Probability for Computer Scientists

Stanford, California

Contents

Acknowledgments v

1 *Counting* 1

1.1 *Sum Rule* 1

1.2 *Product Rule* 2

1.3 *The Inclusion-Exclusion Principle* 3

1.4 *Double Counting and Constraints* 3

1.5 *General Principle of Counting* 4

1.6 *Floor and Ceiling* 5

1.7 *The Pigeonhole Principle* 5

1.8 *Bibliography* 6

2 *Combinatorics* 7

2.1 *Permutations* 7

2.2 *Permutations of Indistinct Objects* 10

2.3 *Combinations* 11

2.4 *Selecting Multiple Groups of Objects* 13

2.5 *Bucketing/Group Assignment* 13

3 Probability 17
3.1 Event Spaces and Sample Spaces 17
3.2 Probability 17
3.3 Axioms of Probability 18
3.4 Provable Identities of Probability 19
3.5 Equally Likely Outcomes 20

4 Conditional Probability 25
4.1 Conditional Probability 25
4.2 Law of Total Probability 26
4.3 Bayes' Theorem 27
4.4 Conditional Paradigm 28

5 Independence 31
5.1 Independence 31
5.2 Conditional Independence 34

6 Random Variables 37
6.1 Probability Mass Function 38
6.2 Expectation 39
6.3 Properties of Expectation 39
6.4 Variance 41

7 Bernoulli and Binomial Random Variables 45
7.1 Bernoulli Random Variable 45
7.2 Binomial Random Variable 46

8 Poisson and Other Discrete Distributions 49

8.1 Binomial in the Limit 49

8.2 Poisson Random Variable 50

8.3 Geometric Distribution 50

8.4 Negative Binomial Distribution 53

9 Continuous Distributions 55

9.1 Probability Density Functions 55

9.2 Cumulative Distribution Function 56

9.3 Expectation and Variance 57

9.4 Uniform Random Variable 60

9.5 Exponential Random Variable 61

10 Normal Distribution 63

10.1 Normal Random Variable 63

10.2 Linear Transform 64

10.3 Projection to Standard Normal 64

10.4 Binomial Approximation 67

11 Joint Distributions 69

11.1 Joint Distributions 69

11.2 Discrete Case 69

11.3 Multinomial Distribution 70

12 Independent Random Variables 73

12.1 Independence with Multiple Discrete Random Variables 73

12.2 Symmetry of Independence 73

12.3 Sums of Independent Random Variables 73

12.4 Convolution: Sum of Independent Random Variables 75

13 *Statistics of Multiple Random Variables* 77

13.1 *Expectation with Multiple Random Variables* 77

13.2 *Expectation of Binomial* 78

13.3 *Expectation of Negative Binomial* 78

13.4 *Expectations of Products Lemma* 79

13.5 *Covariance and Correlation* 81

13.6 *Properties of Covariance* 82

13.7 *Correlation* 83

14 *Conditional Expectation* 85

14.1 *Conditional Distributions* 85

14.2 *Conditional Expectation* 85

14.3 *Properties of Conditional Expectation* 86

14.4 *Law of Total Expectation* 86

15 *Inference* 91

15.1 *Bayesian Networks* 91

15.2 *General Inference via Sampling the Joint Distribution* 94

15.3 *General Inference when Conditioning on Rare Events* 96

16 *Continuous Joint Distributions* 99

16.1 *Independence with Multiple RVs (Continuous Case)* 100

16.2 *Joint CDFs* 100

16.3 *Bivariate Normal Distribution* 101

17 *Continuous Joint Distributions II* 103

17.1 *Convolution: Sum of Independent Random Variables* 103

17.2 *Independent Normals* 103
17.3 *General Independent Case* 103
17.4 *Conditional Distributions (Continuous Case)* 104
17.5 *Conditional Expectation (Continuous Case)* 104

18 *Central Limit Theorem* 109
18.1 *The Theory* 109

19 *Samples and the Bootstrap* 113
19.1 *Estimating Mean and Variance from samples* 113
19.2 *Standard Error* 114
19.3 *Bootstrap* 115

20 *Maximum Likelihood Estimation* 117
20.1 *Parameters* 117
20.2 *Maximum Likelihood* 118

21 *The Beta Distribution* 123
21.1 *Mixing Discrete and Continuous* 123
21.2 *Estimating Probabilities* 123
21.3 *Beta Distribution* 124

22 *Maximum A Posteriori* 127
22.1 *Maximum A Posteriori Estimation* 127

23 *Naïve Bayes* 133
23.1 *Machine Learning: Classification* 133
23.2 *Naïve Bayes algorithm* 134
23.3 *Theory* 135

24 *Linear Regression and Gradient Ascent* 139

24.1 *Regression* 139

24.2 *Gradient Ascent Optimization* 139

24.3 *Linear Regression* 140

25 *Logistic Regression* 145

25.1 *Logistic Regression Overview* 145

25.2 *Parameter Estimation using Gradient Ascent Optimization* 148

25.3 *Derivations* 148

A *Review* 151

B *Calculation Reference* 153

B.1 *Summation Identities* 153

B.2 *Growth Rates of Summations* 155

B.3 *Identities of Products* 155

B.4 *Calculus Review* 156

B.5 *Computing Permutations and Combinations* 157

Problem Set 1, Combinatorics 159

Problem Set 2, Probability 173

Problem Set 3, Random Variables 185

Problem Set 4, Distributions 199

Problem Set 5, Sampling 209

Quiz 1 221

References 227

Acknowledgments

This work is taken from the lecture notes for the course *Probability for Computer Scientists* at Stanford University, CS 109 (`cs109.stanford.edu`). The contributors to the content of this work are Chris Piech, Mehran Sahami, and Lisa Yan—this collection is simply a typesetting of existing lecture notes with minor modifications/additions. We would like to thank the original authors for their contribution. In addition, we wish to thank Mykel Kochenderfer and Tim Wheeler for their contribution to the Tufte-Algorithms LaTeX template, based off of *Algorithms for Optimization*.[1]

Robert J. Moss
Stanford, Calif.
July 29, 2020

[1] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization*. MIT Press, 2019.

Ancillary material is available on the template's webpage:
`https://github.com/sisl/textbook_template`

1 Counting

1.1 Sum Rule

From notes by Mehran Sahami, Chris Piech, and Lisa Yan.

Sum Rule of Counting: If the outcome of an experiment can either be one of m outcomes or one of n outcomes, where none of the outcomes in the set of m outcomes is the same as any of the outcomes in the set of n outcomes, then there are $m + n$ possible outcomes of the experiment.

Rewritten using set notation, the *Sum Rule* states that if the outcomes of an experiment can either be drawn from set A or set B, where $|A| = m$ and $|B| = n$, and $A \cap B = \emptyset$, then the number of outcomes of the experiment is $|A| + |B| = m + n$.

Example 1.1. *Sum Rule* example counting server requests.

Problem: You are running an online social networking application which has its distributed servers housed in two different data centers, one in San Fransisco and the other in Boston. The San Fransisco data center has 100 servers in it and the Boston data center has 50 servers in it. If a server request is sent to the application, how large is the set of servers it may get routed to?

Solution: Since the request can be sent to either of the two data centers and none of the machines in either data center are the same, the *Sum Rule of Counting* applies. Using this rule, we know that the request could potentially be routed to any of the $100 + 50 = 150$ servers.

1.2 *Product Rule*

Product Rule of Counting: If an experiment has two parts, where the first part can result in one of m outcomes and the second part can result in one of n outcomes regardless of the outcome of the first part, then the total number of outcomes for the experiment is mn.

Rewritten using set notation, the *Product Rule* states that if an experiment with two parts has an outcome from set A in the first part, where $|A| = m$, and an outcome from set B in the second part (regardless of the outcome of the first part), where $|B| = n$, then the total number of outcomes of the experiment is $|A||B| = mn$.

Note that the *Product Rule for Counting* is very similar to ''the basic principle of counting'' given in the Ross textbook.[1]''

Example 1.2. *Product Rule* example with two 6-sided dice.

Problem: Two 6-sided dice, with faces numbered 1 through 6, are rolled. How many possible outcomes of the roll are there?

Solution: Note that we are not concerned with the total value of the two dice, but rather the set of all explicit outcomes of the rolls. Since the first die can come up with 6 possible values and the second die similarly can have 6 possible values (regardless of what appeared on the first die), the total number of potential outcomes is $36 = 6 \cdot 6$. These possible outcomes are explicitly listed below as a series of pairs, denoting the values rolled on the pair of dice:

''die'' is the singular form of the word ''dice'' (which is the plural form).

$$\begin{array}{cccccc}
(1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (1,6) \\
(2,1) & (2,2) & (2,3) & (2,4) & (2,5) & (2,6) \\
(3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (3,6) \\
(4,1) & (4,2) & (4,3) & (4,4) & (4,5) & (4,6) \\
(5,1) & (5,2) & (5,3) & (5,4) & (5,5) & (5,6) \\
(6,1) & (6,2) & (6,3) & (6,4) & (6,5) & (6,6)
\end{array}$$

[1] S. M. Ross et al., *A First Course in Probability*. 2006, vol. 7.

Problem: Consider a hash table with 100 buckets. Two arbitrary strings are independently hashed and added to the table. How many possible ways are there for the strings to be stored in the table?

Solution: Each string can be hashed to one of 100 buckets. Since the results of hashing the first string do not impact the hash of the second, there are $100 \cdot 100 = 10{,}000$ ways that the two strings may be stored in the hash table.

Example 1.3. *Product Rule* example with string hashing.

1.3 *The Inclusion-Exclusion Principle*

Inclusion-Exclusion Principle: If the outcome of an experiment can either be drawn from set A or set B, and sets A and B may potentially overlap (i.e. it is not guaranteed that $A \cap B = \emptyset$), then the number of outcomes of the experiment is $|A \cup B| = |A| + |B| - |A \cap B|$.

Note that the *Inclusion-Exclusion Principle* generalizes the *Sum Rule of Counting* for arbitrary sets A and B. In the case where $A \cap B = \emptyset$, the *Inclusion-Exclusion Principle* gives the same result as the *Sum Rule of Counting* since $|\emptyset| = 0$.

Problem: An 8-bit string (one byte) is sent over a network. The valid set of strings recognized by the receiver must either start with `01` or end with `10`. How many such strings are there?

Solution: The potential bit strings that match the receiver's criteria can either be 64 strings that start with `01` (since the last 6 bits are unspecified). Of course, these two sets overlap, (since the middle 4 bits can be arbitrary). Casting this description into corresponding set notation, we have: $|A| = 64$, $|B| = 64$, and $|A \cap B| = 16$, so by the *Inclusion-Exclusion Principle*, there are $64 + 64 - 16 = 112$ strings that match the specified receiver's criteria.

Example 1.4. *Inclusion-Exclusion Principle* example with bit strings.

1.4 *Double Counting and Constraints*

There are many reasons for having counted some elements more than once (aka ''double counted''). One common case, is that there is a constraint in the problem that you must contend with. It goes without saying that if you over-count, then you have to subtract off the number of elements that were double counted. If you did something along the lines of: count every element some multiple, then you can divide your total number of elements by that multiple to get the correct final answer.

1.5 General Principle of Counting

General Principle of Counting: If an experiment has r parts such that part i has n_i outcomes for all $i = 1, \dots, r$, then the total number of outcomes for the experiment is:

$$\prod_{i=1}^{r} n_i = n_1 \times n_2 \times \cdots \times n_r \tag{1.1}$$

In the same way that the *Inclusion-Exclusion Principle* generalizes the *Sum Rule of Counting*, our next rule, the *General Principle of Counting* (also commonly known as the *Fundamental Principle of Counting*) generalizes the *Product Rule of Counting*.

Problem: California license plates prior to 1982 had only 6-place license plates, where the first three places were uppercase letters A-Z, and the last three places were numeric 0-9. How many such 6-place license plates were possible pre-1982?

Solution: We can treat each of the positions of the license plate as a separate part of an overall six part experiment. That is, the first three parts of the experiment each have 26 outcomes, corresponding to the letters A–Z, and the last three parts of the experiment each have 10 outcomes, corresponding to the digits 0–9. By the *General Principle of Counting*, we have $26 \times 26 \times 26 \times 10 \times 10 \times 10 = 17{,}576{,}000$ possible license plates. Interestingly enough the current population of California is 39.5 million residents as of 2017, so this would not nearly be enough license plates such that each person can own one vehicle. Fortunately, in 1982, California changed to 7-place license plates by prepending a numeric digit, resulting in $10 \times 26 \times 26 \times 26 \times 10 \times 10 \times 10 = 175{,}760{,}000$ possible 7-place license plates. This is enough for each resident in California to own approximately 4.5 vehicles.

Example 1.5. *General Principle of Counting* example with California license plates.

1.6 *Floor and Ceiling*

Floor and *ceiling* are two handy functions that we give below just for reference. Besides, their names sound so much neater than "rounding down" and "rounding up", and they are well-defined on negative numbers too.

Floor function: The floor function assigns to the real number x the largest integer that is less than or equal to x. The floor function applied to x is denoted $\lfloor x \rfloor$.

Ceiling function: The ceiling function assigns to the real number x the smallest integer that is greater than or equal to x. The ceiling function applied to x is denoted $\lceil x \rceil$.

Examples of the floor and ceiling functions operating on the same numbers:

$$\begin{array}{rllll} \text{floor:} & \lfloor 1/2 \rfloor = 0 & \lfloor -1/2 \rfloor = -1 & \lfloor 2.9 \rfloor = 2 & \lfloor 8.0 \rfloor = 8 \\ \text{ceiling:} & \lceil 1/2 \rceil = 1 & \lceil -1/2 \rceil = 0 & \lceil 2.9 \rceil = 3 & \lceil 8.0 \rceil = 8 \end{array}$$

1.7 *The Pigeonhole Principle*

Basic Pigeonhole Principle: For positive integers m and n, if m objects are placed in n buckets, where $m > n$, then at least one bucket must contain at least two objects.

General Pigeonhole Principle: In a more general form, this principle can be stated as the following. For positive integers m and n, if m objects are placed in n buckets, then at least one bucket must contain at least $\lceil m/n \rceil$ objects.

Note that the generalized form does not require the constraint that $m > n$, since in the case where $m \leq n$, we have $\lceil m/n \rceil = 1$, and it trivially holds that at least one bucket will contain at least one object.

Example 1.6. *General Pigeonhole Principle* example with hash tables.

Problem: Consider a hash table with 100 buckets. 950 strings are hashed and added to the table.

a. Is it possible that a bucket in the table has no entries?

b. Is it guaranteed that at least one bucket in the table has at least two entries?

c. Is it guaranteed that at least one bucket in the table has at least 10 entries?

d. Is it guaranteed that at least one bucket in the table has at least 11 entries?

Solution:

a. Yes. As one example, it is possible (albeit very improbable) that all 950 strings get hashed to the same bucket (say bucket 0). In this case bucket 1 would have no entries.

b. Yes. Since, 950 objects are placed in 100 buckets and $950 > 100$, by the *Basic Pigeonhole Principle*, it follows that at least one bucket must contain at least two entries.

c. Yes. Since, 950 objects are placed in 100 buckets and $\lceil 950/100 \rceil = \lceil 9.5 \rceil = 10$, by the *General Pigeonhole Principle*, it follows that at least one bucket must contain at least 10 entries.

d. No. As one example, consider the case where the first 50 bucket each contain 10 entries and the second 50 buckets each contain 9 entries. This accounts for all 950 entries $(50 \cdot 10 + 50 \cdot 9 = 950)$, but there is no bucket that contains 11 entries in the hash table.

1.8 *Bibliography*

For additional information on counting, you can consult a good discrete mathematics or probability textbook. Some of the discussion above is based on the treatment in *Discrete Mathematics and its Applications*.[2]

[2] K. Rosen, *Discrete Mathematics and Its Applications*. 2007, vol. 6.

2 Combinatorics

2.1 Permutations

From notes by Mehran Sahami, Chris Piech, and Lisa Yan.

Permutation Rule: A permutation is an ordered arrangement of n distinct objects. Those n objects can be permuted in $n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot 2 \cdot 1 = n!$ ways.

This changes slightly if you are permuting a subset of distinct objects, or if some of your objects are indistinct. We will handle those cases.

Problem, Part A: iPhones used to have 4-digit passcodes. Suppose there are 4 smudges over 4 digits on the screen. How many distinct passcodes are possible?

Solution: Since the order of digits in the code is important, we should use permutations. And since there are exactly four smudges we know that each number is distinct. This, we can plug in the permutation formula: $4! = 24$.

Example 2.1. *Permutation* example for iPhone passcode attempts with $n = 4$ fingerprint smudges.

Problem, Part B: What if there are 3 smudges over 3 digits on the screen?

Solution: One of 3 digits is repeated, but we don't know which one. We can solve this by making three cases, one for each digit that could be repeated (each with the same number of permutations). Let A, B, and C represent the 3 digits, with C repeated twice. We can initially pretend the two C's are distinct. Then each case will have 4! permutations:

$$A \; B \; C_1 \; C_2$$

However, then we need to eliminate the double-counting of the permutations of the identical digits (one A, one B, and two C's):

$$\frac{4!}{2! \cdot 1! \cdot 1!}$$

Adding up the three cases for the different repeated digits gives:

$$3 \cdot \frac{4!}{2! \cdot 1! \cdot 1!} = 3 \cdot 12 = 36$$

Example 2.2. *Permutation* example for iPhone passcode attempts with $n = 3$ fingerprint smudges.

Problem, Part C: What if there are 2 smudges over 2 digits on the screen?

Solution: There are two possibilities: 2 digits used twice each, or 1 digit used 3 times, and the other digit used once.

$$\frac{4!}{2! \cdot 2!} + 2 \cdot \frac{4!}{3! \cdot 1!} = 6 + (2 \cdot 4) = 6 + 8 = 14$$

Example 2.3. *Permutation* example for iPhone passcode attempts with $n = 2$ fingerprint smudges.

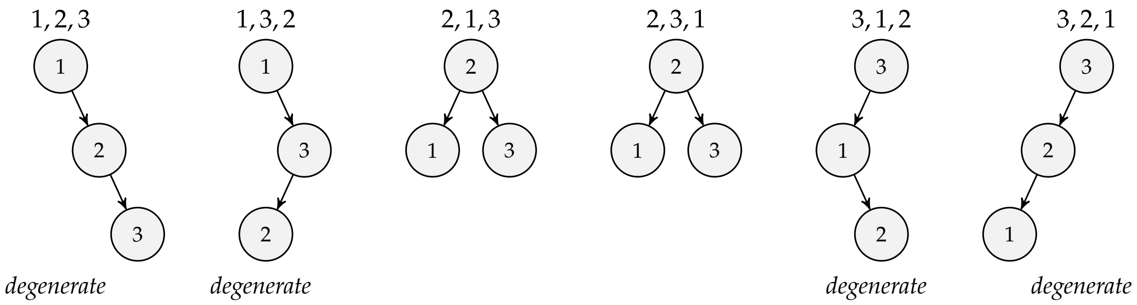


Figure 2.7. *Permutations* of a binary search tree for $n = 3$ integers, showing degenerate trees.

Recall the definition of a *binary search tree* (BST), which is a binary tree that satisfies the following three properties for *every* node n in the tree:

1. n's value is greater than all the values in its left subtree.
2. n's value is less than all the values in its right subtree.
3. both n's left and right subtrees are binary search trees.

Problem: How many possible binary search trees are there which contain the three values, 1, 2, and 3, and have a degenerate structure (i.e. each node in the BST has at most one child)?

Solution: We start by considering the fact that the three values in the BST ($1, 2$, and 3) may have been inserted in any of $3! = 6$ orderings (permutations). For each of the $3!$ ways the values could have been ordered when being inserted into the BST, we can determine what the resulting structure would be and determine which of them are degenerate. We consider each possible ordering of the three values and the resulting BST structure is shown in figure 2.7. We see that there are 4 degenerate BSTs here (the first two and last two).

Example 2.4. *Permutation* example with degenerate binary search trees.

2.2 *Permutations of Indistinct Objects*

Permutations of Indistinct Objects: Generally when there are n objects and

n_1 are the same (indistinguishable),

n_2 are the same,

...

and n_r are the same,

then there are $\frac{n!}{n_1!n_2!\dots n_r!}$ distinct permutations of the objects.[1]

Example 2.5. *Permutation* problem with bit strings.

Problem: How many distinct bit strings can be formed from three `0`'s and two `1`'s?

Solution: 5 total digits would give 5! permutations. But that is assuming the `0`'s and `1`'s are distinguishable (to make that explicit, let's give each one a subscript). Here is a subset of the permutations.

$$\begin{array}{ccccc}
0_1 & 1_1 & 1_2 & 0_2 & 0_3 \\
0_1 & 1_1 & 1_2 & 0_3 & 0_2 \\
0_2 & 1_1 & 1_2 & 0_1 & 0_3 \\
0_2 & 1_1 & 1_2 & 0_3 & 0_1 \\
0_3 & 1_1 & 1_2 & 0_1 & 0_2 \\
0_3 & 1_1 & 1_2 & 0_2 & 0_1
\end{array}$$

If identical digits are indistinguishable, then all the listed permutations are the same. For any given permutation, there are 3! ways of rearranging the `0`'s and 2! ways of rearranging the `1`'s (resulting in indistinguishable strings). We have over-counted. Using the formula for permutations of indistinct objects, we can correct for the over-counting:

$$\text{total} = \frac{5!}{3! \cdot 2!} = \frac{160}{6 \cdot 2} = \frac{120}{12} = 10$$

[1] When n objects are *distinct*, there are $r = n$ groups each with $n_i = 1$, thus this formula becomes $n!$ from section 2.1.

