\mu)$, $X + Y \sim \text{Pois}(\lambda + \mu)$
   (b) Poisson splitting: Poisson(λ) with arrivals dropped independently with probability p is distributed as Poisson(λp); the dropped arrivals form an independent Poisson($\lambda(1-p)$)
5. Gaussian: ubiquitous distribution commonly used for modeling noise
   (a) For independent $X \sim \mathcal{N}(\mu_1, \sigma_1^2), Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$, $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$

5 Moment Generating Functions

1. Moment generating functions are given by $M_X(t) = \mathbb{E}[e^{tX}]$; be able to recognize these for common distributions, and read off the parameters
2. The n-th moment of a random variable is $\mathbb{E}[X^n] = M_X^{(n)}(0) = \frac{d^n M_X}{ds^n}|_{s=0}$
3. For a linear combination $Z = aX + bY$ and X, Y independent, the MGF of Z is $M_Z(t) = M_X(at)M_Y(bt)$, which is often easier for convolutions than integrating

6 Bounds/Concentration Inequalities

1. Union bound: $\mathbb{P}(\bigcup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} \mathbb{P}(A_i)$
2. Markov's inequality: $\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$ for nonnegative random variable X and $a > 0$
3. Chebyshev's inequality: $\mathbb{P}(|X - \mathbb{E}[X]| \geq c) \leq \frac{\text{Var}(X)}{c^2}$ (apply Markov's to $|X - \mathbb{E}[X]|$)
4. We can combine the (nonnegative) MGF with Markov's to get Chernoff's inequality: $\mathbb{P}(X \geq a) = \mathbb{P}(e^{tX} \geq e^{ta}) \leq \frac{\mathbb{E}[e^{tX}]}{[e^{ta}]}$, and taking the min over t to get the best bound

7 Convergence

1. Almost sure convergence: $X_n \xrightarrow[n\to\infty]{\text{a.s.}} X$ if $\mathbb{P}(\lim_{n\to\infty} X_n = X) = 1$, i.e. the sequence $X^{(n)}$ deviates only a finite number of times from X
   (a) Strong Law of Large Numbers (empirical mean converges to true mean almost surely), convergence to stationary dist of irreducible aperiodic DTMC
   (b) Note a synonym for "almost surely" is "with probability one"
2. Convergence in probability: $X_n \xrightarrow[n\to\infty]{\text{i.p.}} X$ if $\lim_{n\to\infty} \mathbb{P}(|X_n - X| > \epsilon) = 0$, i.e. the probability that X_n deviates only from X goes to zero (but can still deviate infinitely)
   (a) Weak Law of Large Numbers (empirical mean converges to true mean in probability)
3. Convergence in distribution: $X_n \xrightarrow[n\to\infty]{\text{d}} X$ if for all x such that $\mathbb{P}(X = x) = 0$, we have $\mathbb{P}(X_n \leq x) \xrightarrow[n\to\infty]{} \mathbb{P}(X \leq x)$, i.e. X_n is modeled by the distribution X

(a) Central Limit Theorem (distribution of outcomes converges to a standard normal), Markov Chains (state distribution converges to stationary distribution)

4. We can use CLT, bounds to generate confidence intervals

8 Information Theory

1. We measure the "surprise" of a distribution with the entropy $H(X) = -\mathbb{E}[\log p(X)]$
   (a) Chain rule for entropy: $H(X, Y) = H(X) + H(Y|X)$
   (b) Mutual information: $I(X; Y) = H(X) - H(X|Y)$
2. Huffman encoding: compresses X to $H(X)$ bits
   (a) Source coding theorem: cannot compress X in less than $H(X)$ bits
3. We can send information through a channel up to the capacity $C = \max_{p_X} I(X; Y)$
4. Common channel examples:
   (a) Binary erasure channel: bit erased with probability p, has capacity $C = 1 - p$
   (b) Binary symmetric channel: bit swapped with probability p, has capacity $C = 1 - H(p)$

9 Discrete-Time Markov Chains

1. Markov chains satisfy the Markov property: $\mathbb{P}(X_n|X_{n-1}, X_{n-2}, ...) = \mathbb{P}(X_n|X_{n-1})$
2. Identify recurrence (positive, null), transience, irreducibility, periodicity, reversibility
3. Solving Markov chains: stationary distribution ($\pi = \pi P$), first step equations, detailed balance equations
4. Big theorem, stationary distribution, balance equations:
   (a) Detailed (a.k.a. local) balance equations hold if the Markov chain as a tree structure
   (b) Flow-in/flow-out holds for any cut, extends detailed balance equations
   (c) Stationary distribution exists for a class iff it is positive recurrent; if it exists, the stationary distribution for a communicating class is unique
   (d) If the whole chain is irreducible, then there is a unique stationary distribution
   (e) If whole chain is also aperiodic, then the chain converges a.s. to the stationary distribution for any initial distribution
5. Other useful properties of Markov chains:
   (a) The reciprocal of the stationary distribution is the expected time to return to a state given that you start from that state
   (b) For undirected graphs, $\pi(i) = \frac{\text{degree}(i)}{2E}$, where E is the number of edges in the graph
6. Be able to handle an infinite number of states if necessary (ex. queue)
7. MCMC: when a probability function is intractable, we can set up a MC and sample from the stationary distribution as a proxy for sampling from the original distribution