2.3 *Combinations*

Permutations account for ordering, where as *combinations* do not.

Binomial Coefficient: A combination is an unordered selection of r objects from a set of n objects. If all objects are distinct, then the number of ways of making the selection is:

$$\frac{n!}{r!(n-r)!} = \binom{n}{r} \tag{2.1}$$

The term $\binom{n}{r}$ is define as a *binomial coefficient* and is often read as "n choose r".

Consider this general way to produce combinations: To select r unordered objects from a set of n objects, e.g., "7 choose 3",

1. First consider permutations of all n objects. There are $n!$ ways to do that.

2. Then select the first r in the permutation. There is one way to do that.

3. Note that the order of r selected objects is irrelevant. There are $r!$ ways to permute them. The selection remains unchanged.

4. Note that the order of $(n-r)$ unselected objects is irrelevant. There are $(n-r)!$ ways to permute them. The selection remains unchanged.

$$\text{total} = \frac{n!}{r!(n-r)!} = \binom{n}{r} = \binom{n}{n-r} \qquad \text{e.g., } \frac{7!}{3!4!} = 35$$

which is the combinations formula.

A useful recursive identity for combinations is as follows:

$$\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}, \quad 0 \leq r \leq n \tag{2.2}$$

This identity can be proved via a combinatorial argument. When we select a group of size r from n distinct objects, then any particular object (say, object 1) will either be part of that group or not part of that group. We can then define sets A and B, where A is the number of ways of selecting a group that contains object 1, and B is the number of ways of selecting a group that does not contain object 1. For set A, if we decide to include object 1, then we must select $r-1$ of the remaining $n-1$ objects (since the membership of object 1 in our selection is already decided), or $1 \times \binom{n-1}{r-1}$. For set B, if we decide to exclude object 1, then we only have $n-1$ possible objects to select from to create a group of size r, or $n-1r$. These sets are mutually exclusive, and therefore by the *Sum Rule of Counting* the total number of possibilities are as above.

Problem: In the Hunger Games, how many ways are there of choosing 2 villagers from district 12, which has a population of 8,000?

Solution: This is a straightforward combinations problem.

$$\binom{8000}{2} = 31{,}996{,}000$$

Example 2.6. *Combination* problem involving the Hunger Games.

Problem, Part A: How many ways are there to select 3 books from a set of 6?

Solution: If each of the books are distinct, then this is another straight forward combination problem.

$$\binom{6}{3} = \frac{6!}{3!3!} = 20$$

Example 2.7. *Combination* problem choosing $r = 3$ books from $n = 6$.

Problem, Part B: How many ways are there to select 3 books if there are two books that should not both be chosen together? For example, if you are choosing 3 out of 6 probability books, don't choose both the 8th and 9th edition of the Ross textbook.

Solution: This problem is easier to solve if we split it up into cases. Consider the following three different cases:

Case 1: Select the 8th Ed. and 2 other non-9th Ed.: There are $\binom{4}{2}$ ways.

Case 2: Select the 9th Ed. and 2 other non-8th Ed.: There are $\binom{4}{2}$ ways.

Case 3: Select 3 from the books that are neither the 8th nor the 9th edition: There are $\binom{4}{3}$ ways.

Using our old friend the *Sum Rule of Counting*, we can add the cases:

$$\text{total} = 2 \cdot \binom{4}{2} + \binom{4}{3} = 16$$

Example 2.8. *Combination* problem choosing $r = 3$ books from $n = 6$ with restrictions.

Alternatively, we could have calculated all the ways of selecting 3 books from 6, and then subtract the "forbidden" ones (i.e., the selections that break the constraint). Chris Piech calls this the *Forbidden City method*.

Forbidden Case: Select 8th edition and 9th edition and 1 other book. There are $\binom{4}{1}$ ways of doing so (which equals 4).

$$\text{total} = \text{all possibilities} - \text{forbidden} = \binom{6}{3} - \binom{4}{1} = 20 - 4 = 16$$

Two different ways to get the same right answer!

Example 2.9. *Forbidden City method* of solving the *combination* problem choosing $r = 3$ books from $n = 6$ with restrictions.

2.4 *Selecting Multiple Groups of Objects*

Multinomial Coefficient: If n objects are distinct, then the number of ways to select r groups of objects, such that group i has size n_i and $\sum_{i=1}^{r} n_i = n$ is:

$$\frac{n!}{n_1!n_2!\cdots n_r!} = \binom{n}{n_1, n_2, \ldots, n_r} \tag{2.3}$$

where $\binom{n}{n_1,n_2,\ldots,n_r}$ is defined as a *multinomial coefficient*.

This situation is a generalization of the combination, where $\binom{n}{r}$ is defined as a binomial coefficient. One way to see this is that the task of selecting r unordered objects from a set of n distinct objects is analogous to the task of separating n objects into two groups 1 and 2, with respective element counts $n_1 = r$ and $n_2 = n - r$. Therefore it is true that the binomial coefficient $\binom{n}{r} = \binom{n}{r,n-r}$, where the latter is the multinomial coefficient.

2.5 *Bucketing/Group Assignment*

You have probably heard about the dreaded "balls and urns" probability examples. What are those all about? They are for counting the many different ways that we can think of stuffing elements into containers. (It turns out that Jacob Bernoulli was into voting and ancient Rome. And in ancient Rome they used urns for ballot boxes.) This "bucketing" or "group assignment" process is a useful metaphor for many counting problems. Note that this bucketing problem is different from the previous combinations problem. In combinations, we have n distinct

Example 2.10. *Multinomial coefficient* used to solve a server to datacenter allocation problem.

Problem: Company Camazon has 13 new servers that they would like to assign to 3 datacenters, where Datacenter A, B, and C have 6, 4, and 3 empty server racks, respectively. How many different divisions of the servers are possible?

Solution: This is a straight forward application of our multinomial coefficient representation. Setting $n_1 = 6, n_2 = 4, n_3 = 3$, $\binom{13}{6,4,3} = 60{,}060$. Another way to do this problem would be from first principles of combinations as a multi-part experiment. We first select the 6 servers to be assigned to Datacenter A, in $\binom{13}{6}$ ways. Now out of the 7 servers remaining, we select the 4 servers to be assigned to Datacenter B, in $\binom{7}{4}$ ways. Finally, we select the 3 servers out of the remaining 3 servers, in $\binom{3}{3}$ ways. By the *Product Rule of Counting*, the total number of ways to assign all servers would be:

$$\binom{13}{6}\binom{7}{4}\binom{3}{3} = \frac{13!}{6!4!3!} = 60{,}060$$

(distinguishable) objects to put in r distinct groups, and we are fixing the number of distinct objects in group i to be n_i (where $\sum_{i=1}^{r} n_i = n$) for every outcome that we count. By contrast, in the bucketing problem we still have n objects to put in r distinct groups, but (1) our objects can be distinct or indistinct, and (2) for each outcome we can vary the number of objects in each distinct group i.

Example 2.11. *Bucketing* used to solve a hash table problem.

Problem: Say you want to put n distinguishable balls into r urns. (No! Wait! Don't say that!) Okay, fine. No urns. Say we are going to put n strings into r buckets of a hash table where all outcomes are equally likely. How many possible ways are there of doing this?

Solution: You can think of this as n independent experiments each with r outcomes. Using the *General Principle of Counting*, this comes out to r^n.

2.5.1 *Divider Method*

While the previous example allowed us to put n distinguishable objects into r distinct groups, the more interesting problem is to work with n indistinguishable objects. This task has a direct analogy to the number of ways to solve the following positive integer equation:

$$x_1 + x_2 + \ldots + x_r = n, \text{ where } x_i \geq 0 \text{ for all } i = 1, \ldots, r \tag{2.4}$$

Divider Method: Suppose you want to place n indistinguishable items into r containers. The *divider method* works by imagining that you are going to solve this problem by sorting two types of objects, your n original elements and $(r-1)$ dividers. Thus, you are permuting $n+r-1$ objects, n of which are same (your elements) and $r-1$ of which are same (the dividers). Thus the total number of outcomes is:

$$\frac{(n+r-1)!}{n!(r-1)!} = \binom{n+r-1}{n} = \binom{n+r-1}{r-1} \tag{2.5}$$

Example 2.12. *Divider Method* used to solve an investment problem.

Problem, Part A: Say you are a startup incubator and you have $10 million to invest in 4 companies (in $1 million increments). How many ways can you allocate this money?

Solution: This is just like putting 10 balls into 4 urns. Using the *Divider Method* we get:

$$\text{total ways} = \binom{10+4-1}{10} = \binom{13}{10} = 286$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 = 10$, where x_i represents the investment in company i such that $x_i \geq 0$ for all $i = 1, 2, 3, 4$.

Problem, Part B: What if you know you want to invest at least $3 million in Company 1?

Solution: There is one way to give $3 million to Company 1. The number of ways to invest the remaining money is the same as putting 7 balls into 4 urns.

$$\text{total ways} = \binom{7+4-1}{7} = \binom{10}{7} = 120$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 = 10$, where $x_1 \geq 3$ and $x_2, x_3, x_4 \geq 0$. To translate this problem into the integer solution equation that we can solve via the divider method, we need to adjust the bounds on x_1 such that the problem becomes $x_1 + x_2 + x_3 + x_4 = 7$, where x_i is defined as in Part A.

Example 2.13. *Divider Method* used to solve an investment problem with requirements.

Problem, Part C: What if you don't have to invest all $10 million? (The economy is tight, say, and you might want to save your money.)

Solution: Imagine that you have an extra company: yourself. Now you are investing $10 million in 5 companies. Thus, the answer is the same as putting 10 balls into 5 urns.

$$\text{total ways} = \binom{10+5-1}{10} = \binom{14}{10} = 1001$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 + x_5 = 10$, such that $x_i \geq 0$ for all $i = 1, 2, 3, 4, 5$.

Example 2.14. *Divider Method* used to solve an investment problem with no minimum requirements.

We can use Julia to verify our answers in examples 2.12 to 2.14.

```
julia> (13 ⋮ 10)
286
julia> (10 ⋮ 7)
120
julia> (14 ⋮ 10)
1001
```

Example 2.15. *Binomial coefficient* in Julia to check answers from examples 2.12 to 2.14. As a shorthand, we define the alias `⋮ = binomial`, where ⋮ is created by typing `\vdots` and hitting tab. The normal syntax is `binomial(n,k)`.

`julialang.org`

3 Probability

3.1 Event Spaces and Sample Spaces

From notes by Chris Piech and Lisa Yan.

A *sample space* S is the set of all possible outcomes of an experiment. For example:

- Coin flip: $S = \{\text{Heads, Tails}\}$
- Flipping two coins: $S = \{(H,H),\ (H,T),\ (T,H),\ (T,T)\}$
- Roll of 6-sided die: $S = \{1, 2, 3, 4, 5, 6\}$
- Number of emails in a day: $S = \{x \mid x \in \mathbb{Z},\ x \geq 0\}$ (non-negative integers)
- Number of Netflix hours in a day: $S = \{x \mid x \in \mathbb{R},\ 0 \leq x \leq 24\}$

An *event space* E is some subset of S that we ascribe meaning to. In set notation, $E \subseteq S$.

- Coin flip is heads: $E = \{\text{Heads}\}$
- ≥ 1 head on 2 coin flips: $E = \{(H,H),\ (H,T),\ (T,H)\}$
- Roll of die is 3 or less: $E = \{1, 2, 3\}$
- Number of emails in a day ≤ 20: $E = \{x \mid x \in \mathbb{Z},\ 0 \leq x \leq 20\}$
- ''Wasted day'' (≤ 5 Netflix hours): $E = \{x \mid x \in \mathbb{R},\ 5 \leq x \leq 24\}$

3.2 Probability

In the 20th century, people figured out one way to define what a probability is:

$$P(E) = \lim_{n \to \infty} \frac{n(E)}{n}, \tag{3.1}$$

where n is the number of trials performed and $n(E)$ is the number of trials with an outcome in E. In English this reads: say you perform n trials of an experiment. The probability of a desired event E is defined as the ratio of the number of trials that result in an outcome in E to the number of trials performed (in the limit as your number of trials approaches infinity). You can also give other meanings to the concept of a probability, however. One common meaning ascribed is that $P(E)$ is a measure of the chance of E occurring. I often think of a probability in another way: I don't know everything about the world. As a result I have to come up with a way of expressing my belief that E will happen given my limited knowledge. This interpretation (often referred to as the *Bayesian* interpretation) acknowledges that there are two sources of probabilities: natural randomness and our own uncertainty. Later in the quarter, we will contrast the *frequentist* definition we gave you above with this other Bayesian definition of probability.

3.3 *Axioms of Probability*

Here are some basic truths about probabilities:

Axiom 1: $0 \leq P(E) \leq 1$

Axiom 2: $P(S) = 1$

Axiom 3: If E and F are mutually exclusive $(E \cap F = \emptyset)$,
then $P(E) + P(F) = P(E \cup F)$

You can convince yourself of the first axiom by thinking about the definition of probability above: when performing some number of trials of an actual experiment, it is not possible to get more occurrences of the event than there are trials (so probabilities are at most 1), and it is not possible to get less than 0 occurrences of the event (so probabilities are at least 0). The second axiom makes intuitive sense as well: if your event space is the same as the sample space, then each trial must produce an outcome from the event space. Of course, this is just a restatement of the definition of the sample space; it is sort of like saying that the probability of you eating cake (event space) if you eat cake (sample space) is 1.

3.4 *Provable Identities of Probability*

We often refer to these as corollaries that are directly provable from the three axioms given above.

Identity 1: $P(E^c) = 1 - P(E)$ $\quad (= P(S) - P(E))$

Identity 2: If $E \subseteq F$, then $P(E) \leq P(F)$

Identity 3: $P(E \cup F) = P(E) + P(F) - P(EF)$ $\quad$ (where $EF = E \cap F$)

General Inclusion-Exclusion Identity:

$$P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{r=1}^{n}(-1)^{r+1} \sum_{i_1 < \cdots < i_r} P(E_{i_1} E_{i_2} \ldots E_{i_r}) \tag{3.2}$$

This last rule is somewhat complicated, but the notation makes it look far worse than it is. What we are trying to find is the probability that any of a number of events happens. The outer sum loops over the possible sizes of event subsets (that is, first we look at all single events, then pairs of events, then subsets of events of size 3, etc.). The "-1" term tells you whether you add or subtract terms with that set size. The inner sum sums over all subsets of that size. The less-than signs ensure that you don't count a subset of events twice, by requiring that the indices $i_1, \ldots, i_r$ are in ascending order.

Here's how that looks for three events (E_1, E_2, E_3):

$$\begin{aligned} P(E_1 \cup E_2 \cup E_3) &= P(E_1) + P(E_2) + P(E_3) \\ &\quad - P(E_1E_2) - P(E_1E_3) - P(E_2E_3) \\ &\quad + P(E_1E_2E_3) \end{aligned}$$

Problem: On a university campus, 28% of all students program in Java, 7% program in Python, and 5% program in both Java and Python. You meet a random student on campus. What is the probability that they do not program in Java or Python?

Solution: Let E be the event that a randomly selected student programs in Java and F be the event that a randomly selected student programs in Python. We would like to compute $P((E \cup F)^c)$:

$$\begin{aligned} P\left((E \cup F)^c\right) &= 1 - P(E \cup F) && \text{(Identity 1)} \\ &= 1 - [P(E) + P(F) - P(EF)] && \text{(Identity 3)} \\ &= 1 - (0.28 + 0.07 - 0.05) = 0.7 \end{aligned}$$

We can confirm this by drawing a Venn diagram as seen in figure 3.1.

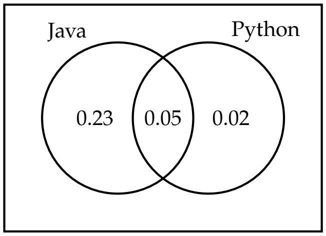


Figure 3.1. Venn diagram of the probability space for Java and Python programmers on a university campus.

3.5 *Equally Likely Outcomes*

Some sample spaces have outcomes that are all equally likely. We like those sample spaces; they make it simple to compute probabilities. Examples of sample spaces with equally likely outcomes:

- Coin flip: $S = \{\text{Heads, Tails}\}$
- Flipping two coins: $S = \{(H, H), (H, T), (T, H), (T, T)\}$
- Roll of 6-sided die: $S = \{1, 2, 3, 4, 5, 6\}$

Probability with equally likely outcomes: For a sample space S in which all outcomes are equally likely,

$$P(\text{each outcome}) = \frac{1}{|S|}$$

and for any event $E \subseteq S$,

$$P(E) = \frac{\text{number of outcomes in } E}{\text{number of outcomes in } S} = \frac{|E|}{|S|}. \tag{3.3}$$

Problem: You roll two six-sided dice. What is the probability that the sum of the two rolls is 7?

Solution: Define the sample space as a space of pairs, where the two elements are the outcomes of the first and second dice rolls, respectively. The event is the subset of this sample space where the sum of the paired elements is 7.

$$\begin{aligned} S = \{&(1,1),\ (1,2),\ (1,3),\ (1,4),\ (1,5),\ (\mathbf{1},\mathbf{6}) \\ &(2,1),\ (2,2),\ (2,3),\ (2,4),\ (\mathbf{2},\mathbf{5}),\ (2,6) \\ &(3,1),\ (3,2),\ (3,3),\ (\mathbf{3},\mathbf{4}),\ (3,5),\ (3,6) \\ &(4,1),\ (4,2),\ (\mathbf{4},\mathbf{3}),\ (4,4),\ (4,5),\ (4,6) \\ &(5,1),\ (\mathbf{5},\mathbf{2}),\ (5,3),\ (5,4),\ (5,5),\ (5,6) \\ &(\mathbf{6},\mathbf{1}),\ (6,2),\ (6,3),\ (6,4),\ (6,5),\ (6,6)\} \end{aligned}$$

$$E = \{(\mathbf{6},\mathbf{1}),\ (\mathbf{5},\mathbf{2}),\ (\mathbf{4},\mathbf{3}),\ (\mathbf{3},\mathbf{4}),\ (\mathbf{2},\mathbf{5}),\ (\mathbf{1},\mathbf{6})\}$$

Since all outcomes are equally likely, the probability of this event is:

$$P(E) = \frac{|E|}{|S|} = \frac{6}{36} = \frac{1}{6}$$

Example 3.1. *Equally likely outcomes* for two six-sided dice rolls that sum to 7. Bold indicates those rolls belonging to the event space E.

The reason we can choose either an ordered or unordered approach is because probability is a *ratio*. As we saw last time, any unordered counting task can be generated by first creating an ordered list, splitting the list at marked intervals, then dividing out by the overcounted cases due to ordering. If our sample space is ordered, then our event (being a subset of the sample space) is also ordered, and therefore we should account for the overcounted cases. However, probability being a ratio means that these overcounted cases get cancelled out.

The key to solving many of this section's problems involves (1) deciding whether to count distinct objects to create an equally likely outcome space, and then (2) defining the sample space and event space to consistently be ordered or unordered.

Example 3.2. *Equally likely outcomes* picking fruit out of a bag.

Problem: There are 4 oranges and 3 apples in a bag. You draw out 3. What is the probability that you draw 1 orange and 2 apples?

Solution 1: If we treat the oranges and apples as indistinct, we do not have a space with equally likely outcomes. We therefore treat all objects as distinct.

Suppose we treat each outcome in the sample space as an *ordered* list of three distinct items. The size of the sample space S is simply the total number of ways to order 3 of 7 distinct items: $|S| = 7 \cdot 6 \cdot 5 = 210$. We can then decompose the event E into three mutually exclusive events, where we pick the orange first, second, or third, respectively: $|E| = 4 \cdot 3 \cdot 2 + 3 \cdot 4 \cdot 2 + 3 \cdot 2 \cdot 4 = 72$. The probability of our event is therefore $P(E) = 72/210 = 12/35$.

Solution 2: Another approach is to treat each outcome in the sample space as an *unordered* group. The size of the sample space S is the total number of ways to choose any 3 of 7 distinct items: $|S| = \binom{7}{3}$. The event space is then the way to pick 1 distinct orange (out of 4) and 2 distinct apples (out of 3), which we combine with the product rule: $|E| = \binom{4}{1}\binom{3}{2}$. The probability of our event is therefore $P(E) = \frac{\binom{4}{1}\binom{3}{2}}{\binom{7}{3}} = 12/35$.

Example 3.3. *Equally likely outcomes* flipping the next cards in a deck of 52.

Problem: In a 52-card deck, cards are flipped one at a time. After the first ace (of any suit) appears, consider the next card. Is the next card more likely to be the Ace of Spades than the 2 of Clubs? (This problem is based on Example 5j in Chapter 2.5 of Ross's textbook, 10th Edition.)

Solution: **No**; the probabilities are **equal**. The difficulty of this problem stems from defining an experiment that gives equally likely outcomes while preserving the specifications of the original problem. An incorrect approach is to define the experiment as just drawing a pair of cards (first ace, next card) because then we discard all information about the cards flipped prior to the pair. Instead, consider the experiment to be shuffling the full 52-card deck, where $|S| = 52!$. We can then reconstruct all outcomes of the pairs of cards that we care about (if we so wish—but we just care about getting an equally likely outcome sample space).

Define E_{AS} as the event where the next card is the Ace of Spades. To construct a 52-card order where this event holds, we first take out the Ace of Spades, then shuffle the remaining 51 cards (51! ways), then insert the Ace of Spades immediately after the first ace (1 way). By the product rule, $|E_{AS}| = 51! \cdot 1$. Then define E_{2C} as the event where the next card is the 2 of Clubs. To construct a 52-card order where this event holds, we perform exactly the same steps, but with the 2 of Clubs instead. Then $|E_{2C}| = 51! \cdot 1$. Therefore, $P(E_{AS}) = 51!/52! = P(E_{2C})$.

For many readers, it may seem apparent that the first ace drawn could very well be the Ace of Spades, and so it is less likely that the next card is the Ace of Spades. Yet by a similar train of thought, the 2 of Clubs could very well have been drawn prior to the first ace drawn, and so we must consider all of those cases as well. This example serves to highlight the difficulty of probability: Mathematics often trumps intuition (no pun intended).

4 Conditional Probability

4.1 Conditional Probability

From notes by Chris Piech and Lisa Yan.

In English, a *conditional probability* answers the question: "What is the chance of an event E happening, given that I have already observed some other event F?" Conditional probability quantifies the notion of updating one's beliefs in the face of new evidence.

When you condition on an event happening you are entering the universe where that event has taken place. Mathematically, if you condition on F, then F becomes your new sample space. In the universe where F has taken place, all rules of probability still hold!

Definition of Conditional Probability: The probability of E given that (i.e. conditioned on) event F already happened:[1]

$$P(E \mid F) = \frac{P(EF)}{P(F)} = \frac{P(E,F)}{P(F)} = \frac{P(E \cap F)}{P(F)} \tag{4.1}$$

A visualization might help you understand this definition. Consider events E and F which have outcomes that are subsets of a sample space with 63 equally likely outcomes, each one drawn as a hexagon shown in figure 4.1.

Conditioning on F means that we have entered the world where F has happened (and F, which has 14 equally likely outcomes, has become our new sample space).

Given that event F has occured, the conditional probability that event E occurs is the subset of the outcomes of E that are consistent with F. In this case we can visually see that those are the three outcomes in $E \cap F$. Thus we have the probability:

$$P(E \mid F) = \frac{P(EF)}{P(F)} = \frac{3/63}{14/63} = \frac{3}{14} \approx 0.21$$

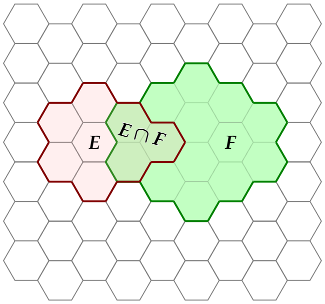


Figure 4.1. Probability of event E *conditioning* on event F.

[1] As a reminder, EF means the same thing as $E \cap F$, which is read E "and" F.

Even though the visual example (with equally likely outcome spaces) is useful for gaining intuition, the above definition of conditional probability applies regardless of whether the sample space has equally likely outcomes.

The Chain Rule: The definition of conditional probability can be rewritten as:

$$P(EF) = P(E \mid F)P(F) \tag{4.2}$$

which we call the *Chain Rule*. Intuitively, it states that the probability of observing events E and F is the probability of observing F, multiplied by the probability of observing E, given that you have observed F. Here is the general form of the Chain Rule:[2]

$$P(E_1E_2\dots E_n) = P(E_1)P(E_2 \mid E_1)\dots P(E_n \mid E_1E_2\dots E_{n-1}) \tag{4.3}$$

4.2 Law of Total Probability

An astute person once observed that in a picture like the one in figure 4.1, event F can be thought of as having two parts, the part that is in E (that is, $E \cap F = EF$), and the part that isn't ($E^c \cap F = E^cF$). This is true because E and E^c are mutually exclusive sets of outcomes which together cover the entire sample space. After further investigation this was proved to be a general mathematical truth, and there was much rejoicing:

$$P(F) = P(EF) + P(E^cF) \tag{4.4}$$

This observation is called the *law of total probability*; however, it is most commonly seen in combination with the *chain rule*.

The Law of Total Probability: For events E and F,

$$P(F) = P(F \mid E)P(E) + P(F \mid E^c)P(E^c). \tag{4.5}$$

There is a more general version of the rule. If you can divide your sample space into any number of events $E_1, E_2, \dots, E_n$ that are *mutually exclusive* and *exhaustive*—that is, *every* outcome in sample space falls into *exactly one* of those events—then:

$$P(F) = \sum_{i=1}^{n} P(F \mid E_i)P(E_i) \tag{4.6}$$

[2] A simple example of the chain rule: Let E, F, and G be events with nonzero probabilities. An equivalent expression for $P(EFG)$ would be:

$$\begin{aligned} P(EFG) &= P(E \mid FG)P(FG) \\ &= P(E \mid FG)P(F \mid G)P(G) \end{aligned}$$

The word "total" refers to the fact that the events in E_i must combine to form the totality of the sample space.

4.3 Bayes' Theorem

Bayes' theorem (or *Bayes' rule*) is one of the most ubiquitous results in probability for computer scientists. Very often we know a conditional probability in one direction, say $P(E \mid F)$, but we would like to know the conditional probability in the other direction. Bayes' theorem provides a way to convert from one to the other. We can derive Bayes' theorem by starting with the definition of conditional probability:

$$P(E \mid F) = \frac{P(FE)}{P(F)} \tag{4.7}$$

We can expand $P(FE)$ using the chain rule, which results in Bayes' theorem.

Bayes' theorem: The most common form of Bayes' theorem is:

$$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F)} \tag{4.8}$$

Each term in the Bayes' rule formula has its own name. The $P(E \mid F)$ term is often called the *posterior*; the $P(E)$ term is often called the *prior*; the $P(F \mid E)$ term is called the *likelihood* (or the *update*); and $P(F)$ is often called the *normalization constant*.

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalization constant}} \tag{4.9}$$

If the normalization constant (the probability of the event you were initially conditioning on) is not known, you can expand it using the *law of total probability*:

$$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F \mid E)P(E) + P(F \mid E^c)P(E^c)} = \frac{P(F \mid E)P(E)}{\sum_i P(F \mid E_i)P(E_i)} \tag{4.10}$$

Again, for this to hold, all the events E_i must be *mutually exclusive* and *exhaustive*.

A common scenario for applying the Bayes Rule formula is when you want to know the probability of something "unobservable" given an "observed" event. For example, you want to know the probability that a student understands a concept, given that you observed them solving a particular problem. It turns out it is much easier to first estimate the probability that a student can solve a problem given that they understand the concept and then to apply Bayes' theorem.

The ''expanded'' version of Bayes' rule in equation (4.10) allows you to work around not immediately knowing the denominator $P(F)$. It is worth exploring this in more depth, because this ''trick'' comes up often, and in slightly different forms. Another way to get to the exact same result is to reason that because the posterior of Bayes Theorem, $P(E \mid F)$, is a probability, we know that $P(E \mid F) + P(E^c \mid F) = 1$. If you expand out $P(E^c \mid F)$ using Bayes, you get:

$$P(E^c \mid F) = \frac{P(F \mid E^c)P(E^c)}{P(F)} \tag{4.11}$$

Now we have:

$$\begin{aligned}
1 &= P(E \mid F) + P(E^c \mid F) && \text{(since } P(E \mid F) \text{ is a probability)} \\
1 &= \frac{P(F \mid E)P(E)}{P(F)} + \frac{P(F \mid E^c)P(E^c)}{P(F)} && \text{(by Bayes' rule (twice))} \\
1 &= \frac{1}{P(F)}\left[P(F \mid E)P(E) + P(F \mid E^c)P(E^c)\right] \\
P(F) &= P(F \mid E)P(E) + P(F \mid E^c)P(E^c)
\end{aligned}$$

We call $P(F)$ the normalization constant because it is the term whose value can be calculated by making sure that the probabilities of all outcomes sum to 1 (they are ''normalized'').

4.4 Conditional Paradigm

As we mentioned above, when you condition on an event you enter the universe where that event has taken place, all the laws of probability still hold. Thus, as long as you condition consistently on the same event, every one of the tools we have learned still apply. Let's look at a few of our old friends when we condition consistently on an event (in this case G, often read as ''given'' G):

Applying Bayes Rule: Consider the following (hypothetical) scenario regarding an illness. Consider that 8% of all people have the illness and further that there has been a test developed for the illness with a 95% true positive rate (correctly says someone does have the illness when they do) and a 7% false positive rate (incorrectly says someone has the illness when they don't). Given that I test positive for the illness, what is the probability that I actually have the disease?

Solution: We apply Bayes Rule with an expanded denominator (using the law of total probability to find the probability of testing positive whether you have the illness or not, shown in equation (4.10)) and using ''+'' to denote the event that I test positive.

$$P(\text{ill} \mid +) = \frac{P(+ \mid \text{ill})P(\text{ill})}{P(+)} = \frac{P(+ \mid \text{ill})P(\text{ill})}{P(+ \mid \text{ill})P(\text{ill}) + P(+ \mid \text{not ill})P(\text{not ill})}$$

Since the true positive rate is 95%, $P(+ \mid \text{ill})$ is 95%, and since false positive rate is 7%, $P(+ \mid \text{not ill}) = 0.07$.

$$= \frac{(0.95)(0.08)}{(0.95)(0.08) + (0.07)(0.92)} \approx 0.541$$

Notice that now you have a much higher chance of being ill than you did before you got tested, but still only about a 1/2 chance!

Example 4.1. An application of *Bayes rule* for disease testing. Originally from the concept check for lecture 4.

Name of Rule	Original Rule	Conditional Rule
First axiom of probability	$0 \leq P(E) \leq 1$	$0 \leq P(E \mid G) \leq 1$
Corollary 1 (complement)	$P(E) = 1 - P(E^c)$	$P(E \mid G) = 1 - P(E^c \mid G)$
Chain Rule	$P(EF) = P(E \mid F)P(F)$	$P(EF \mid G) = P(E \mid FG)P(F \mid G)$
Bayes Theorem	$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F)}$	$P(E \mid FG) = \frac{P(F \mid EG)P(E \mid G)}{P(F \mid G)}$

Table 4.1. Conditional probability rules, conditioning on G.

5 Independence

5.1 Independence

From notes by Chris Piech and Lisa Yan.

Independence is a big deal in machine learning and probabilistic modeling. Knowing the ''joint'' probability of many events (the probability of the ''and'' of the events) requires exponential amounts of data. By making *independence* and *conditional independence* claims, computers can essentially decompose how to calculate the *joint probability*, making it faster to compute, and requiring less data to learn probabilities.

Independence: Two events, E and F, are *independent* if and only if:

$$P(EF) = P(E)P(F) \tag{5.1}$$

Otherwise, they are called *dependent* events.

This property applies regardless of whether or not E and F are from an equally likely sample space and whether or not the events are mutually exclusive.

The independence principle extends to more than two events. In general, n events $E_1, E_2, \ldots, E_n$ are independent if for every subset with r elements (where $r \leq n$) it holds that:

$$P(E_a, E_b, \ldots, E_r) = P(E_a)P(E_b)\ldots P(E_r) \tag{5.2}$$

The general definition implies that for three events E, F, and G to be independent, *all* of the following must be true:

$$P(EFG) = P(E)P(F)P(G) \tag{5.3}$$

$$P(EF) = P(E)P(F) \tag{5.4}$$

$$P(EG) = P(E)P(G) \tag{5.5}$$

$$P(FG) = P(F)P(G) \tag{5.6}$$

Problems with more than two independent events come up frequently. For example: the outcomes of n separate flips of a coin are all independent of one another. Each flip in this case is called a "trial" of the experiment.

In the same way that the mutual exclusion property makes it easier to calculate the probability of the OR of two events, independence makes it easier to calculate the AND of two events.

Example 5.1. Probablity of getting k heads when flipping a biased coin.

Flipping a Biased Coin: A biased coin is flipped n times. Each flip (independently) comes up heads with probability p, and tails with probability $1 - p$. What is the probability of getting exactly k heads?

Solution: Consider all the possible orderings of heads and tails that result in k heads. There are $\binom{n}{k}$ such orderings, and all of them are mutually exclusive. Since all of the flips are independent, to compute the probability of any one of these orderings, we can multiply the probabilities of each of the heads and each of the tails. There are k heads and $n - k$ tails, so the probability of each ordering is $p^k(1 - p)^{n-k}$. Adding up all the different orderings gives us the probability[1] of getting exactly k heads:

$$\binom{n}{k} p^k (1 - p)^{n-k} \tag{5.7}$$

[1] Spoiler alert: This is the probability density of a *binomial distribution*. Intrigued by that term? Stay tuned for next week!

Hash Map: Suppose m strings are hashed (unequally) into a hash table with n buckets. Each string hashed is an independent trial, with probability p_i of getting hashed to bucket i. Calculate the probability of these three events:

A. E = *the first* bucket has ≥ 1 string hashed to it.

B. E = *at least* 1 of buckets 1 to k has ≥ 1 string hashed to it.

C. E = *each* of buckets 1 to k has ≥ 1 string hashed to it.

Example 5.2. Independence of individual hashed strings in hash map.

Part A: Let S_i be the event that string i is hashed into the first bucket. Note that all S_i are independent of one another. The complement, S_i^c , is the event that string i is not hashed into the first bucket; by mutual exclusion, $P(S_i^c) = 1 - p_1 = p_2 + p_3 + + p_n$.

$$\begin{aligned}
P(E) &= P(S_1 \cup S_2 \cup \cdots \cup S_m) && \text{(definition of } S_i\text{)} \\
&= 1 - P((S_1 \cup S_2 \cup \cdots \cup S_m)^c) && \text{(complement)} \\
&= 1 - P(S_1^c S_2^c \dots S_m^c) && \text{(by De Morgan's Law)} \\
&= 1 - P(S_1^c)P(S_2^c) \dots P(S_m^c) && \text{(since the events are independent)} \\
&= 1 - (1 - p_1)^m && \text{(calculating } P(S_i) \text{ by mutual exclusion)}
\end{aligned}$$

Example 5.3. Solution to Part A of hash map *independence* example 5.2.

Part B: Let F_i be the event that at least one string is hashed into bucket i. Note that the F_i's are neither independent nor mutually exclusive.

$$\begin{aligned}
P(E) &= P(F_1 \cup F_2 \cup \cdots \cup F_k) \\
&= 1 - P([F_1 \cup F_2 \cup \cdots \cup F_k]^c) && \text{(since } P(A) + P(A^c) = 1\text{)} \\
&= 1 - P(F_1^c F_2^c \dots F_k^c) && \text{(by De Morgan's law)} \\
&= 1 - (1 - p_1 - p_2 - \cdots - p_k)^m \\
& && \text{(mutual exclusion, independence of strings)}
\end{aligned}$$

The last step is calculated by realizing that $P(F_1^c F_2^c \dots F_k^c)$ is only satisfied by m independent hashes into buckets other than 1 through k.

Example 5.4. Solution to Part B of hash map *independence* example 5.2.

Part C: Let F_i be the same as in Part B.

$$\begin{aligned}
P(E) &= P(F_1 F_2 \cdots F_k) \\
&= 1 - P([F_1 F_2 \cdots F_k]^c) && \text{(since } P(A) + P(A^c) = 1\text{)} \\
&= 1 - P(F_1^c \cup F_2^c \cup \cdots \cup F_k^c) && \text{(by De Morgan's (other) law)} \\
&= 1 - P\left(\bigcup_{i=1}^{k} F_i^c\right) \\
&= 1 - \sum_{r=1}^{k} (-1)^{r+1} \sum_{i_1 < \cdots < i_r} P(F_{i_1}^c F_{i_2}^c \ldots F_{i_r}^c) \\
& && \text{(by General Inclusion/Exclusion equation (3.2))}
\end{aligned}$$

where $P(F_1^c F_2^c \ldots F_k^c) = (1 - p_1 - p_2 - \cdots - p_k)^m$ just like in example 5.4.

Example 5.5. Solution to Part C of hash map *independence* example 5.2.

5.2 *Conditional Independence*

Two events E and F are called *conditionally independent* given a third event G, if

$$P(EF \mid G) = P(E \mid G)P(F \mid G) \tag{5.9}$$

Or, equivalently:

$$P(E \mid FG) = P(E \mid G) \tag{5.10}$$

5.2.1 *Conditioning Breaks Independence*

An important caveat about conditional independence is that ordinary independence does not imply conditional independence, nor the other way around.

Knowing when exactly conditioning breaks or creates independence is a big part of building complex probabilistic models; the first few weeks of CS 228 are dedicated to some general principles for reasoning about conditional independence. We will talk about this in another lecture. An example was included for completeness.

`cs228.stanford.edu`

Example 5.6. Dice rolls in a game of craps as *independent* trials.

Simplified Craps: Two 6-sided dice are rolled repeatedly. Consider the sum of the two dice. What is $P(E)$, where E is defined as the event where a sum of 5 is rolled before a sum of 7?

Solution: Define our independent trials to be each paired roll. We can then define F_i as the event where we observe our first 5 before 7 on the n-th trial. In other words, no 5 or 7 was rolled in the first $n-1$ trials, and a sum of 5 was rolled on the n-th trial. Notice that the F_i for $i = 1, \ldots, \infty$ are mutually exclusive, as there is only ever one first occurrence of the sum of 5. The probability of rolling a sum of 5 or a sum of 7 is $\frac{4}{36}$ and $\frac{6}{36}$, respectively, and therefore, $P(F_i) = (\frac{26}{36})^{n-1}(\frac{4}{36})$.

$$\begin{aligned} P(E) = P(F_1 \cup F_2 \cup \cdots \cup F_i \cup \cdots) &= \sum_{i=1}^{\infty} P(F_i) \qquad (F_i \text{ mutually exclusive}) \\ &= \frac{4}{36} \sum_{i=1}^{\infty} \left(\frac{26}{36}\right)^{n-1} = \frac{4}{36} \sum_{i=0}^{\infty} \left(\frac{26}{36}\right)^{n} \\ &= \frac{4}{36} \cdot \frac{1}{1 - \frac{26}{36}} = \frac{2}{5} \end{aligned}$$

where the last line comes from the property of infinite geometric series, where

$$|x| < 1 : \sum_{i=1}^{\infty} x_i = \frac{1}{1-x}. \tag{5.8}$$

Fevers: Let's say a person has a fever if they either have malaria or have an infection. We are going to assume that getting malaria and having an infection are independent: knowing if a person has malaria does not tell us if they have an infection. Now, a patient walks into a hospital with a fever. Your belief that the patient has malaria is high and your belief that the patient has an infection is high. Both explain why the patient has a fever.

Now, given our knowledge that the patient has a fever, gaining the knowledge that the patient has malaria will change your belief the patient has an infection. The malaria explains why the patient has a fever, and so the alternate explanation becomes less likely. The two events (which were previously independent) are dependent when conditioned on the patient having a fever.

Faculty Night: At faculty night with a CS professor in attendance, you observe 44 students. Of these, you find out that 30 are straight-A students. Additionally, 20 of the 44 are CS majors, and of these 20, 6 are straight-A students. Let A be the event that a student gets straight A's, C be the event that a student is a CS major, and F be the event that a student attends faculty night. In probability notation, $P(A \mid F) = 30/44 \approx 0.68$, but $P(A \mid C, F) = 6/20 = 0.30$. It would seem that being a CS major decreases your chance of being a straight-A student!

You decide to investigate further by surveying your whole dorm. There are 100 students in your dorm; 30 of these are straight-A students, 20 are CS majors, and 6 are straight-A CS majors. That is, overall, $P(A) = 30/100 = 0.30$, and $P(A \mid C) = 6/20 = 0.30$. So A and C are independent! What happened at faculty night?

As it turns out, faculty night attracted two types of people: straight-A students (who go to all the faculty nights), and CS majors. So the non-straight-A students at this faculty night are more likely to be CS majors! It's not because CS students are slackers, or because CS is harder; it's because non-straight-A students with other majors didn't come to faculty night.

In both of these examples, conditioning on an event E leads to dependence between previously independent events A and B when A and B are independent causes of E.

6 Random Variables

From notes by Chris Piech and Lisa Yan.

A *random variable* (RV) is a variable that probabilistically takes on different values. You can think of an RV as being like a variable in a programming language. They take on values, have types and have domains over which they are applicable. We can define events that occur if the random variable takes on values that satisfy a numerical test (e.g., does the variable equal 5? is the variable less than 8?). We often need to know the probabilities of such events.

As an example, let's say we flip three fair coins. We can define a random variable Y to be the total number of ''heads'' on the three coins. We can ask about the probability of Y taking on different values using the following notation:

- $P(Y = 0) = 1/8 \qquad (T, T, T)$
- $P(Y = 1) = 3/8 \qquad (H, T, T), (T, H, T), (T, T, H)$
- $P(Y = 2) = 3/8 \qquad (H, H, T), (H, T, H), (T, H, H)$
- $P(Y = 3) = 1/8 \qquad (H, H, H)$
- $P(Y \geq 4) = 0$

Even though we use the same notation for random variables and for events (both use capital letters), they are distinct concepts. An event is a situation, a random variable is an object. The situation in which a random variable takes on a particular value (or range of values) is an event. When possible, we will try to use letters E, F, G for events and X, Y, Z for random variables.