10 Poisson Processes

1. Understand what a Poisson process is, memorylessness, independence of non-overlapping intervals; the number of arrivals in an interval of length t is distributed as Poisson(λt)
2. Distribution of arrival times, relationship between Poisson and exponential
   (a) $T_k \sim \text{Erlang}(k, \lambda)$ is the distribution of sum of k independent exponentials with rate λ
   (b) Conditioned on n arrivals up to a certain time t ($N_t = n$), $T_1, ..., T_n$ are distributed according to the order statistics of n $U[0, t]$ random variables (e.g. $\mathbb{E}[T_{i+1} - T_i] = \frac{t}{n+1}$)
3. Poisson merging: the sum of independent Poisson processes with rates λ, μ is a new Poisson process with rate $\lambda + \mu$
4. Poisson splitting: for a Poisson process with rate λ, if we label each arrival 0/1 with probability p, the 0/1 arrivals as Poisson processes with rate $p\lambda, (1-p)\lambda$ (resp.)
5. RIP: from the perspective of a given point, the expected interarrival time is doubled. A Poisson process backwards is still a Poisson process.

11 Continuous-Time Markov Chains

1. Understand how to set up rate matrix, what it means to jump states, that the "holding time" is the min of exponentials, how to calculate transition probabilities
2. Understand detailed balance equations for continuous time
3. Identify recurrence (positive, null) and transience
4. Be able to solve CTMCs for stationary distribution ($\pi Q = 0$), expected hitting times
5. Jump/embedded chain: create a DTMC that models the "jumps" of a CTMC, i.e. the visitation order of the states, by considering the transition probabilities as the min of exponentials
   (a) Transition probability from k to j is $\mathbb{P}(k, j) = \frac{\lambda_{k,j}}{\sum_{i=1}^{n} \lambda_{k,i}}$
   (b) ex. modeling who will win a basketball game (first to 10 points), where teams score according to a Poisson distribution
   (c) crucially, no self loops, so does not take into account holding time.
6. Uniformization: create a DTMC that has the same stationary distribution of the CTMC, by relating the rates in terms of a fixed discrete rate
   (a) Choose a fixed rate λ (we frequently use the largest sum of outgoing rates of any state in the CTMC, but any greater value also works)
   (b) Transition probability from k to j (for $k \neq j$) is $\mathbb{P}(k, j) = \frac{\lambda_{k,j}}{\lambda}$
   (c) Transition probability from k to k (self-loop) is $\mathbb{P}(k, k) = 1 - \sum_{i=1, i\neq k}^{n} \mathbb{P}(k, i)$
   (d) Also can write transition matrix P in terms of rate matrix Q as $P = I + \frac{1}{\lambda} Q$
   (e) Has the same stationary distribution: $\pi P = \pi(\frac{1}{\lambda} Q + I) = \pi(0 + I) = \pi$

12 Erdos-Renyi Random Graphs

1. ER Graph $\mathcal{G}(n, p)$: random graph of n vertices where each edge is independently picked to be in the graph with probability p, no self-loops
2. Know how to do calculations for degree distribution, expectations, and other simple random quantities associated with the graph
3. Understand how tail bounds can help establish thresholds for random graphs (see Random Graphs note on website)

13 Estimation, MLE, MAP

1. The expected error of an estimator $\hat{y} = f(X)$ is $\mathbb{E}[(f(X) - y)^2]$
2. The bias of an estimator is $\mathbb{E}[f(X) - y]$ – when this is zero, we say the estimator is unbiased; note that minimizing bias is not the same as minimizing expected error
3. Maximum Likelihood Estimation (MLE): find parameters θ of model that maximize the likelihood $l(X|\theta)$
   (a) Most of the time, $l(X|\theta)$ has form $\prod_{i=1}^{n} f(x_i|\theta)$, so it is easier to equivalently maximize the log-likelihood, which will have form $\sum_{i=1}^{n} \log f(x_i|\theta)$
   (b) The MLE estimator can be biased, ex. German tank problem, finding variance of a Gaussian from samples
   (c) MLE is a special case of MAP, where the prior over θ is uniform
   (d) Most statistics/machine learning is MLE with a uniform prior on the dataset
4. Maximum a Posteriori Estimation (MAP): find parameters θ of model that maximize $l(X|\theta)f(\theta)$, where $f(\theta)$ is a prior probability distribution over θ
   (a) Adding a regularizer to a cost function typically corresponds to some form of MAP estimation, ex. $L(x) = \sum_{i=1}^{n}(y_i - wx_i)^2 + \lambda|w|$ corresponds to a Laplace prior
   (b) When performing statistics/machine learning on a dataset, MAP can correspond to some datapoints being more important than others