Using random variables is a convenient notation that assists in decomposing problems. There are many different types of random variables (indicator, binary, choice, Bernoulli, etc). The two main families of random variable types are discrete and continuous. For now we are going to develop intuition around discrete random variables.

6.1 Probability Mass Function

For a discrete random variable, the most important thing to know is the probability that the random variable will take on each of its possible values. The *probability mass function* (PMF) of a random variable is a function that maps possible outcomes of a random variable to the corresponding probabilities. Because it is a function, we can plot PMF graphs where the x-axis contains the values that the random variable can take on and the y-axis contains the probability of the random variable taking on said value:

There are many ways that probability mass functions can be specified. We can draw a graph. We can build a table (or for you CS folks, a `map/HashMap/dict`) that lists out all the probabilities for all possible events. Or we could write out a mathematical expression.

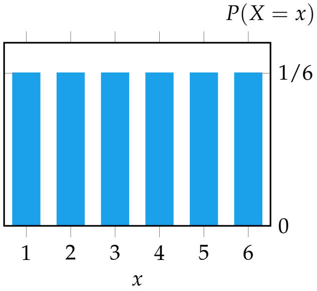


Figure 6.1. The *probability mass function* of a single 6-sided die roll.

For example, consider the random variable X which is the sum of two dice rolls. The probability mass function can be defined by the graph on the right of figure 6.1. It can also be defined using the equation:

$$p_X(x) = P(X = x) = \begin{cases} \frac{x-1}{36} & \text{if } x \in \mathbb{Z}, 1 \leq x \leq 7 \\ \frac{13-x}{36} & \text{if } x \in \mathbb{Z}, 8 \leq x \leq 12 \\ 0 & \text{otherwise} \end{cases}$$

The probability mass function, $p_X(x)$, defines the probability of X taking on the value x. The new notation $p_X(x)$ is simply different notation for writing $P(X = x)$. Using this new notation makes it more apparent that we are specifying a function. Try a few values of x, and compare the value of $p_X(x)$ to the graph in figure 6.2. They should be the same.

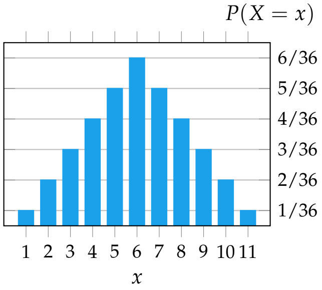


Figure 6.2. *Probability mass function* of the sum of two dice rolls.

6.2 Expectation

A useful piece of information about a random variable is the average value of the random variable over many repetitions of the experiment it represents. This average is called the *expectation*. The expectation of a discrete random variable X is defined as:

$$\mathbb{E}[X] = \sum_{x \in X} x \cdot p_X(x), \text{ where } p_X(x) > 0 \tag{6.1}$$

It goes by many other names: *mean*, *expected value*, *weighted average*, *center of mass*, and *first moment*.

The random variable X represents the outcome of one roll of a six-sided die. What is $\mathbb{E}[X]$? This is the same as asking for the average value of a die roll.

$$\mathbb{E}[X] = 1(1/6) + 2(1/6) + 3(1/6) + 4(1/6) + 5(1/6) + 6(1/6) = 7/2 = 3.5$$

Example 6.1. *Expected value* of a six-sided die roll.

6.3 Properties of Expectation

Expectations preserve *linearity*.[1] Mathematically, this means that:

$$\mathbb{E}[aX + bY + c] = a\mathbb{E}[X] + b\mathbb{E}[Y] + c \tag{6.2}$$

So if you have an expectation of a sum of quantities, this is equal to the sum of the expectations of those quantities. We will return to the implications of this very useful fact later in the course.

One can also calculate the expected value of a function $g(X)$ of a random variable X when one knows the probability distribution of X but one does not explicitly know the distribution of $g(X)$:

$$\mathbb{E}[g(X)] = \sum_{x} g(x) \cdot p_X(x) \tag{6.3}$$

This identity has the humorous name of "the Law of the Unconscious Statistician" (LOTUS), for the fact that even statisticians are known—perhaps unfairly—to ignore the difference between this identity and the basic definition of expectation (the basic definition doesn't have a function g).

[1] For a comprehensive review of *linear algebra*, see the textbook for Stanford's Math 51: `http://web.stanford.edu/class/math51/stanford/math51book.pdf`

We can use this to compute, for example, the expectation of the square of a random variable (called the *second moment* or *second central moment*):

$$\begin{aligned}
\mathbb{E}[X^2] &= \mathbb{E}[g(X)] && \text{(where } g(X) = X^2\text{)} \\
&= \sum_x g(x) \cdot p_X(x) && \text{(by LOTUS)} \\
&= \sum_x x^2 \cdot p_X(x) && \text{(definition of } g\text{)}
\end{aligned}$$

A school has 3 classes with 5, 10, and 150 students. Each student is only in one of the three classes. If we randomly choose a class with equal probability and let X be the the size of the chosen class:

$$\begin{aligned}
\mathbb{E}[X] &= 5(1/3) + 10(1/3) + 150(1/3) \\
&= 165/3 = 55
\end{aligned}$$

However, if instead we randomly choose a student with equal probability and let Y be the the size of the class the student is in:

$$\begin{aligned}
\mathbb{E}[Y] &= 5(5/165) + 10(10/165) + 150(150/165) \\
&= 22635/165 \approx 137
\end{aligned}$$

Example 6.2. Class size *expected value* based on the choice of the *random variable*.

Consider a game played with a fair coin which comes up heads with $p = 0.5$. Let n = the number of coin flips before the first tails. In this game you win \$$2^n$. How many dollars do you expect to win? Let X be a random variable which represents your winnings.

$$\begin{aligned}
\mathbb{E}[X] &= \left(\frac{1}{2}\right)^1 2^0 + \left(\frac{1}{2}\right)^2 2^1 + \left(\frac{1}{2}\right)^3 2^2 + \cdots = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{i+1} 2^i \\
&= \sum_{i=0}^{\infty} \frac{1}{2} = \infty
\end{aligned}$$

Example 6.3. *Expected value* game resulting in an infinite money paradox.

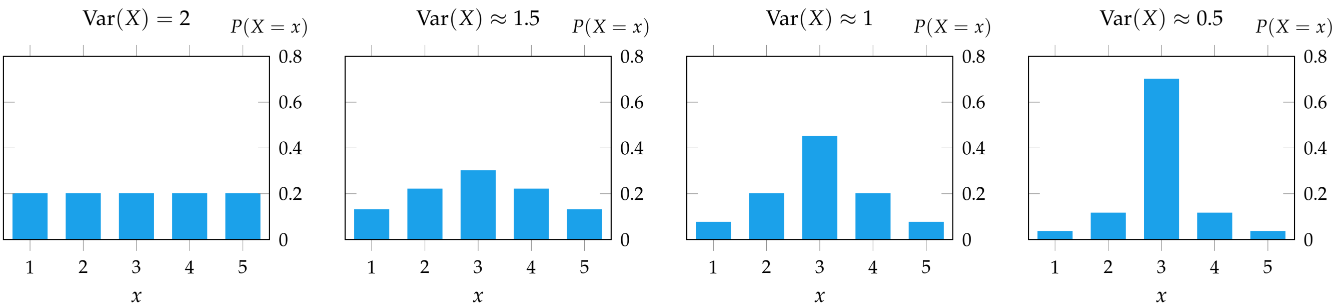


Figure 6.3. Different *variance* in *probability mass functions* that each have an expected value of $\mathbb{E}[X]{=}3$.

6.4 *Variance*

Expectation is a useful statistic, but it does not give a detailed view of the probability mass function. Consider the 4 distributions in figure 6.3 (PMFs). All four have the same expected value $\mathbb{E}[X] = 3$ but the "spread" in the distributions is quite different. *Variance* is a formal quantification of "spread". There is more than one way to quantify spread; variance uses the average square distance from the mean.

The variance of a discrete random variable X with expected value μ is defined:[2]

$$\begin{aligned} \mathrm{Var}(X) &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\ &= \mathbb{E}[(X - \mu)^2] \end{aligned} \tag{6.4}$$

When computing the variance, we often use a different form of the same equation:

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \tag{6.5, Property 1}$$

$$\begin{aligned} \mathrm{Var}(X) &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\ &= \mathbb{E}[(X - \mu)^2] && (\text{Let } \mu = \mathbb{E}[X]) \\ &= \sum_x (x - \mu)^2 p(x) \\ &= \sum_x (x^2 - 2\mu x + \mu^2) p(x) \\ &= \sum_x x^2 p(x) - 2\mu \sum_x x p(x) + \mu^2 \sum_x p(x) \\ &= \mathbb{E}[X^2] - 2\mu \mathbb{E}[X] + \mu^2 \cdot 1 \\ &= \mathbb{E}[X^2] - 2\mu^2 + \mu^2 \\ &= \mathbb{E}[X^2] - \mu^2 \\ &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \end{aligned}$$

A useful identity for variance, making it *non-linear*, is that:

$$\mathrm{Var}(aX + b) = a^2 \, \mathrm{Var}(X) \tag{6.6, Property 2}$$

Adding a constant doesn't change the "spread"; multiplying by one does.

To stay in the units of X, the *standard deviation* is the square root of variance:

$$\mathrm{SD}(X) = \sigma = \sqrt{\mathrm{Var}(X)} \tag{6.7}$$

Intuitively, standard deviation is a kind of average distance of a sample to the mean.[3] Variance is the square of this average distance.

[2] *Variance* has squared units relative to X.

[3] Specifically, it is a *root-mean-square* (RMS) average.

Let X be the value on one roll of a 6-sided die. What is $\mathrm{Var}(X)$?

Solution: First, we can calculate $\mathbb{E}[X^2]$:

$$\mathbb{E}[X^2] = (1^2)\frac{1}{6} + (2^2)\frac{1}{6} + (3^2)\frac{1}{6} + (4^2)\frac{1}{6} + (5^2)\frac{1}{6} + (6^2)\frac{1}{6} = \frac{91}{6}$$

Recall that $\mathbb{E}[X] = 7/2$, and we can use the expectation formula for variance:

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{35}{12}$$

Example 6.4. *Variance* calculation of a single 6-sided die roll.

```julia
𝔼(X) = sum(X .* P)
```

Algorithm 6.1. *Expected value* of random variable `X` with probabilities `P`, written in Julia. The symbol `𝔼` can be created by typing `\bbE` and hitting tab. The `.*` syntax broadcasts multiplication element-wise.

```julia
Var(X) = 𝔼(X.^2) - 𝔼(X)^2
```

Algorithm 6.2. *Variance* of random variable `X` using *expectation* `𝔼`.

Using the `𝔼` function from algorithm 6.1 and the `Var` function from algorithm 6.2, we can recompute the answer to example 6.4.

```julia
julia> X = 1:6;
julia> P = fill(1//6, 6);
julia> 𝔼(X)
7//2
julia> Var(X)
35//12
```

Example 6.5. *Expected value* and *variance* functions in Julia; recomputing example 6.4. Note the use of `//` to indicate a `Rational` type.

7 Bernoulli and Binomial Random Variables

From notes by Chris Piech and Lisa Yan.

There are some classic random variable abstractions that show up in many problems. At this point in the class you will learn about several of the most significant discrete distributions. When solving problems, if you are able to recognize that a random variable fits one of these formats, then you can use its precalculated probability mass function (PMF), expectation, variance, and other properties.

Random variables of this sort are called *parametric* random variables. If you can argue that a random variable falls under one of the studied parametric types, you simply need to provide parameters. A good analogy is a class in programming. Creating a parametric random variable is very similar to calling a constructor with input parameters.

7.1 Bernoulli Random Variable

A *Bernoulli random variable* maps "success" to 1 and "failure" to 0. Support for *Bernoulli*: $\{0, 1\}$

A *Bernoulli random variable* is the simplest kind of random variable. It can take on two values, 1 and 0. It takes on a 1 if an experiment with probability p resulted in success and a 0 otherwise.[1] Some example uses include a coin flip, a random binary digit, whether a disk drive crashed, and whether someone likes a Netflix movie. If X is a Bernoulli random variable, denoted[2] $X \sim \text{Ber}(p)$:

$$\begin{aligned} \textit{Probability mass function:} \quad & P(X = 1) = p && (7.1) \\ & P(X = 0) = (1 - p) && (7.2) \\ \textit{Expectation:} \quad & \mathbb{E}[X] = p && (7.3) \\ \textit{Variance:} \quad & \text{Var}(X) = p(1 - p) && (7.4) \end{aligned}$$

Bernoulli random variables and *indicator variables* are two aspects of the same concept. A random variable I is an indicator variable for an event A if $I = 1$ when A occurs and $I = 0$ if A does not occur. $P(I{=}1){=}P(A)$ and $\mathbb{E}[I]{=}P(A)$. Indicator random variables are Bernoulli random variables, with $p{=}P(A)$.

[1] The Bernoulli random variable is the simplest random variable (i.e. an *indicator* or *boolean* random variable)

[2] Sampling x from a distribution D can also be written $x \sim D$, where $\sim$ is read as "is distributed as".

7.2 *Binomial Random Variable*

A *binomial random variable* is random variable that represents the number of successes in n successive independent trials of a Bernoulli experiment. Some example uses include the number of heads in n coin flips, the number of disk drives that crashed in a cluster of 1000 computers, and the number of advertisements that are clicked when 40,000 are served.

A *binomial random variable* is the number of successes in n trials. Note that $\text{Ber}(p) = \text{Bin}(1, p)$.

Support for *binomial*: $\{0, 1, \ldots, n\}$

If X is a Binomial random variable, we denote this $X \sim \text{Bin}(n, p)$, where p is the probability of success in a given trial. A binomial random variable has the following properties:[3]

$$\textit{Probability mass function:} \quad \begin{cases} P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} & \text{if } k \in \mathbb{N},\ 0 \le k \le n \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

$$\textit{Expectation:} \quad \mathbb{E}[X] = np \tag{7.6}$$

$$\textit{Variance:} \quad \text{Var}(X) = np(1-p) \tag{7.7}$$

Example 7.1. *Binomial random variable* for $n = 3$ coin flips with probability $p = 0.5$.

Let X = number of heads after a coin is flipped three times. $X \sim \text{Bin}(3, 0.5)$. What is the probability of each of the different values of X?

$$P(X = 0) = \binom{3}{0} p^0 (1-p)^3 = \frac{1}{8}$$
$$P(X = 1) = \binom{3}{1} p^1 (1-p)^2 = \frac{3}{8}$$
$$P(X = 2) = \binom{3}{2} p^2 (1-p)^1 = \frac{3}{8}$$
$$P(X = 3) = \binom{3}{3} p^3 (1-p)^0 = \frac{1}{8}$$

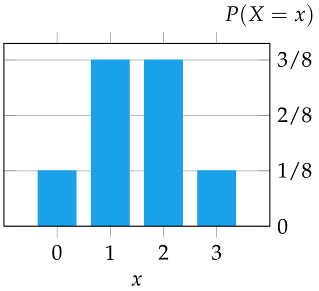


Figure 7.1. *Probability mass function* of a *binomial random variable*; number of heads after three coin flips.

[3] A binomial random variable is the sum of Bernoulli random variables.

Example 7.2. *Binomial random variable* for bit encoding using a Hamming code.

When sending messages over a network, there is a chance that the bits will be corrupted. A Hamming code allows for a 4 bit code to be encoded as 7 bits, with the advantage that if 0 or 1 bit(s) are corrupted, then the message can be perfectly reconstructed. You are working on the Voyager space mission and the probability of any bit being lost in space is 0.1. How does reliability change when using a Hamming code?

Image we use error correcting codes. Let $X \sim \text{Bin}(7, 0.1)$.

$$
\begin{aligned}
P(X=0) &= \binom{7}{0}(0.1)^0(0.9)^7 \approx 0.468 \\
P(X=1) &= \binom{7}{1}(0.1)^1(0.9)^6 = 0.372 \\
P(X=0) + P(X=1) &= 0.850
\end{aligned}
$$

```
julia> X = Bin(7, 0.1);
julia> pdf(X,0) + pdf(X,1)
0.8503056000000002
```

What if we didn't use error correcting codes? Let $X \sim \text{Bin}(4, 0.1)$.

$$
P(X=0) = \binom{4}{0}(0.1)^0(0.9)^4 \approx 0.656
$$

```
julia> X = Bin(4, 0.1);
julia> pdf(X,0)
0.6561
```

Using Hamming Codes improves reliability by about 30%!

8 Poisson and Other Discrete Distributions

8.1 Binomial in the Limit

From notes by Chris Piech and Lisa Yan.

Recall the example of sending a bit string over a network. In our last class we used a binomial random variable to represent the number of bits corrupted out of 4 with a high corruption probability (each bit had independent probability of corruption $p = 0.1$). That example was relevant to sending data to spacecraft, but for earthly applications like HTML data, voice or video, bit streams are much longer (length $\approx 10^4$) and the probability of corruption of a particular bit is very small ($p \approx 10^{-6}$). Extreme n and p values arise in many cases: # visitors to a website, # server crashes in a giant data center.

Unfortunately, $X \sim \text{Bin}(10^4, 10^{-6})$ is unwieldy to compute. However, when values get that extreme, we can make approximations that are accurate and make computation feasible. Recall that the parameters of the binomial distribution are $n = 10^4$ and $p = 10^{-6}$. First, define $\lambda = np$. We can rewrite the binomial PMF as:

$$\begin{aligned} P(X = i) &= \frac{n!}{i!(n-1)!}\left(\frac{\lambda}{n}\right)^i \left(1 - \frac{\lambda}{n}\right)^{n-i} && (8.1) \\ &= \frac{n(n-1)\dots(n-i-1)}{n^i} \cdot \frac{\lambda^i}{i!} \cdot \frac{(1-\lambda/n)^n}{(1-\lambda/n)^i} && (8.2) \end{aligned}$$

This equation can be made simpler using some approximations that hold when n is sufficiently large and p is sufficiently small:

Recall the definition:

$$e^{-\lambda} = \lim_{n\to\infty}\left(1 - \frac{\lambda}{n}\right)^n$$

$$\begin{aligned} \frac{n(n-1)\dots(n-i-1)}{n^i} &\approx 1 && (8.3) \\ (1-\lambda/n)^n &\approx e^{-\lambda} && (8.4) \\ (1-\lambda/n)^i &\approx 1 && (8.5) \end{aligned}$$

Using these reduces our original equation (8.1) to:

$$P(X = i) = \frac{\lambda^i}{i!}e^{-\lambda} \qquad (8.6)$$

This simplification, derived by assuming extreme values of n and p, turns out to be so useful that it gets its own random variable type: the *Poisson random variable*.

8.2 Poisson Random Variable

A *Poisson random variable* approximates Binomial random variables where n is large, p is small, and $\lambda = np$ is ''moderate''. Interestingly, to calculate the things we care about (e.g., PMF, expectation, variance), we no longer need to know n and p. We only need to provide λ, which we call the *rate*.

A *Poisson random variable* models number of successes of fix interval time, an approximation of $\text{Bin}(n, p)$ when n is large and p is small. Also an approximation of a *binomial* even when success in trials are not entirely independent. See chapter 7 for the *binomial random variable* definition.

There are different interpretations of ''moderate''. Commonly accepted ranges are $n > 20$ and $p < 0.05$ or $n > 100$ and $p < 0.1$.

Here are the key formulas you need to know for Poisson. If X is a Poisson random variable, denoted $X \sim \text{Poi}(\lambda)$, then:

Support for *Poisson*: $\{0, 1, \ldots\}$

$$\textit{Probability mass function:} \quad P(X = i) = \frac{\lambda^i}{i!} e^{-\lambda} \tag{8.7}$$

$$\textit{Expectation:} \quad \mathbb{E}[X] = \lambda \tag{8.8}$$

$$\textit{Variance:} \quad \text{Var}(X) = \lambda \tag{8.9}$$

Units of λ: $\dfrac{\text{\# successes}}{\text{time}}$

Poisson examples:

- # earthquakes *per* year
- # server hits *per* second
- # of emails *per* day

Example 8.1. Using the *Poisson* distribution to approximate the probability of a corrupt bit; relating it to the *Binomial* distribution.

Let's say you want to send a bit string of length $n = 10^4$ where each bit is independently corrupted with $p = 10^{-6}$. What is the probability that the message will arrive uncorrupted? You can solve this using a Poisson with $\lambda = np = 10^4 10^{-6} = 0.01$. Let $X \sim \text{Poi}(0.01)$ be the number of corrupted bits. Using the PMF for Poisson:

$$P(X = 0) = \frac{\lambda^i}{i!} e^{-\lambda} = \frac{0.01^0}{0!} e^{-0.01} \approx 0.9900498$$

```
julia> X = Poi(0.01);
julia> pdf(X,0)
0.9900498337491681
```

We could have also modeled X as a binomial such that $X \sim \text{Bin}(10^4, 10^{-6})$. That would have been harder to compute but would have resulted in the same number (to 8 decimal places).

```
julia> Xₙ = Bin(10^4,10^-6);
julia> pdf(X,0) - pdf(Xₙ,0)
4.950252541213729e-9
```

Example 8.2. A *Poisson* distribution used to approximate the probability of an earthquake.

The Poisson distribution is often used to model the number of events that occur independently at any time in an interval of time or space, with a constant average rate. Earthquakes are a good example of this. Suppose there are an average of 2.8 major earthquakes in the world each year. What is the probability of getting more than one major earthquake next year?

Let $X \sim \text{Poi}(2.8)$ be the number of major earthquakes next year. We want to know $P(X > 1)$. We can use the complement rule to rewrite this as $1 - P(X = 0) - P(X = 1)$. Using the PMF for Poisson:

$$\begin{aligned} P(X > 1) &= 1 - P(X = 0) - P(X = 1) \\ &= 1 - e^{-2.8}\frac{2.8^0}{0!} - e^{-2.8}\frac{2.8^1}{1!} = 1 - e^{-2.8} - 2.8e^{-2.8} \\ &\approx 1 - 0.06 - 0.17 = 0.77 \end{aligned}$$

```julia
julia> X = Poi(2.8);
julia> 1 - pdf(X,0) - pdf(X,1)
0.7689217620241717
```

8.3 Geometric Distribution

A *geometric random variable* is the number of trials until the first success.

Support for *geometric*: $\{1, 2, \ldots\}$

The variable X is a *geometric random variable*, denoted $X \sim \text{Geo}(p)$, if X is number of the independent trials until the first success and p is probability of success on each trial. If $X \sim \text{Geo}(p)$:

$$\textit{Probability mass function:} \quad P(X = n) = (1 - p)^{n-1}p \qquad (8.10)$$

$$\textit{Expectation:} \quad \mathbb{E}[X] = 1/p \qquad (8.11)$$

$$\textit{Variance:} \quad \text{Var } X = (1 - p)/p^2 \qquad (8.12)$$

The PMF, $P(X = n)$, can be derived using the independence assumption. Let E_i represent the event that the i-th trial succeeds. Then the probability that X is exactly n is the probability that the first $n - 1$ trials fail, and the n-th succeeds:

$$\begin{aligned} P(X = n) &= P(E_1^c E_2^c \ldots E_{n-1}^c E_n) && (8.13) \\ &= P(E_1^c)P(E_2^c) \ldots P(E_{n-1}^c)P(E_n) && (8.14) \\ &= (1 - p)^{n-1}p && (8.15) \end{aligned}$$

A similar argument can be used to derive the *cumulative distribution function* (CDF), the probability that $X \leq n$. This is equal to $1 - P(X > n)$, and $P(X > n)$ is the probability that at least the first n trials fail:

$$\begin{aligned} P(X \leq n) &= 1 - P(X > n) && (8.16)\\ &= 1 - P(E_1^c E_2^c \ldots E_n^c) && (8.17)\\ &= 1 - P(E_1^c)P(E_2^c)\ldots P(E_n^c) && (8.18)\\ &= 1 - (1-p)^n && (8.19) \end{aligned}$$

In the *Pokémon* games, one captures Pokémon by throwing Poké Balls at them. Suppose each ball independently has probability $p = 0.1$ of catching the Pokémon. What is the average number of balls required for a successful capture?

Solution: Let X be the number of balls used until (and including) the capture. $X \sim \text{Geo}(p)$, so the average number needed is $\mathbb{E}[X] = 1/p = 10$.

Example 8.3. *Geometric random variable* of independent *Pokémon* capturing trials. Use `X = Geo(p)` for the distribution and `𝔼(X)` for expectation.

Suppose we want to ensure that the probability of a capture before we run out of Poké Balls is at least 0.99. How many balls do we need to carry?

Solution: We want to know n such that $P(X \leq n) \geq 0.99$.

$$\begin{aligned} P(X \leq n) = 1 - (1-p)^n &\geq 0.99\\ (1-p)^n &\leq 0.01\\ \log[(1-p)^n] &\leq \log 0.01\\ n\log(1-p) &\leq \log 0.01\\ n &\geq \frac{\log 0.01}{\log(1-p)} = \frac{\log 0.01}{\log 0.9} \approx 43.7 \end{aligned}$$

So we need 44 Poké Balls. (Note that we flipped the inequality on the last line because we divided both sides by $\log(1-p)$. Since $1 - p < 1$, we know $\log(1-p) < 0$, so we're dividing by a negative number!)

Example 8.4. *Geometric random variable* of independent *Pokémon* capturing trials, using the *cumulative distribution function* `cdf(X,n)`, introduced in section 9.2.

8.4 Negative Binomial Distribution

A variable X is a *negative binomial random variable*, denoted $X \sim \text{NegBin}(r, p)$, if X is the number of independent trials until r successes and p is probability of success on each trial. If $X \sim \text{NegBin}(r, p)$:

A *negative binomial random variable* is the number of trials until r successes.

Support for *negative binomial*: $\{r, r+1, \dots\}$

$$\textit{Probability mass function:} \quad P(X = n) = \binom{n-1}{r-1} p^r (1-p)^{n-r} \qquad \text{where } r \le n \tag{8.20}$$

$$\textit{Expectation:} \quad \mathbb{E}[X] = r/p \tag{8.21}$$

$$\textit{Variance:} \quad \operatorname{Var} X = r(1-p)/p^2 \tag{8.22}$$

Problem: A grad student needs 3 published papers to graduate. (Not how it works in real life!) On average, how many papers will the student need to submit to a conference, if the conference accepts each paper randomly and independently with probability $p = 0.25$? (Also not how it works in real life...though the NIPS Experiment suggests there is a grain of truth in this model!)

Solution: Let X be the number of submissions required to get $r = 3$ acceptances.

$$X \sim \text{NegBin}(r = 3, p = 0.25)$$

Therefore we get:

$$\mathbb{E}[X] = \frac{r}{p} = \frac{3}{0.25} = 12$$

```
julia> X = NegBin(3, 0.25);
julia> 𝔼(X)
12.0
```

Example 8.5. Using the *negative binomial distribution* to determine conference submissions required for acceptance.

http://blog.mrtz.org/2014/12/15/the-nips-experiment.html

9 Continuous Distributions

From notes by Chris Piech and Lisa Yan.

So far, all random variables we have seen have been *discrete*. In all the cases we have seen in CS 109, this meant that our RVs could only take on integer values. Now it's time for *continuous random variables*, which can take on values in the real number domain $\mathbb{R}$. Continuous random variables can be used to represent measurements with arbitrary precision (e.g., height, weight, or time).

9.1 Probability Density Functions

In the world of discrete random variables, the most important property of a random variable was its probability mass function (PMF), which told you the probability of the random variable taking on a certain value. When we move to the world of continuous random variables, we are going to need to rethink this basic concept. If I were to ask you what the probability is of a child being born with a weight of *exactly* 3.523112342234 kilograms, you might recognize that question as ridiculous. No child will have precisely that weight. Real values are defined with infinite precision; as a result, the probability that a random variable takes on a specific value is not very meaningful when the random variable is continuous. The PMF doesn't apply. We need another idea.

In the continuous world, every random variable has a *probability density function* (PDF), which says how likely it is that a random variable takes on a particular value, relative to other values that it could take on. The PDF has the nice property that you can integrate over it to find the probability that the random variable takes on values within a range (a, b).

The random variable X is a *continuous random variable* if there is a function $f(x)$ for $-\infty \leq x \leq \infty$, called the *probability density function* (PDF), such that:

See appendix B.4 for a calculus review.

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{9.1}$$

To preserve the axioms that guarantee $P(a \leq X \leq b)$ is a probability, the following properties must also hold:

$$0 \leq P(a \leq X \leq b) \leq 1 \tag{9.2}$$

$$P(-\infty < X < \infty) = 1 \tag{9.3}$$

A common misconception is to think of $f(x)$ as a probability. It is instead what we call a probability density. It represents probability *divided by the units of* X. Generally this is only meaningful when we either take an integral over the PDF **or** we *compare* probability densities. As we mentioned when motivating probability densities, the probability that a continuous random variable takes on a specific value (to infinite precision) is 0.

Integrate $f(x)$ to get probabilities.

$$P(X = a) = \int_a^a f(x)dx = 0 \tag{9.4}$$

This is very different from the discrete setting, in which we often talked about the probability of a random variable taking on a particular value exactly.

PDF units: probability per units of X.

9.2 Cumulative Distribution Function

Having a probability density is great, but it means we are going to have to solve an integral every single time we want to calculate a probability. To save ourselves some effort, for most of these variables we will also compute a *cumulative distribution function* (CDF). The CDF is a function which takes in a number and returns the probability that a random variable takes on a value *less than* (*or equal to*) that number. If we have a CDF for a random variable, we don't need to integrate to answer probability questions!

For a continuous random variable X, the *cumulative distribution function* is:

$$F_X(a) = P(X \leq a) = \int_{-\infty}^a f(x)dx \tag{9.5}$$

This can be written $F(a)$, without the subscript, when it is obvious which random variable we are using.

Why is the CDF the probability that a random variable takes on a value *less than* (*or equal to*) the input value as opposed to greater than? It is a matter of convention. But it is a useful convention. Most probability questions can be solved simply by knowing the CDF (and taking advantage of the fact that the integral over the range $-\infty$ to ∞ is 1). Here are a few examples of how you can answer probability questions by just using a CDF:

Probability Query	Solution	Explanation
$P(X \leq a)$	$F(a)$	This is the definition of the CDF
$P(X < a)$	$F(a)$	Note that $P(X = a) = 0$
$P(X > a)$	$1 - F(a)$	$P(X \leq a) + P(X > a) = 1$
$P(a < X < b)$	$F(b) - F(a)$	$F(a) + P(a < X < b) = F(b)$

As we mentioned briefly earlier, the cumulative distribution function can also be defined for discrete random variables, but there is less utility to a CDF in the discrete world, because with the exception of the geometric random variable, none of our discrete random variables had ''closed form'' (that is, without any summations) functions for the CDF:

$$F_X(a) = \sum_{i=0}^{a} P(X = i) \tag{9.6}$$

9.3 *Expectation and Variance*

For a continuous random variable X:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \cdot f(x) dx \tag{9.7}$$

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x) \cdot f(x) dx \tag{9.8}$$

$$\mathbb{E}[X^n] = \int_{-\infty}^{\infty} x^n \cdot f(x) dx \tag{9.9}$$

Similar to the discrete case, but summation is replaced with integration and $p(x)$ is replaced with the PDF $f(x)$.

Let X be a continuous random variable (CRV) with PDF:

$$f(x) = \begin{cases} C(4x - 2x^2) & \text{if } 0 < x < 2 \\ 0 & \text{otherwise} \end{cases}$$

In this function, C is a constant. What value is C? Since we know that the PDF must sum to 1:

$$\begin{aligned} \int_0^2 C(4x - 2x^2) dx &= 1 \\ C\left(2x^2 - \frac{2x^3}{3}\right)\Bigg|_{x=0}^{2} &= 1 \\ C\left(\left(8 - \frac{16}{3}\right) - 0\right) &= 1 \end{aligned}$$

Solving this equation for C gives $C = 3/8$. What is $P(X > 1)$?

$$\begin{aligned} \int_1^\infty f(x)dx &= \int_1^2 \frac{3}{8}(4x - 2x^2)dx \\ &= \frac{3}{8}\left(2x^2 - \frac{2x^3}{3}\right)\Bigg|_{x=1}^{2} \\ &= \frac{3}{8}\left[\left(8 - \frac{16}{3}\right) - \left(2 - \frac{2}{3}\right)\right] = \frac{1}{2} \end{aligned}$$

Example 9.1. Using the properties of the *probability density function* (PDF) of a *continuous random variable* (CRV) to solve for an unknown.

Let X be a RV representing the number of days of use before your disk crashes, with PDF:

$$f(x) = \begin{cases} \lambda e^{-x/100} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

First, determine λ. Recall that $\int Ae^{Au}du = e^{Au}$, from equation (B.32):

$$\begin{aligned} \int_0^\infty \lambda e^{-x/100} dx &= 1 \\ -100\lambda \int_0^\infty \frac{-1}{100} e^{-x/100} dx &= 1 \\ -100\lambda \cdot e^{-x/100} \Big|_{x=0}^{\infty} &= 1 \\ 100\lambda \cdot 1 &= 1 \implies \lambda = \frac{1}{100} \end{aligned}$$

What is $P(X < 10)$?

$$\begin{aligned} F(10) &= \int_0^{10} \frac{1}{100} e^{-x/100} dx \\ &= -e^{-x/100} \Big|_{x=0}^{10} \\ &= -e^{-1/10} + 1 \approx 0.095 \end{aligned}$$

Example 9.2. Calculating the *cumulative distribution function* for disk crashes.

For both continuous and discrete RVs:

$$\begin{aligned} \mathbb{E}[aX+b] &= a\mathbb{E}[X]+b \\ \operatorname{Var}(X) &= \mathbb{E}[(X-\mu)^2] && \text{(with } \mu = \mathbb{E}[X]\text{)} \\ &= \mathbb{E}[X^2]-\mathbb{E}[X]^2 \\ \operatorname{Var}(aX+b) &= a^2\operatorname{Var}(X) \end{aligned}$$

9.4 Uniform Random Variable

Support for *Uniform*: $[\alpha, \beta]$

The most basic of all the continuous random variables is the *uniform random variable*, which is equally likely to take on any value in its range $[\alpha, \beta]$.

The variable X is a *uniform random variable*, denoted $X \sim \text{Uni}(\alpha, \beta)$, if it has probability density function (PDF):

$$f(x) = \begin{cases} \dfrac{1}{\beta-\alpha} & \text{if } \alpha \leq x \leq \beta \\ 0 & \text{otherwise} \end{cases} \tag{9.10}$$

Notice how the density $1/(\beta-\alpha)$ is exactly the same regardless of the value for x. That makes the density uniform. So why is the PDF $1/(\beta-\alpha)$ and not 1? That is the constant that makes it such that the integral over all possible inputs evaluates to 1.

The cumulative distribution function (CDF), expectation, and variance of the *uniform random variable* are:

$$\begin{aligned} P(a \leq X \leq b) &= \int_a^b f(x)dx \qquad \text{for } \alpha \leq a \leq b \leq \beta && (9.11) \\ &= \frac{b-a}{\beta-\alpha} && (9.12) \\ \mathbb{E}[X] &= \int_{-\infty}^{\infty} x \cdot f(x)dx && (9.13) \\ &= \int_\alpha^\beta \frac{x}{\beta-\alpha}dx = \left.\frac{x^2}{2(\beta-\alpha)}\right|_{x=\alpha}^{\beta} && (9.14) \\ &= \frac{\alpha+\beta}{2} && (9.15) \\ \operatorname{Var} X &= \frac{(\beta-\alpha)^2}{12} && (9.16) \end{aligned}$$

9.5 *Exponential Random Variable*

Support for *Exponential*: $[0, \infty)$

Units of λ: $\frac{\text{event}}{\text{time}}$

An *exponential random variable*, denoted $X \sim \text{Exp}(\lambda)$, represents the time until an event occurs. It is parametrized by $\lambda > 0$, the (constant) rate at which the event occurs. This is the same λ as in the Poisson distribution; a Poisson variable counts the *number of events that occur* in a fixed interval, while an exponential variable measures the *amount of time until the next event occurs*.[1]

The probability density function (PDF) for an *exponential random variable* is:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9.17}$$

The expectation and variance are as follows:

$$\mathbb{E}[X] = \frac{1}{\lambda} \tag{9.18}$$

$$\text{Var}(X) = \frac{1}{\lambda^2} \tag{9.19}$$

There is a closed form for the cumulative distribution function (CDF):

$$F(x) = 1 - e^{-\lambda x} \text{ where } x \geq 0 \tag{9.20}$$

Exponential examples:

- time until next earthquake
- time for request to reach web server
- time until end of cell phone contract

[1] Example 9.2 sneakily introduced you to the exponential distribution already; now we get to use formulas we've already computed to work with it without integrating anything.

Let X be a random variable that represents the number of minutes until a visitor leaves your website. You have calculated that on average a visitor leaves your site after 5 minutes, and you decide that an exponential distribution is appropriate to model how long a person stays before leaving the site. What is the $P(X > 10)$?

We can compute $\lambda = \frac{1}{5}$ either using the definition of $\mathbb{E}[X]$ or by thinking of how many people leave every minute (answer: ''one-fifth of a person''). Thus $X \sim \text{Exp}(1/5)$.

$$\begin{aligned} P(X > 10) &= 1 - F(10) \\ &= 1 - (1 - e^{-\lambda \cdot 10}) \\ &= e^{-2} \\ &\approx 0.1353 \end{aligned}$$

Example 9.3. Using an *exponential random variable* to determine duration a user stays on a website.

Let X be the number of hours of use until your laptop dies. On average, laptops die after 5000 hours of use. If you use your laptop for 7300 hours during your undergraduate career (assuming usage equals 5 hours/day and four years of university), what is the probability that your laptop lasts all four years?

As above, we can find λ either using $\mathbb{E}[X]$ or thinking about laptop deaths per hour: Therefore, $X \sim \text{Exp}(\frac{1}{5000})$.

$$\begin{aligned} P(X > 7300) &= 1 - F(7300) \\ &= 1 - (1 - e^{-7300/5000}) \\ &= e^{-1.46} \approx 0.2322 \end{aligned}$$

Example 9.4. Using an *exponential random variable* to determine if your laptop will last all four years of university.

10 Normal Distribution

10.1 Normal Random Variable

From notes by Chris Piech and Lisa Yan.

The single most important random variable type is the *Normal* (aka *Gaussian*) random variable, parameterized by a mean μ and variance σ^2. If X is a normal variable we write $X \sim \mathcal{N}(\mu, \sigma^2)$. The normal is important for many reasons: it is generated from the summation of independent random variables and as a result it occurs often in nature. Many things in the world are not distributed normally but data scientists and computer scientists model them as Normal distributions anyways. Why? Because it is the most entropic (conservative) distribution that we can apply to data with a measured mean and variance.

Support for *Normal*: $\{-\infty, \infty\}$

The probability density function (PDF) for a Normal is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{10.1}$$

$$\frac{1}{\text{normalizing constant}} \left(\text{exponential tail} \right)^{\frac{\text{symmetric around } \mu}{\text{variance } \sigma^2 \text{ manages spread}}}$$

By definition, a Normal has $\mathbb{E}[X] = \mu$ and $\text{Var}(X) = \sigma^2$.

There is no closed form for the integral of the Normal PDF, however since a linear transform of a Normal produces another Normal we can always map our distribution to the *Standard Normal* (mean 0 and variance 1) which has a precomputed cumulative distribution function (CDF). The CDF of an arbitrary normal is:

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) \tag{10.2}$$

where Φ is a precomputed function that represents that CDF of the Standard Normal.

The PDF of a Normal is symmetric about the mean μ:

$$f(\mu - x) = 1 - f(\mu + x)$$

Symmetry of the PDF of a Normal implies (for the standard normal CDF):

$$\Phi(-x) = 1 - \Phi(x)$$

10.2 *Linear Transform*

If X is a Normal such that $X \sim \mathcal{N}(\mu, \sigma^2)$ and Y is a linear transform of X such that $Y = aX + b$, then Y is also a Normal where:

$$Y \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$$

10.3 *Projection to Standard Normal*

For any Normal random variable X, we can find a linear transform from X to the *Standard Normal* $\mathcal{N}(0,1)$. That is, if you subtract the mean μ of the normal and divide by the standard deviation σ, the result is distributed according to the standard normal (also called the *unit Normal*). We can prove this mathematically. Let $Z = \frac{X-\mu}{\sigma}$:

$$\begin{aligned}
Z &= \frac{X-\mu}{\sigma} && \text{(Transform } X\text{: subtract } \mu \text{ and divide by } \sigma\text{)} \\
&= \frac{1}{\sigma}X - \frac{\mu}{\sigma} && \text{(Use algebra to rewrite the equation)} \\
&= aX + b && \text{(Define } a = \tfrac{1}{\sigma},\ b = -\tfrac{\mu}{\sigma}\text{)} \\
&\sim \mathcal{N}(a\mu + b, a^2\sigma^2) && \text{(The linear transform of a normal is another normal)} \\
&\sim \mathcal{N}\left(\frac{\mu}{\sigma} - \frac{\mu}{\sigma}, \frac{\sigma^2}{\sigma^2}\right) && \text{(Substitute values in for } a \text{ and } b\text{)} \\
&\sim \mathcal{N}(0,1) && \text{(The Standard Normal)}
\end{aligned}$$

An extremely common use of this transform is to express $F_X(x)$, the CDF of X, in terms of the CDF of Z, $F_Z(x)$. Since the CDF of the Standard Normal is so common, it gets its own Greek symbol, $\Phi(x)$.

$$\begin{aligned}
F_X(x) &= P(X \le x) && (10.3) \\
&= P\left(\frac{X-\mu}{\sigma} \le \frac{x-\mu}{\sigma}\right) && (10.4) \\
&= P\left(Z \le \frac{x-\mu}{\sigma}\right) && (10.5) \\
&= \Phi\left(\frac{x-\mu}{\sigma}\right) && (10.6)
\end{aligned}$$

Why is this useful? Well, in the days when we couldn't call `scipy.stats.norm.cdf` (or on exams, when one doesn't have a calculator), people would look up values of the CDF in a table. Using the Standard Normal means you only need to build a table of one distribution, rather than an indefinite number of tables for all the different values of μ and σ. We also have an online calculator on the CS 109 website. You should learn how to use the Standard Normal table for the exams, however!