14 Hypothesis Testing

1. Neyman-Pearson hypothesis testing: a form of frequentist hypothesis testing, where we assume no prior over possible values for the parameter X we want to estimate
2. We suppose there are two outcomes, either $X = 0$ – the null hypothesis – or $X = 1$, the alternate hypothesis
3. Since we have no prior, there is no notion of the "most likely" outcome: we have to instead measure PFA, PCD, which we can measure since they assume that a given outcome is true
4. Probability of False Alarm (PFA): $P(\hat{X} = 1|X = 0)$
5. Probability of Correct Detection (PCD): $P(\hat{X} = 1|X = 1)$
6. Goal of N-P hypothesis testing: maximize PCD such that the PFA is less than β
7. ROC curve: maximizing PCD is equivalent to maximizing PFA subject to the PFA constraint β; area under the curve (AUC) is the probability a randomly chosen positive sample is ranked higher than a randomly chosen negative sample
8. In order to maximize PFA, we may need to add some probability γ of reporting that $\hat{X} = 0$ on the decision boundary
9. Intuitively, we seek to answer in the form of a p-value: given that $X = 0/X = 1$ is true, what is the probability we observed this data?

15 Vector Space of Random Variables

1. We consider a Hilbert space of random variables, where $\langle X, Y \rangle = \mathbb{E}[XY]$, with corresponding norm $\|X\|^2 = \mathbb{E}[X^2]$
2. Linear Least Squares Estimation (LLSE): we have three vectors $1, X, Y$ – we want to find Y given the data X, it terms of a linear combination of X and 1
   (a) This corresponds to finding the projection of Y onto both $\tilde{X}$ and 1, where $\tilde{X}$ is the transformed X such that $\langle \tilde{X}, 1 \rangle = 0$
   (b) $L[Y|X] = \mathbb{E}[Y] + \frac{\text{cov}(X,Y)}{\text{var}(X)}(X - \mathbb{E}[X])$
   (c) If the noise is Gaussian, the LLSE is also the MMSE
3. Minimum Mean Square Estimation (MMSE): find the best function ϕ to minimize the expected squared error $\mathbb{E}(Y - \phi(X))^2]$
   (a) Here, we want $\langle Y - \phi(X), f(X) \rangle = 0$ for all other f
   (b) In particular, the MMSE estimator is $\mathbb{E}[Y|X]$
4. Both the LLSE and MMSE are unbiased, as $\mathbb{E}[X - \mathbb{E}[X]] = \mathbb{E}[\mathbb{E}[Y|X] - \mathbb{E}[Y]] = 0$

16 Kalman Filtering

1. Jointly Gaussian: two random variables X, Y are said to be jointly Gaussian if (X, Y) is Gaussian; the linear combination of Gaussians is Gaussian
   (a) Also written as: a jointly Gaussian vector Y can be written as $Y = AZ + \mu$, where Z is a vector of standard iid Gaussians
   (b) Uncorrelated jointly Gaussian RVs are independent
2. Setup: we have some hidden, time varying parameters X_i that output noisy observations Y_i, where both sources of noise are Gaussian
3. Kalman filtering: given observations $Y_1, ..., Y_n$, a recursive set of update rules to estimate the underlying parameter X_n: $\mathbb{E}[X_n|Y_1, Y_2, ...Y_n]$
4. Kalman smoothing: given observations $Y_1, ..., Y_n$, estimate the past underlying parameter X_i, for some $i < n$: $\mathbb{E}[X_i|Y_1, Y_2, ..., Y_n]$

17 Labs

All material from labs is in scope, but in particular you should be comfortable with the following ideas:

1. Fountain codes: familiar with the general setup and decoding scheme: problem is to design some distribution over the number of chunks in each packet
2. Matrix sketching: we want to compute $A^T B$ so we "sketch" each matrix as SA and SB for some "fat" $d \times n$ random matrix S such that $S^T S \approx I_n$.
3. The general idea of sampling from a distribution with an inhibitively large sample space by simulating a random walk on a Markov chain with the correct stationary distribution. Specifically, be familiar with the MH algorithm idea of proposing a next state and accepting with some probability.
4. Perlin Noise can be used to generate correlated noise which is useful for applications such as word generation
5. Guess My Word: you should be familiar with the idea of binary search and using Huffman encodings to "ask" as few questions as possible