Example 10.1. *Normal distribution* using the defined CDF.

Let $X \sim \mathcal{N}(3, 16)$, what is $P(X > 0)$?

$$\begin{aligned} P(X > 0) &= P\left(\frac{X-3}{4} > \frac{0-3}{4}\right) = P\left(Z > -\frac{3}{4}\right) = 1 - P\left(Z \le -\frac{3}{4}\right) \\ &= 1 - \Phi\left(-\frac{3}{4}\right) = 1 - \left(1 - \Phi\left(\frac{3}{4}\right)\right) = \Phi\left(\frac{3}{4}\right) \approx 0.7734 \end{aligned}$$

An alternative approach uses the idea that if F is the CDF of $X \sim \mathcal{N}(\mu, \sigma^2)$, then $F(x) = P\left(Z < \frac{x-\mu}{\sigma}\right) = \Phi\left(\frac{x-\mu}{\sigma}\right)$:

$$\begin{aligned} P(X > 0) &= 1 - F(0) = 1 - \Phi(-3/4) \\ &= 1 - (1 - \Phi(3/4)) = \Phi(3/4) \approx 0.7734 \end{aligned}$$

What is $P(2 < X < 5)$?

$$\begin{aligned} P(2 < X < 5) &= P\left(\frac{2-3}{4} < \frac{X-3}{4} < \frac{5-3}{4}\right) = P\left(-\frac{1}{4} < Z < \frac{2}{4}\right) \\ &= \Phi\left(\frac{2}{4}\right) - \Phi\left(-\frac{1}{4}\right) = \Phi\left(\frac{1}{2}\right) - \left(1 - \Phi\left(\frac{1}{4}\right)\right) \approx 0.2902 \end{aligned}$$

Alternative solution:

$$\begin{aligned} P(2 < X < 5) &= F(5) - F(2) = \Phi\left(\frac{5-3}{4}\right) - \Phi\left(\frac{2-3}{4}\right) \\ &= \Phi(1/2) - (1 - \Phi(1/4)) \approx 0.2902 \end{aligned}$$

What is $P(|X - 3| < 6)$?

$$\begin{aligned} P(|X-3| > 6) &= P(X < -3) + P(X > 9) = F(-3) + (1 - F(9)) \\ &= \Phi\left(\frac{-3-3}{4}\right) + \left(1 - \Phi\left(\frac{9-3}{4}\right)\right) \\ &= \Phi(-3/2) + (1 - \Phi(3/2)) = 2(1 - \Phi(3/2)) \approx 0.1337 \end{aligned}$$

You send voltage of 2 or -2 on a wire to denote 1 or 0. Let X = voltage sent and let R = voltage received. Note $R = X + Y$, where $Y \sim \mathcal{N}(0, 1)$ is noise. When decoding, if $R \geq 0.5$, we interpret the voltage as 1, else 0.

Example 10.2. *Standard Normal distribution* used as signal noise.

What is $P(\text{error after decoding} \mid \text{original bit } = 1)$?

Given that we sent a 1, $X = 2$ and therefore $R = 2 + Y$. A decoding error occurs if we incorrectly interpret the signal as 0; this occurs if $R < 0.5$. Note that Y is the Standard Normal and therefore has CDF Φ:

$$\begin{aligned} P(R < 0.5 \mid X = 2) &= P(X + Y < 0.5 \mid X = 2) = P(2 + Y < 0.5) \\ &= P(Y < -1.5) = \Phi(-1.5) = 1 - \Phi(1.5) \approx 0.0668 \end{aligned}$$

What is $P(\text{error after decoding} \mid \text{original bit } = 0)$?

Given that we sent a 0, $X = -2$ and therefore $R = -2 + Y$. A decoding error occurs if we incorrectly interpret the signal as 1; this occurs if $R \geq 0.5$.

$$\begin{aligned} P(R \geq 0.5 \mid X = -2) &= P(X + Y \geq 0.5 \mid X = -2) = P(-2 + Y \geq 0.5) \\ &= P(Y \geq 2.5) = 1 - \Phi(2.5) \approx 0.0062 \end{aligned}$$

This example demonstrates an asymmetric decoding boundary, where there is lower probability of erroneously decoding a 0 as a 1 than vice versa. In many engineering circumstances, we may suffer stronger consequences if we turn something ‘‘on’’ when it was supposed to stay turned off. By setting the boundary of our decoding process asymmetrically, we can decrease the probability of this undesirable error.

10.4 *Binomial Approximation*

You can use a Normal distribution to approximate a Binomial $X \sim \text{Bin}(n, p)$. To do so, define a normal distribution:

$$Y \sim \mathcal{N}(\mathbb{E}[X], \text{Var}(X))$$

Now using the Binomial formulas for expectation and variance:

$$\mu = np$$
$$\sigma^2 = np(1-p)$$

$$Y \sim \mathcal{N}(np, np(1-p))$$

This approximation holds for large n. Since a Normal is continuous and Binomial is discrete, we have to use a continuity correction to discretize the Normal:

$$P(X = k) \approx P\left(k - \frac{1}{2} < Y < k + \frac{1}{2}\right) = \Phi\left(\frac{k - np + 0.5}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{k - np - 0.5}{\sqrt{np(1-p)}}\right)$$

100 visitors to your website are given a new design. Let X = # of people who were given the new design and spend more time on your website. Your CEO will endorse the new design if $X \geq 65$.

What is $P(\text{CEO endorses change} \mid \text{it has no effect})$?

$$\mathbb{E}[X] = np = 50$$
$$\text{Var}(X) = np(1-p) = 25$$
$$\sigma = \sqrt{\text{Var}(X)} = 5$$

We can thus use a Normal approximation: $Y \sim \mathcal{N}(50, 25)$.

$$P(X \geq 65) \approx P(Y > 64.5) = P\left(\frac{Y - 50}{5} > \frac{64.5 - 50}{5}\right)$$
$$= 1 - \Phi(2.9) \approx 0.0019$$

Example 10.3. Approximating a *binomial distribution* with a *Normal distribution* for website visit statistics.

Discrete	**Continuous**
$x = 6$	$5.5 < x < 6.5$
$x > 6$	$x > 6.5$
$x \leq 6$	$x < 6.5$
$x < 6$	$x < 5.5$
$x \geq 6$	$x > 5.5$

Stanford accepts 2480 students and each student has a 68% chance of attending. Let X = # students who will attend. $X \sim \text{Bin}(2480, 0.68)$. What is $P(X > 1745)$?

$$\begin{aligned} \mathbb{E}[X] &= np = 1686.4 \\ \text{Var}(X) &= np(1-p) = 539.7 \\ \sigma &= \sqrt{\text{Var}(X)} = 23.23 \end{aligned}$$

We can thus use a Normal approximation: $Y \sim \mathcal{N}(1686.4, 539.7)$.

$$\begin{aligned} P(X > 1745) \approx P(Y > 1745.5) &= P\left(\frac{Y - 1686.4}{23.23} > \frac{1745.5 - 1686.4}{23.23}\right) \\ &= 1 - \Phi(2.54) \approx 0.0055 \end{aligned}$$

Example 10.4. Approximating a *binomial distribution* with a *Normal distribution* for Stanford acceptance statistics.

11 Joint Distributions

11.1 Joint Distributions

From notes by by Chris Piech and Lisa Yan.

Often you will work on problems where there are several random variables (often interacting with one another). We are going to start to formally look at how those interactions play out.

For now we will think of *joint probabilities* with two events $X = a$ and $Y = b$. For this chapter, we will assume both X and Y are discrete random variables, and we will tackle the continuous in a future chapter.

11.2 Discrete Case

In the discrete case, a joint probability mass function tells you the probability of any combination of events $X = a$ and $Y = b$:

$$p_{X,Y}(a,b) = P(X = a, Y = b) \tag{11.1}$$

This function tells you the probability of all combinations of events (the ",'' means "and''). If you want to back calculate the probability of an event only for one variable you can calculate a *marginal* from the joint probability mass function:

$$p_X(a) = P(X = a) = \sum_y p_{X,Y}(a,y) \tag{11.2}$$

$$p_Y(b) = P(Y = b) = \sum_x p_{X,Y}(x,b) \tag{11.3}$$

In the continuous case, a joint probability density function tells you the relative probability of any combination of events $X = a$ and $Y = y$.

In the discrete case, we can define the function $p_{X,Y}$ non-parametrically. Instead of using a formula for p we simply state the probability of each possible outcome.

11.3 Multinomial Distribution

Say you perform n independent trials of an experiment where each trial results in one of m outcomes, with respective probabilities: $p_1, p_2, \ldots, p_m$ (constrained so that $\sum_i p_i = 1$). Define X_i to be the number of trials with outcome i. A *multinomial distribution* is a closed form function that answers the question: What is the probability that there are c_i trials with outcome i. Mathematically:

$$P(X_1 = c_1, X_2 = c_2, \ldots, X_m = c_m) = \binom{n}{c_1, c_2, \ldots, c_m} p_1^{c_1} p_2^{c_2} \ldots p_m^{c_m} \tag{11.4}$$

A 6-sided die is rolled 7 times. What is the probability that you roll: 1 one, 1 two, 0 threes, 2 fours, 0 fives, 3 sixes (disregarding order).

$$P(X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 2, X_5 = 0, X_6 = 3) =$$
$$\frac{7!}{2!3!}\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^2\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^3 = 420\left(\frac{1}{6}\right)^7$$

Example 11.1. *Multinomial distribution* to calculate the *joint probability* of outcomes from rolls of a 6-sided die.

Federalist Papers: In class, we wrote a program to decide whether or not James Madison or Alexander Hamilton wrote Fedaralist Paper 49. Both men have claimed to be have written it, and hence the authorship is in dispute. First we used historical essays to estimate p_i, the probability that Hamilton generates the word i (independent of all previous and future choices or words). Similarly we estimated q_i, the probability that Madison generates the word i. For each word i we observe the number of times that word occurs in Fedaralist Paper 49 (we call that count c_i). We assume that, given no evidence, the paper is equally likely to be written by Madison or Hamilton.

Define three events: H is the event that Hamilton wrote the paper, M is the event that Madison wrote the paper, and D is the event that a paper has the collection of words observed in Fedaralist Paper 49. We would like to know whether $P(H \mid D)$ is larger than $P(M \mid D)$. This is equivalent to trying to decide if $P(H \mid D)/P(M \mid D)$ is larger than 1.

The event $(D \mid H)$ is a multinomial parameterized by the values p. The event $(D \mid M)$ is also a multinomial, this time parameterized by the values q.

Example using *multinomial distributions* to estimate who wrote the Federalist Paper 49, highlighting log numerical stability tricks.

Using Bayes Rule we can simplify the desired probability.

$$\begin{aligned}\frac{P(H \mid D)}{P(M \mid D)} &= \frac{\frac{P(D|H)P(H)}{P(D)}}{\frac{P(D|M)P(M)}{P(D)}} = \frac{P(D \mid H)P(H)}{P(D \mid M)P(M)} = \frac{P(D \mid H)}{P(D \mid M)} \\ &= \frac{\binom{n}{c_1,c_2,\dots,c_m}\prod_i p_i^{c_i}}{\binom{n}{c_1,c_2,\dots,c_m}\prod_i q_i^{c_i}} = \frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}\end{aligned}$$

This seems great! We have our desired probability statement expressed in terms of a product of values we have already estimated. However, when we plug this into a computer, both the numerator and denominator come out to be zero. The product of many numbers close to zero is too hard for a computer to represent. To fix this problem, we use a standard trick in computational probability: We apply a log to both sides and apply some basic rules of logs:

$$\begin{aligned}\log\left(\frac{P(H \mid D)}{P(M \mid D)}\right) &= \log\left(\frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}\right) \\ &= \log\left(\prod_i p_i^{c_i}\right) - \log\left(\prod_i q_i^{c_i}\right) \\ &= \sum_i \log\left(p_i^{c_i}\right) - \sum_i \log\left(q_i^{c_i}\right) \\ &= \sum_i c_i \log(p_i) - \sum_i c_i \log(q_i)\end{aligned}$$

This expression is ''numerically stable'' and my computer returned that the answer was a negative number. We can use exponentiation to solve for $P(H \mid D)/P(M \mid D)$. Since the exponent of a negative number is a number smaller than 1, this implies that $P(H \mid D)/P(M \mid D)$ is smaller than 1. As a result, we conclude that Madison was more likely to have written Fedaralist Paper 49.

12 *Independent Random Variables*

From notes by by Chris Piech and Lisa Yan.

12.1 *Independence with Multiple Discrete Random Variables*

Same ideas, different notation.

Two discrete random variables X and Y are called *independent* if:

$$P(X = x, Y = y) = P(X = x)P(Y = y) \text{ for all } x, y \tag{12.1}$$

$$p_{X,Y}(x, y) = p_X(x)p_Y(y) \text{ for all } x, y \tag{12.2}$$

Intuitively: knowing the value of X tells us nothing about the distribution of Y. If two variables are not independent, they are called *dependent*.[1] This is a similar conceptually to independent events, but we are dealing with multiple *variables*. Make sure to keep your events and variables distinct.

12.2 *Symmetry of Independence*

Independence is symmetric. That means that if random variables X and Y are independent, X is independent of Y and Y is independent of X. This claim may seem meaningless but it can be very useful. Imagine a sequence of events $X_1, X_2, \ldots$. Let A_i be the event that X_i is a "record value" (e.g., it is larger than all previous values). Is A_{n+1} independent of A_n? It is easier to answer that A_n is independent of A_{n+1}. By symmetry of independence both claims must be true.

12.3 *Sums of Independent Random Variables*

Independent Binomials with Equal p: For any two Binomial random variables with the same "success" probability: $X \sim \text{Bin}(n_1, p)$ and $Y \sim \text{Bin}(n_2, p)$ the sum of those two random variables is another binomial: $X + Y \sim \text{Bin}(n_1 + n_2, p)$. This does not hold when the two distributions have different parameters p.

This holds in the general case, let $X_i \sim \text{Bin}(n_i, p)$ for X_i independent variables for $i = 1, \ldots, n$:

$$\sum_{i=1}^{n} X_i \sim \text{Bin}\left(\sum_{i=1}^{n} n_i, p\right) \tag{12.3}$$

[1] To prove *dependence*, simply find a counterexample.

Example 12.1. *Independence* of two discrete Binomial random variables

Let N be the number of requests to a web server/day and that $N \sim \text{Poi}(\lambda)$. Each request comes from a human (probability = p) or from a ''bot'' (probability = $(1-p)$), independently. Define X to be the number of requests from humans/day and Y to be the number of requests from bots/day.

Since requests come in independently, the probability of X conditioned on knowing the number of requests is a Binomial. Specifically, conditioned:

$$\begin{aligned}(X \mid N) &\sim \text{Bin}(N, p)\\ (Y \mid N) &\sim \text{Bin}(N, 1-p)\end{aligned}$$

Calculate the probability of getting exactly i human requests and j bot requests. Start by expanding using the chain rule:

$$P(X=i, Y=j) = P(X=i, Y=j \mid X+Y=i+j)P(X+Y=i+j)$$

We can calculate each term in this expression:

$$\begin{aligned}P(X=i, Y=j \mid X+Y=i+j) &= \binom{i+j}{i} p^i(1-p)^j\\ P(X+Y=i+j) &= e^{-\lambda}\frac{\lambda^{i+j}}{(i+j)!}\end{aligned}$$

Now we can put those together and simplify:

$$P(X=i, Y=j) = \binom{i+j}{i} p^i(1-p)^j e^{-\lambda}\frac{\lambda^{i+j}}{(i+j)!}$$

As an exercise you can simplify this expression into two independent Poisson distributions.

Independent Poissons: For any two Poisson random variables: $X \sim \text{Poi}(\lambda_1)$ and $Y \sim \text{Poi}(\lambda_2)$ the sum of those two random variables is another Poisson: $X + Y \sim \text{Poi}(\lambda_1 + \lambda_2)$. This holds even if λ_1 is not the same as λ_2.

This holds in the general case, let $X_i \sim \text{Poi}(\lambda_i)$ for X_i independent variables for $i = 1, \ldots, n$:

$$\sum_{i=1}^{n} X_i \sim \text{Poi}\left(\sum_{i=1}^{n} \lambda_i\right) \tag{12.4}$$

Example 12.2. *Independence* of two discrete Poisson random variables.

Let's say we have two independent random Poisson variables for requests received at a web server in a day: X = number of requests from humans/day, $X \sim \text{Poi}(\lambda_1)$ and Y = number of requests from bots/day, $Y \sim \text{Poi}(\lambda_2)$. Since the convolution of Poisson random variables is also a Poisson, we know that the total number of requests $(X + Y)$ is also a Poisson: $(X + Y) \sim \text{Poi}(\lambda_1 + \lambda_2)$. What is the probability of having k human requests on a particular day given that there were n total requests?

$$\begin{aligned} P(X = k \mid X + Y = n) &= \frac{P(X = k, Y = n - k)}{P(X + Y = n)} = \frac{P(X = k)P(Y = n - k)}{P(X + Y = n)} \\ &= \frac{e^{-\lambda_1}\lambda_1^k}{k!} \cdot \frac{e^{-\lambda_2}\lambda_2^{n-k}}{(n-k)!} \cdot \frac{n!}{e^{-(\lambda_1+\lambda_2)}(\lambda_1 + \lambda_2)^n} \\ &= \binom{n}{k}\left(\frac{\lambda_1}{\lambda_1 + \lambda_2}\right)^k \left(\frac{\lambda_2}{\lambda_1 + \lambda_2}\right)^{n-k} \\ \therefore (X \mid X + Y = n) &\sim \text{Bin}\left(n, \frac{\lambda_1}{\lambda_1 + \lambda_2}\right) \end{aligned}$$

12.4 *Convolution: Sum of Independent Random Variables*

So far, we have had it easy: If our two independent random variables are both Poisson, or both Binomial with the same probability of success, then their sum has a nice, closed form. In the general case, however, the distribution of two independent random variables can be calculated as a *convolution* of probability distributions.

For two independent random variables, you can calculate the CDF or the PDF of the sum of two random variables using the following formulas:

$$F_{X+Y}(n) = P(X+Y \leq n) = \sum_{k=-\infty}^{\infty} F_X(k) F_Y(n-k) \tag{12.5}$$

$$p_{X+Y}(n) = \sum_{k=-\infty}^{\infty} p_X(k) p_Y(n-k) \tag{12.6}$$

For independent *discrete* random variables, the convolution of p_X and p_Y (in different notation):

$$P(X+Y=n) = \sum_k P(X=k)P(Y=n-k)$$

Most importantly, convolution is the process of finding the sum of the random variables themselves, and not the process of adding together probabilities.

Example 12.3. Proving that the sum of two independent Poisson random variables is also a Poisson.

Let's go about proving that the sum of two independent Poisson random variables is also Poisson. Let $X \sim \text{Poi}(\lambda_1)$ and $Y \sim \text{Poi}(\lambda_2)$ be two independent random variables, and $Z = X + Y$. What is $P(Z = n)$?

$$\begin{aligned}
P(Z=n) &= P(X+Y=n) \\
&= \sum_{k=-\infty}^{\infty} P(X=k)P(Y=n-k) && \text{(Convolution)} \\
&= \sum_{k=0}^{n} P(X=k)P(Y=n-k) && \text{(Range of } X \text{ and } Y\text{)} \\
&= \sum_{k=0}^{n} e^{-\lambda_1} \frac{\lambda_1^k}{k!} e^{-\lambda_2} \frac{\lambda_2^{n-k}}{(n-k)!} && \text{(Poisson PMF)} \\
&= e^{-(\lambda_1+\lambda_2)} \sum_{k=0}^{n} \frac{\lambda_1^k \lambda_2^{n-k}}{k!(n-k)!} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{n!} \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} \lambda_1^k \lambda_2^{n-k} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{n!} (\lambda_1+\lambda_2)^n && \text{(Binomial theorem)}
\end{aligned}$$

Note that the Binomial Theorem (which we did not cover in this class, but is often used in contexts like expanding polynomials) says that for two numbers a and b and positive integer n, then: $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$

13 *Statistics of Multiple Random Variables*

From notes by by Chris Piech and Lisa Yan.

As you can imagine, reporting probability mass functions or distributions is often not ideal: We either have to find a common distribution that fits our experiment, or we have to report a probability table or a bar graph. In the single random variable case, we often report expectation or variance as *statistics* that characterize our randomness. A similar paradigm applies for the multiple random variable case! In this section, we discuss statistics of two random variables; in particular, (1) how to easily calculate the expectation of the sum of multiple random variables, and (2) how to report how two random variables vary with one another.

13.1 *Expectation with Multiple Random Variables*

Expectation over a joint distribution is not nicely defined because it is not clear how to compose the multiple variables. However, expectations over functions of random variables (for example sums or products) are nicely defined: $\mathbb{E}[g(X,Y)] = \sum_{x,y} g(x,y)p(x,y)$ for any function $g(X,Y)$. When you expand that result for the function $g(X,Y) = X + Y$ you get a beautiful result:

$$\begin{aligned}
\mathbb{E}[X+Y] = \mathbb{E}[g(X,Y)] &= \sum_{x,y} g(x,y)p(x,y) = \sum_{x,y}[x+y]p(x,y) && (13.1)\\
&= \sum_{x,y} xp(x,y) + \sum_{x,y} yp(x,y) && (13.2)\\
&= \sum_{x} x \sum_{y} p(x,y) + \sum_{y} y \sum_{x} p(x,y) && (13.3)\\
&= \sum_{x} xp(x) + \sum_{y} yp(y) && (13.4)\\
&= \mathbb{E}[X] + \mathbb{E}[Y] && (13.5)
\end{aligned}$$

This can be generalized to multiple variables:

$$\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] \tag{13.6}$$

Let's go back to our old friends—the Binomial and Negative Binomial RVs—and show how we could have derived expressions for their expectation.

13.2 *Expectation of Binomial*

First let's start with some practice with the sum of expectations of indicator variables. Let $Y \sim \text{Bin}(n, p)$, in other words if Y is a Binomial random variable. We can express Y as the sum of n Bernoulli random indicator variables $X_i \sim \text{Ber}(p)$. Since X_i is a Bernoulli, $\mathbb{E}[X_i] = p$

$$Y = X_1 + X_2 + \cdots + X_n = \sum_{i=1}^{n} X_i \tag{13.7}$$

Let's formally calculate the expectation of Y:

$$\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i}^{n} X_i\right] = \sum_{i}^{n} \mathbb{E}[X_i] = \mathbb{E}[X_0] + \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_n] = np \tag{13.8}$$

13.3 *Expectation of Negative Binomial*

Recall that a Negative Binomial is a random variable that semantically represents the number of trials until r successes. Let $Y \sim \text{NegBin}(r, p)$.

Let X_i = # trials to get success after the $(i-1)$-th success. We can then think of each X_i as a Geometric random variable: $X_i \sim \text{Geo}(p)$. Thus, $\mathbb{E}[X_i] = 1/p$. We can express Y as:

$$Y = X_1 + X_2 + \cdots + X_r = \sum_{i=1}^{r} X_i \tag{13.9}$$

Let's formally calculate the expectation of Y:

$$\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i=1}^{r} X_i\right] = \sum_{i=1}^{r} \mathbb{E}[X_i] = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \cdots + \mathbb{E}[X_r] = \frac{r}{p} \tag{13.10}$$

Coupon Collector's Problem: There are several versions of the coupon collector's problem in probability theory, but the most common formulation is as follows: You would like to collect coupons from cereal boxes, but you must purchase a box of cereal to open and discover what coupon type you have. More formally, suppose you buy n boxes of cereal, and there are k different types of coupons. For each box you buy, you "collect" a coupon of type i. What is the expected number of boxes that you must purchase until you have at least one coupon of each type?

How does this relate to computer science? Suppose you are a big cloud provider, and you have to service n web requests with a limited number of k servers. Each web request is a request to server i. What is the expected number of utilized servers after n requests?

Example 13.1. The coupon collector's problem setup.

13.4 *Expectations of Products Lemma*

We know that the expectation of the sum of two random variables is equal to the sum of the expectations of the two variables. However, the expectation of the product of two random variables only has a nice decomposition in the case where the random variables are independent of one another.

$$\mathbb{E}[g(X)h(Y)] = \mathbb{E}[g(X)]\mathbb{E}[h(Y)] \qquad \text{if } X \text{ and } Y \text{ are independent}$$

Here's a proof for independent discrete random variables X and Y. If you would like to prove this for independent continuous random variables, just interchange the summations with integrals.

$$\begin{aligned}
\mathbb{E}[g(X)h(Y)] &= \sum_y \sum_x g(x)h(y)p_{X,Y}(x,y) \\
&= \sum_y \sum_x g(x)h(y)p_X(x)p_Y(y) \\
&= \sum_y \left(h(y)p_Y(y) \sum_x g(x)p_X(x) \right) \\
&= \left(\sum_x g(x)p_X(x) \right) \left(\sum_y h(y)p_Y(y) \right) \\
&= \mathbb{E}[g(X)]\mathbb{E}[h(Y)]
\end{aligned}$$

Example 13.2. Hash tables as a variation of the coupon collector's problem.

Problem: Yes, hash table problems can be a variation of the coupon collector's problem! Consider a hash table with k buckets. You hash each string to bucket i. What is the expected number of strings to hash until each bucket has at least 1 string?

Solution: Define Y as the number of strings to hash until each bucket has at least 1 string. We want to compute $\mathbb{E}[Y]$. Let us also define Y_i to be the number of trials (strings) until the next success, after we've seen our i-th success. For example, Y_0 is the number of strings hashed until our first hash into an empty bucket (we start with k empty buckets), Y_1 is the number of additional strings to hash until we hash into an empty bucket (we have 1 non-empty bucket and $k-1$ empty buckets, etc.). In the general case, we have i non-empty buckets and $k-i$ empty buckets after the i-th success, and we are successful if we hash a string to one of the $k-i$ empty buckets. The probability of success p is then $p_i = \frac{k-i}{k}$. With this definition of Y_i, let $Y_i \sim \text{Geo}(p)$, and $\mathbb{E}[Y_i] = \frac{1}{p_i} = \frac{k}{k-i}$.

Note that $Y = Y_0 + Y_1 + Y_2 + \cdots + Y_{n-1}$. We can show the following:

$$\begin{aligned}
\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i=0}^{n} Y_i\right] &= \sum_{i=0}^{n} \mathbb{E}[Y_i] \\
&= \sum_{i=0}^{n} \frac{k}{k-i} = \frac{k}{k} + \frac{k}{k-1} + \frac{k}{k-2} + \cdots + \frac{k}{1} \\
&= k\left[\frac{1}{k} + \frac{1}{k-1} + \cdots + 1\right] = O(k \log k)
\end{aligned}$$

13.5 *Covariance and Correlation*

Consider the two multivariate distributions shown in figures 13.1 and 13.2. In both images I have plotted one thousand samples drawn from the underlying joint distribution. Clearly the two distributions are different. However, the mean and variance are the same in both the x and the y dimension. What is different?

Covariance is a quantitative measure of the extent to which the deviation of one variable from its mean matches the deviation of the other from its mean. It is a mathematical relationship that is defined as:

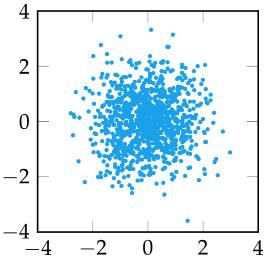


Figure 13.1. Two independent normal random variables.

$$\mathrm{Cov}(X,Y) = \mathbb{E}[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])] \tag{13.11}$$

That is a little hard to wrap your mind around (but worth pushing on a bit). The outer expectation will be a weighted sum of the inner function evaluated at a particular (x,y) weighted by the probability of (x,y). If x and y are both above their respective means, or if x and y are both below their respective means, that term will be positive. If one is above its mean and the other is below, the term is negative. If the weighted sum of terms is positive, the two random variables will have a positive correlation. We can rewrite the above equation to get an equivalent equation:

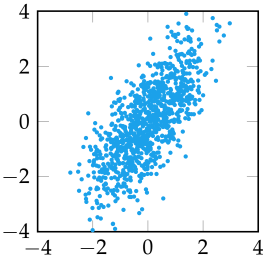


Figure 13.2. Two normal random variables with the same mean and variance as figure 13.1.

$$\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[Y]\mathbb{E}[X] \tag{13.12}$$

Using this equation (and the product lemma) is it easy to see that if two random variables are independent their covariance is 0. The reverse is *not* true in general.

$$\begin{aligned}
\mathrm{Cov}(X,Y) &= \mathbb{E}[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])] \\
&= \mathbb{E}[XY - X\mathbb{E}[Y] - \mathbb{E}[X]Y + \mathbb{E}[X]\mathbb{E}[Y]] \\
&= \mathbb{E}[XY] - \mathbb{E}[X\mathbb{E}[Y]] - \mathbb{E}[\mathbb{E}[X]Y] + \mathbb{E}[\mathbb{E}[X]\mathbb{E}[Y]] \\
&= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y] + \mathbb{E}[X]\mathbb{E}[Y] \\
&= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]
\end{aligned}$$

13.6 Properties of Covariance

Say that X and Y are arbitrary random variables:

$$\text{Cov}(X, Y) = \text{Cov}(Y, X) \tag{13.13}$$

$$\text{Cov}(X, X) = \mathbb{E}[X^2] - \mathbb{E}[X]\mathbb{E}[X] = \text{Var}(X) \tag{13.14}$$

$$\text{Cov}\left(\sum_i X_i, \sum_j Y_j\right) = \sum_i \sum_j \text{Cov}(X_i, Y_j) \tag{13.15}$$

$$\text{Cov}(aX + b, Y) = a\,\text{Cov}(X, Y) \tag{13.16}$$

Let $X = X_1 + X_2 + \cdots + X_n$ and let $Y = Y_1 + Y_2 + \cdots + Y_m$. The covariance of X and Y is:

$$\text{Cov}(X, Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} \text{Cov}(X_i, Y_j) \tag{13.17}$$

$$\text{Cov}(X, X) = \text{Var}(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Cov}(X_i, X_j) \tag{13.18}$$

$$= \sum_{i=1}^{n} \text{Var}(X_i) + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{Cov}(X_i, X_j) \tag{13.19}$$

That last property gives us a third way to calculate variance. You could use this definition to calculate the variance of the binomial.

For any random variables X and Y, the variance of the sum necessarily includes covariance (unless the two random variables are independent):

$$\begin{aligned}\text{Var}(X + Y) &= \text{Cov}(X + Y, X + Y) \\ &= \text{Cov}(X, X) + \text{Cov}(X, Y) + \text{Cov}(Y, X) + \text{Cov}(Y, Y) \\ &= \text{Var}(X) + 2\,\text{Cov}(X, Y) + \text{Var}(Y)\end{aligned}$$

More generally:

$$\text{Var}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \text{Var}(X_i) + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{Cov}(X_i, X_j)$$

But for *independent* X and Y:

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] \tag{13.20}$$

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) \tag{13.21}$$

13.7 Correlation

Covariance is interesting because it is a quantitative measurement of the relationship between two variables. Correlation between two random variables, $\rho(X,Y)$ is the covariance of the two variables normalized by the variance of each variable. This normalization cancels the units out and normalizes the measure so that it is always in the range $[-1,1]$:

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{13.22}$$

$$\sigma_X = \sqrt{\mathrm{Var}(X)} \iff \sigma_X^2 = \mathrm{Var}(X)$$
$$\sigma_Y = \sqrt{\mathrm{Var}(Y)} \iff \sigma_Y^2 = \mathrm{Var}(Y)$$

Correlation measures linearity between X and Y.

$$\begin{aligned}
\rho(X,Y) &= 1 && Y = aX + b \text{ where } a = \sigma_Y/\sigma_X \\
\rho(X,Y) &= -1 && Y = aX + b \text{ where } a = -\sigma_Y/\sigma_X \\
\rho(X,Y) &= 0 && \text{absence of linear relationship}
\end{aligned}$$

If $\rho(X,Y) = 0$ we say that X and Y are *uncorrelated*. If two varaibles are independent,[1] then their correlation will be 0. However, it doesn't go the other way. A correlation of 0 does not imply independence.

When people use the term correlation, they are actually referring to a specific type of correlation called ''Pearson'' correlation. It measures the degree to which there is a linear relationship between the two variables. An alternative measure is ''Spearman'' correlation which has a formula almost identical to your regular correlation score, with the exception that the underlying random variables are first transformed into their rank.

Conditional statements regarding independence and correlation:

$$\text{independence} \implies \text{no correlation}$$
$$\text{correlation} \implies \text{dependence}$$

But, ''correlation does not imply causation'':

$$\text{dependence} \not\Longrightarrow \text{correlation}$$
$$\text{no correlation} \not\Longrightarrow \text{independence}$$

Somewhat paradoxically, though independence implies zero covariance, zero covariance does not imply independence. Consider the following example: you have a discrete random variable X with PMF $P(X = x) = 1/3$ for $x \in \{-1, 0, 1\}$ and we then define another random variable $Y = X^2$. Clearly X and Y are not independent, but do they have zero covariance? We'll learn in class that we tend to use a measure called *correlation* to indicate zero covariance, because the units of covariance don't matter so much as the sign.

Example 13.3. Zero covariance does not imply independence (directionality matters).

[1] If X and Y are independent random variables, then:

$$\begin{aligned}
\mathrm{Cov}(X,Y) &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \\
&= \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y] \\
&= 0
\end{aligned}$$

14 Conditional Expectation

14.1 Conditional Distributions

From notes by Chris Piech and Lisa Yan.

Before we looked at conditional probabilities for events. Here we formally go over conditional probabilities for random variables. The equations for the discrete case is an intuitive extension of our understanding of conditional probability:

Discrete: The conditional probability mass function (PMF) for the discrete case:

$$p_{X|Y}(x \mid y) = P(X = x \mid Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{p_{X,Y}(x, y)}{p_Y(y)} \tag{14.1}$$

The conditional cumulative density function (CDF) for the discrete case:

$$F_{X|Y}(a \mid y) = P(X \leq a \mid Y = y) = \frac{\sum_{x \leq a} p_{X,Y}(x, y)}{p_Y(y)} = \sum_{x \leq a} p_{X|Y}(x \mid y) \tag{14.2}$$

14.2 Conditional Expectation

We have gotten to know a kind and gentle soul, conditional probability. And we know another funky fool, expectation. Let's get those two crazy kids to play together.

Let X and Y be jointly discrete random variables. We define the *conditional expectation* of X given $Y = y$ to be:

$$\mathbb{E}[X \mid Y = y] = \sum_x x p_{X|Y}(x \mid y) \tag{14.3}$$

14.3 Properties of Conditional Expectation

Here are some helpful, intuitive properties of conditional expectation:

$$\begin{aligned}
\mathbb{E}[g(X) \mid Y = y] &= \sum_x g(x) p_{X|Y}(x \mid y) && \text{if } X \text{ and } Y \text{ are discrete} \\
\mathbb{E}[g(X) \mid Y = y] &= \int_{-\infty}^{\infty} g(x) f_{X|Y}(x \mid y) dx && \text{if } X \text{ and } Y \text{ are continuous} \\
\mathbb{E}\left[\sum_{i=1}^{n} X_i \mid Y = y\right] &= \sum_{i=1}^{n} \mathbb{E}[X_i \mid Y = y]
\end{aligned}$$

14.4 Law of Total Expectation

The law of total expectation states that:

$$\mathbb{E}[\mathbb{E}[X \mid Y]] = \mathbb{E}[X] \tag{14.4}$$

What?! How is that a thing? Check out this proof:

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[X \mid Y]] = \mathbb{E}[g(Y)] &= \sum_y \mathbb{E}[X \mid Y = y] P(Y = y) \\
&= \sum_y \sum_x x P(X = x \mid Y = y) P(Y = y) \\
&= \sum_y \sum_x x P(X = x, Y = y) = \sum_x \sum_y x P(X = x, Y = y) \\
&= \sum_x x \sum_y P(X = x, Y = y) \\
&= \sum_x x P(X = x) \\
&= \mathbb{E}[X]
\end{aligned}$$

If we only have a conditional PMF of X on some discrete variable Y, we can compute $\mathbb{E}[X]$ as follows:

1. Compute conditional expectation of X given some value of $Y = y$
2. Repeat step 1 for all values of $Y = y$
3. Compute a weighted sum (where weights are $P(Y = y)$)

Example 14.1. *Conditional expectation* of dice rolls.

You roll two 6-sided dice D_1 and D_2. Let $S = D_1 + D_2$.

- What is $\mathbb{E}[S \mid D_2 = 6]$?

$$\begin{aligned}\mathbb{E}[S \mid D_2 = 6] &= \sum_x xP(S = x \mid D_2 = 6) \\ &= \frac{1}{6}(7 + 8 + 9 + 10 + 11 + 12) = \frac{57}{6} = 9.5\end{aligned}$$

  This makes intuitive sense since $6 + \mathbb{E}[\text{value of } D_1] = 6 + 3.5$.

- What is $\mathbb{E}[S \mid D_2 = d_2]$, where $d_2 = 1, \ldots, 6$? Note that $S = D_1 + D_2$ and that D_1 and D_2 are independent.

$$\begin{aligned}\mathbb{E}[S \mid D_2 = d_2] &= \mathbb{E}[D_1 + D_2 \mid D_2 = d_2] = \mathbb{E}[D_1 + d_2 \mid D_2 = d_2] \\ &= d_2 + \mathbb{E}[D_1 \mid D_2 = d_2] \\ &\qquad\qquad (d_2 \text{ is a constant with respect to } D_1) \\ &= d_2 + \sum_{d_1} d_1 P(D_1 = d_1 \mid D_2 = d_2) \\ &= d_2 + \sum_{d_1} d_1 P(D_1 = d_1) \qquad (D_1, D_2 \text{ are independent}) \\ &= d_2 + 3.5\end{aligned}$$

  Note that $\mathbb{E}[S \mid D_2 = d_2]$ depends on the value d_2. In other words, $\mathbb{E}[S \mid D_2]$ is a function of the random variable D_2.

Example 14.2. Expected return of the recursive function `recurse`.

Consider the following code with random numbers:

```
function recurse()
    x = rand(1:3) # Equally likely values
    if x == 1     return 3
    elseif x == 2 return 5 + recurse()
    else          return 7 + recurse() end
end
```

Let Y = value returned by `recurse`. What is $\mathbb{E}[Y]$? In other words, what is the expected return value. Note that this is the exact same approach as calculating the expected run time.

$$\begin{aligned}\mathbb{E}[Y] = \mathbb{E}[Y \mid X = 1]P(X = 1)\\ + \mathbb{E}[Y \mid X = 2]P(X = 2)\\ + \mathbb{E}[Y \mid X = 3]P(X = 3)\end{aligned}$$

First lets calculate each of the conditional expectations:

$$\begin{aligned}\mathbb{E}[Y \mid X = 1] &= 3\\ \mathbb{E}[Y \mid X = 2] &= \mathbb{E}[5 + Y] = 5 + \mathbb{E}[Y]\\ \mathbb{E}[Y \mid X = 3] &= \mathbb{E}[7 + Y] = 7 + \mathbb{E}[Y]\end{aligned}$$

Now we can plug those values into the equation. Note that the probability of X taking on 1, 2, or 3 is $1/3$:

$$\begin{aligned}\mathbb{E}[Y] &= \mathbb{E}[Y \mid X = 1]P(X = 1)\\ &\quad + \mathbb{E}[Y \mid X = 2]P(X = 2)\\ &\quad + \mathbb{E}[Y \mid X = 3]P(X = 3)\\ &= 3(1/3) + (5 + \mathbb{E}[Y])(1/3) + (7 + \mathbb{E}[Y])(1/3)\\ &= 15\end{aligned}$$

Example 14.3. How many software engineers do you have to interview before hiring the optimal?

You are interviewing n software engineer candidates and will hire only 1 candidate. All orderings of candidates are equally likely. Right after each interview you must decide to hire or not hire. You can not go back on a decision. At any point in time you can know the relative ranking of the candidates you have already interviewed.

The strategy that we propose is that we interview the first k candidates and reject them all. Then you hire the next candidate that is better than all of the first k candidates. What is the probability that the best of all the n candidates is hired for a particular choice of k? Let's denote that result $P_k(\text{best})$. Let X be the position in the ordering of the best candidate:

$$\begin{aligned}
P_k(\text{best}) &= \sum_{i=1}^{n} P_k(\text{best} \mid X=i)P(X=i) \\
&= \frac{1}{n}\sum_{i=1}^{n} P_k(\text{best} \mid X=i) \qquad \text{(since each position is equally likely)}
\end{aligned}$$

What is $P_k(\text{best} \mid X=i)$? If $i \leq k$ then the probability is 0 because the best candidate will be rejected without consideration. Sad times. Otherwise we will chose the best candidate, who is in position i, only if the best of the first $i-1$ candidates is among the first k interviewed. If the best among the first $i-1$ is not among the first k, that candidate will be chosen over the true best. Since all orderings are equally likely the probability that the best among the $i-1$ candidates is in the first k is $\dfrac{k}{i-1}$, if $i > k$. Plugging back in:

$$\begin{aligned}
P_k(\text{best}) &= \frac{1}{n}\sum_{i=1}^{n} P_k(\text{best} \mid X=i) \\
&= \frac{1}{n}\sum_{i=k+1}^{n} \frac{k}{i-1} \qquad \text{(since we know } P_k(\text{best} \mid X=i)\text{)} \\
&\approx \frac{1}{n}\int_{i=k+1}^{n} \frac{k}{i-1}di \qquad \text{(by Riemann Sum approximation)} \\
&= \frac{k}{n}\ln(i=1)\Big|_{k+1}^{n} = \frac{k}{n}\ln\frac{n-1}{k} \approx \frac{k}{n}\ln\frac{n}{k}
\end{aligned}$$

If we think of $P_k(\text{best}) = \frac{k}{n}\ln\frac{n}{k}$ as a function of k we can take find the value of k that optimizes it by taking its derivative and setting it equal to 0. The optimal value of k is n/e. Where e is Euler's number.

15 Inference

From notes by Chris Piech and Lisa Yan.

At this point in CS109 we have developed tools for analytically solving for probabilities. We can calculate the likelihood of random variables taking on values, even if they are interacting with other random variables (which we have called multivariate models, or we say the random variables are jointly distributed). We have also started to study samples and sampling.

As a capstone for this part of the class I would like to consider the task of *general inference* in the world of disease prediction. A website, WebMd Symptom Checker, exemplifies our task. They have built a probabilistic model with random variables which roughly fall under three categories: symptoms, risk factors and diseases. For any combination of observed symptoms and risk factors, they can calculate the probability of any disease. For example, they can calculate the probability that I have influenza given that I am a 21-year-old female who has a fever and who is tired: $P(I = 1 \mid A = 21, G = 1, T = 1, F = 1)$. Or they could calculate the probability that I have a cold given that I am a 30-year-old with a runny nose: $P(C = 1 \mid A = 30, R = 1)$. At first blush this might not seem difficult. But as we dig deeper we will realize just how hard it is. There are two challenges: (1) sufficiently specifying the probabilistic model and (2) calculating any desired probability.

15.1 Bayesian Networks

Before we jump into how to solve probability (aka inference) questions, let's take a moment to go over how an expert doctor could specify the relationship between so many random variables. Ideally we could have our expert sit down and specify the entire ''joint distribution'' (see the first lecture on multivariate models). She could do so either by writing a single equation that relates all the variables (which is as impossible as it sounds), or she could come up with a joint distribution table where she specifies the probability of any possible combination of assignments to variables. It turns out that is not feasible either. Why? Imagine there are $N = 100$ binary random variables in our WebMD model. Our expert doctor would have to specify a probability for each of the $2^N > 10^{30}$ combinations of assignments to those variables, which is approaching the number of atoms in the universe. Thankfully, there is a better way. We can simplify our task if we know the *generative* process that creates a joint assignment. Based on the generative process we can make a data structure known as a *Bayesian network*. Here are two networks of random variables for diseases:

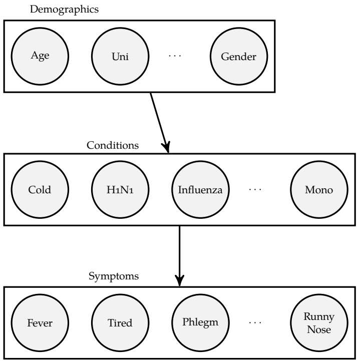


Figure 15.1. Full disease model where flow of influence is directed.

For diseases, the flow of influence is directed. The states of "demographic" random variables influence whether someone has particular "conditions", which influence whether someone shows particular "symptoms". On the right is a simple model with only four random variables. Though this is a less interesting model it is easier to understand when first learning Bayesian networks. Being in university (binary) influences whether or not someone has influenza (binary). Having influenza influences whether or not someone has a fever (binary) and the state of university and influenza influences whether or not someone feels tired (also binary).

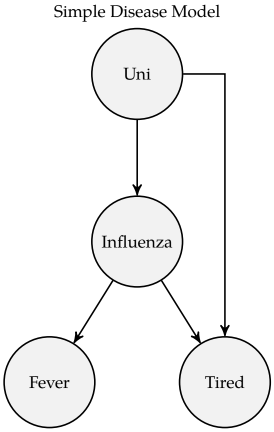


Figure 15.2. *Bayesian network* simple disease model.

In a Bayesian network, an arrow from random variable X to random variable Y articulates our assumption that X directly influences the likelihood of Y. We say that X is a *parent* of Y. To fully define the Bayesian network we **must** provide a way to compute the probability of each random variable X_i conditioned on knowing the value of all their parents: $P(X_i = k \mid \text{parents of } X_i \text{ take on specified values})$. Here is a concrete example of what needs to be defined for the simple disease model. Recall that each of the random variables is binary:

$$P(\text{Uni} = 1) = 0.8$$

$$P(\text{Influenza} = 1 \mid \text{Uni} = 1) = 0.2 \qquad P(\text{Fever} = 1 \mid \text{Influenza} = 1) = 0.9$$

$$P(\text{Influenza} = 1 \mid \text{Uni} = 0) = 0.1 \qquad P(\text{Fever} = 1 \mid \text{Influenza} = 0) = 0.05$$

$$P(\text{Tired} = 1 \mid \text{Uni} = 0, \text{Influenza} = 0) = 0.1$$

$$P(\text{Tired} = 1 \mid \text{Uni} = 0, \text{Influenza} = 1) = 0.9$$

$$P(\text{Tired} = 1 \mid \text{Uni} = 1, \text{Influenza} = 0) = 0.8$$

$$P(\text{Tired} = 1 \mid \text{Uni} = 1, \text{Influenza} = 1) = 1.0$$

Let's put this in programming terms. All that we need to do in order to code up a Bayesian network is to define a function: `getProbXi(i, k, parents)` which returns the probability that X_i (the random variable with index `i`) takes on the value k given a value for each of the parents of X_i encoded by `parents`: $P(X_i = k_i \mid \text{parents of } X_i \text{ take on specified values})$

Deeper understanding: The reason that a Bayes Net is so useful is that the "joint" probability can be expressed in exponentially less space as the product of the probabilities of each random variable conditioned on its parents! Without loss of generality, let X_i refer to the ith random variable (such that if X_i is a parent of X_j then $i < j$):

$$\begin{aligned} P(\text{joint}) &= P(X_1 = k_1, \ldots, X_n = k_n) \\ &= \prod_i P(X_i = k_i \mid \text{parents of } X_i \text{ take on specified values}) \end{aligned} \tag{15.1}$$

Using the chain rule we can decompose the joint probability. To make the following math easier to digest I am going to use k_i as shorthand for the event that $X_i = k_i$:

$$\begin{aligned} P(k_1, \ldots, k_n) &= P(k_n \mid k_{n-1}, \ldots, k_1) P(k_{n-1} \mid k_{n-2}, \ldots, k_1) \cdots P(k_2 \mid k_1) P(k_1) && \text{(chain rule)} \\ &= \prod_i P(k_i \mid k_{i-1}, \ldots, k_1) && \text{(change in notation)} \\ &= \prod_i P(k_i \mid \text{parents of } X_i \text{ take on their values}) && \text{(implied by Bayes Net)} \end{aligned}$$

The central assumption made is:

$$P(k_i \mid k_{i-1}, \ldots, k_1) = P(k_i \mid \text{parents of } X_i \text{ take on their values})$$

In other words, each random variable is conditionally independent of its non-descendants, given its parents.

In the next part of CS109 we are going to talk about how we could learn such probabilities from data. For now let's start with the (reasonable) assumption that an expert can write `getProbXi`. We haven't talked about continuous of multinomial random variables in Bayes Nets. None of the theory changes: the expert will just have to define `getProbXi` to handle more values of `k` than 0 or 1.

Great! We have a feasible way to define a large network of random variables. First challenge complete. However a Bayesian network is not very interesting to us unless we can use it to solve different conditional probability questions.

15.2 *General Inference via Sampling the Joint Distribution*

Now we have a reasonable way to specify the joint probability of a network of many random variables. Before we celebrate, realize that we still don't know how to use such a network to answer probability questions. There are many techniques for doing so. I am going to introduce you to one of the great ideas in probability for computer science: We can use *sampling* to solve inference questions on Bayesian networks. Sampling is frequently used in practice because it is relatively easy to understand and easy to implement.

As a warmup consider what it would take to sample an assignment to each of the random variables in our Bayes net. Such a sample is often called a *particle* (as in a particle of sand). To sample a particle, simply sample a value for each random variable one at a time based on the value of the random variable's parents. This means that if X_i is a parent of X_j, you will have to sample a value for X_i before you sample a value for X_j.

Let's work through an example of sampling a "particle" for the Simple Disease Model in the previous section:

1. Sample from $P(\text{Uni} = 1)$: $\text{Ber}(0.8)$.
   Sampled value for Uni is 1.

2. Sample from $P(\text{Influenza} = 1 \mid \text{Uni} = 1)$: $\text{Ber}(0.2)$.
   Sampled value for Influenza is 0.

3. Sample from $P(\text{Fever} = 1 \mid \text{Influenza} = 0)$: $\text{Ber}(0.05)$.
   Sampled value for Fever is 0.

4. Sample from $P(\text{Tired} = 1 \mid \text{Uni} = 1, \text{Influenza} = 0)$: $\text{Ber}(0.8)$.
   Sampled value for Tired is 0.

Thus the sampled particle is: $[\text{Uni} = 1, \text{Influenza} = 0, \text{Fever} = 0, \text{Tired} = 0]$. If we were to run the process again we would get a new particle (with likelihood determined by the joint probability).

Now our strategy is simple: we are going to generate N samples where N is in the hundreds of thousands (if not millions). Then we can compute probability queries by counting. Let $N(\mathbf{X} = \mathbf{k})$ be notation for the number of particles where random variables $\mathbf{X}$ take on values $\mathbf{k}$. Recall that the bold notation $\mathbf{X}$ means that $\mathbf{X}$ is a vector with one or more elements. By the "frequentist" definition of probability:

$$P(\mathbf{X} = \mathbf{k}) = \frac{N(\mathbf{X} = \mathbf{k})}{N}$$

Counting for the win! But what about conditional probabilities? Well using the definition of conditional probabilities, we can see it's still some pretty straightforward counting:

$$P(\mathbf{X} = \mathbf{a} \mid \mathbf{Y} = \mathbf{b}) = \frac{P(\mathbf{X} = \mathbf{a}, \mathbf{Y} = \mathbf{b})}{P(\mathbf{Y} = \mathbf{b})} = \frac{\frac{N(\mathbf{X}=\mathbf{a},\mathbf{Y}=\mathbf{b})}{N}}{\frac{N(\mathbf{Y}=\mathbf{b})}{N}} = \frac{N(\mathbf{X} = \mathbf{a}, \mathbf{Y} = \mathbf{b})}{N(\mathbf{Y} = \mathbf{b})}$$

Let's take a moment to recognize that this is straight-up fantastic. General inference based on analytic probability (math without samples) is hard even given a Bayesian network (if you don't believe me, try to calculate the probability of flu conditioning on one demographic and one symptom in the Full Disease Model). However if we generate enough samples we can calculate any conditional probability question by reducing our samples to the ones that are consistent with the condition $(\mathbf{Y} = \mathbf{b})$ and then counting how many of those are also consistent with the query $(\mathbf{X} = \mathbf{a})$. Here is the algorithm in code:

```
N = 10000
# query: the assignment to variables we want probabilities for
# condition: the assignments to variables we will condition on
function get_any_probability(query, condition)
    particles = generate_many_joint_samples(N)
    cond_particles = reject_non_consistent(particles, condition)
    K = count_consistent_samples(cond_particles, query)
    return K / length(cond_particles)
end
```

Algorithm 15.1. Sampling to get an approximate probability.

This algorithm is sometimes called *rejection sampling* because it works by generating many particles from the joint distribution and rejecting the ones that are not consistent with the set of assignments we are conditioning on. Of course this algorithm is an approximation, though with enough samples it often works out to be a very good approximation. However, in cases where the event we're conditioning on is rare enough that it doesn't occur after millions of samples are generated, our algorithm will not work. The last line of our code will result in a divide by 0 error. See the next section for solutions!

15.3 General Inference when Conditioning on Rare Events

Rejection sampling is a powerful technique that takes advantage of computational power. But it doesn't always work. In fact it doesn't work any time that the probability of the event we are conditioning is rare enough that we are unlikely to ever produce samples that exactly match the event. The simplest example is with continuous random variables. Consider the Simple Disease Model. Let's change Fever from being a binary variable to being a continuous variable. To do so the only thing we need to do is re-specify the likelihood of fever given assignments to its parents (influenza). Let's say that the likelihoods come from the normal PDF:

$$\begin{aligned}\text{if Influenza } &= 0\text{, then Fever} \sim \mathcal{N}(\mu = 98.3, \sigma = 0.7)\\ \therefore f(\text{Fever} = x) &= \frac{1}{\sqrt{2\pi \cdot 0.7}} e^{-\frac{(x-98.3)^2}{2 \cdot 0.7}}\\ \text{if Influenza } &= 1\text{, then Fever} \sim \mathcal{N}(\mu = 100.0, \sigma = 1.8)\\ \therefore f(\text{Fever} = x) &= \frac{1}{\sqrt{2\pi \cdot 1.8}} e^{-\frac{(x-100.0)^2}{2 \cdot 1.8}}\end{aligned}$$

Drawing samples (aka particles) is still straightforward. We apply the same process until we get to the step where we sample a value for the Fever random variable (in the example from the previous section that was step 3). If we had sampled a 0 for influenza we draw a value for fever from the normal for healthy adults (which has $\mu = 98.3$). If we had sampled a 1 for influenza we draw a value for fever from the normal for adults with the flu (which has $\mu = 100.0$). The problem comes in the ''rejection'' stage of sampling the joint distribution.

When we sample values for fever we get numbers with infinite precision (eg 100.819238 etc). If we condition on someone having a fever equal to 101 we would reject every single particle. Why? No particle will have exactly a fever of 101.

There are several ways to deal with this problem. One especially easy solution is to be less strict when rejecting particles. We could round all fevers to whole numbers.

There is an algorithm called *likelihood weighting* which sometimes helps, but which we don't cover in CS109. Instead, in class we talked about a new algorithm called Markov Chain Monte Carlo (MCMC) that allowed us to sample from the ''posterior'' probability: the distribution of random variables after (post) us fixing variables in the conditioned event. The version of MCMC we talked about is called *Gibbs Sampling*. While I don't require that students in CS109 know how to implement Gibbs Sampling, I wanted everyone to know that it exists and that it isn't beyond your capabilities. If you need to use it, you can learn it given the knowledge you have now.

MCMC does require more math than rejection sampling. For every random variable you will need to specify how to calculate the likelihood of assignments given the variable's: parents, children and parents of its children (a set of variables cozily called a ''blanket''). Want to learn more? Take CS221, CS228, or CS238!

Thoughts: While there are slightly-more-powerful ''general inference algorithms'' that you will get to learn in the future, it is worth recognizing that at this point we have reached an important milestone in CS109. You can take very complicated probability models (encoded as Bayesian networks) and can answer general inference queries on them. To get there we worked through the concrete example of predicting disease. While the WebMd website is great for home users, similar probability models are being used in thousands of hospitals around the world. As you are reading this general inference is being used to improve health care (and sometimes even save lives) for real human beings. That's some probability for computer scientists that is worth learning. Now there is just one last question for us to look into in CS109. What if we don't have an expert? Could we learn those probabilities from data?

16 Continuous Joint Distributions

From notes by Chris Piech and Lisa Yan.

Of course joint variables don't have to be discrete only, they can also be continuous. As an example: consider throwing darts at a dart board. Because a dart board is two dimensional, it is natural to think about the X location of the dart and the Y location of the dart as two random variables that are varying together (aka they are joint). However since x and y positions are continuous we are going to need new language to think about the likelihood of different places a dart could land. Just like in the non-joint case continuous is a little tricky because it isn't easy to think about the probability that a dart lands at a location defined to infinite precision. What is the probability that a dart lands at exactly $(X = 456.234231234122355, Y = 532.12344123456)$?

Lets build some intuition by first starting with discretized grids. On the left of the image above you could imagine where your dart lands is one of 25 different cells in a grid. We could reason about the probabilities now! But we have lost all nuance about how likelihood is changing within a given cell. If we make our cells smaller and smaller we eventually will get a second derivative of probability: once again a probability density function. If we integrate under this joint-density function in both the x and y dimension we will get the probability that x takes on the values in the integrated range and y takes on the values in the integrated range! Random variables X and Y are *jointly continuous* if there exists a probability density function (PDF) $f_{X,Y}$ such that:

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = \int_{a1}^{a2} \int_{b1}^{b2} f_{X,Y}(x, y) dy\, dx \tag{16.1}$$

Using the PDF we can compute marginal probability densities:

$$f_X(a) = \int_{-\infty}^{\infty} f_{X,Y}(a,y)dy \tag{16.2}$$

$$f_Y(b) = \int_{-\infty}^{\infty} f_{X,Y}(x,b)dx \tag{16.3}$$

16.1 Independence with Multiple RVs (Continuous Case)

Two continuous random variables X and Y are called *independent* if:

$$P(X \leq a, Y \leq b) = P(X \leq a)P(Y \leq b) \qquad \text{for all } a, b$$

This can be stated equivalently as:

$$F_{X,Y}(a,b) = F_X(a)F_Y(b) \qquad \text{for all } a, b$$
$$f_{X,Y}(a,b) = f_X(a)f_Y(b) \qquad \text{for all } a, b$$

$$\begin{aligned} f_{X,Y}(x,y) &= \frac{\partial^2}{\partial x \partial y} F_{X,Y}(x,y) \\ &= \frac{\partial^2}{\partial x \partial y} F_X(x)F_Y(y) \\ &= \frac{\partial}{\partial x}\frac{\partial}{\partial y} F_X(x)F_Y(y) \\ &= \frac{\partial}{\partial x} F_X(x) \frac{\partial}{\partial y} F_Y(y) \\ &= f_X(x)f_Y(y) \end{aligned}$$

More generally, if you can factor the joint density function, then your continuous random variables are independent:

$$f_{X,Y}(x,y) = g(x)h(y) \qquad \text{where } -\infty < x, y < \infty$$

16.2 Joint CDFs

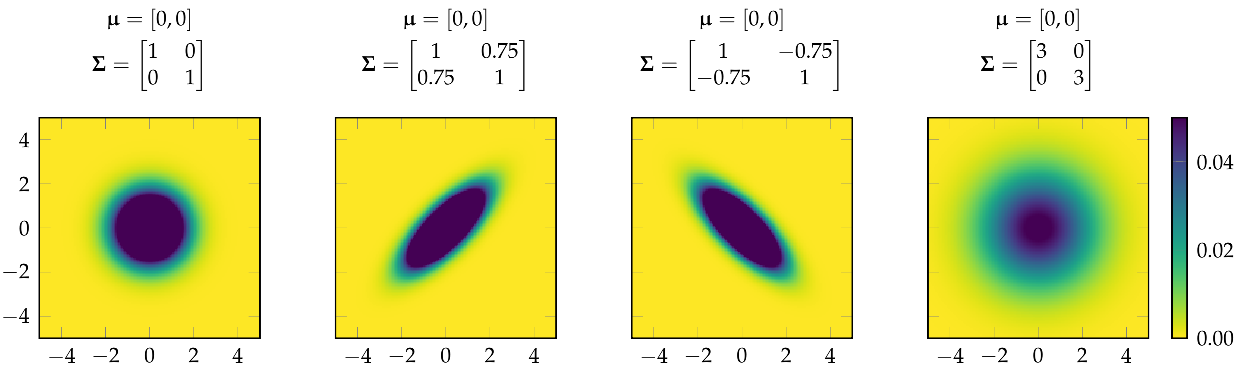
For two random variables X and Y that are jointly distributed, the *joint cumulative distribution function* $F_{X,Y}$ can be defined as

$$F_{X,Y}(a,b) = P(X \leq a, Y \leq b)$$

$$\begin{aligned} F_{X,Y}(a,b) &= \sum_{x \leq a} \sum_{y \leq b} p_{X,Y}(x,y) && X, Y \text{ discrete} \\ F_{X,Y}(a,b) &= \int_{-\infty}^{a} \int_{-\infty}^{b} f_{X,Y}(x,y)dydx && X, Y \text{ continuous} \\ f_{X,Y}(a,b) &= \frac{\partial^2}{\partial a\, \partial b} F_{X,Y}(a,b) && X, Y \text{ continuous} \end{aligned}$$

It can be shown via geometry that to calculate probabilities of joint distributions, we can use the CDF as follows, for both jointly discrete and continuous RVs:

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = F_{X,Y}(a_2,b_2) - F_{X,Y}(a_1,b_2) - F_{X,Y}(a_2,b_1) + F_{X,Y}(a_1,b_1)$$



$\boldsymbol{\mu} = [0,0]$, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

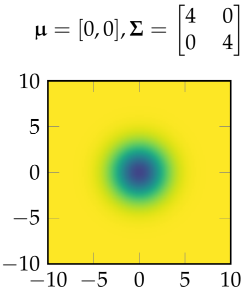
$\boldsymbol{\mu} = [0,0]$, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix}$

$\boldsymbol{\mu} = [0,0]$, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & -0.75 \\ -0.75 & 1 \end{bmatrix}$

$\boldsymbol{\mu} = [0,0]$, $\boldsymbol{\Sigma} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$

16.3 *Bivariate Normal Distribution*

Many times, we talk about multiple Normal (Gaussian) random variables, otherwise known as *Multivariate Normal* (Gaussian) distributions. Here, we talk about the two-dimensional case, called a *Bivariate Normal Distribution*. Variables X_1 and X_2 follow a bivariate normal distribution if their joint PDF is:



$\boldsymbol{\mu} = [0,0], \boldsymbol{\Sigma} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$

$$f_{X_1,X_2}(x_1,x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right)}$$

We often write the distribution of the vector $\mathbf{X} = (X_1, X_2)$ as $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = (\mu_1, \mu_2)$ is a *mean vector* and $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$ is a *covariance matrix*.

Note that ρ is the correlation between X_1 and X_2, and $\sigma_1, \sigma_2 > 0$. We defer to Ross Chapter 6, Example 5d, for the full proof, but it can be shown that the marginal distributions of X_1 and X_2 are $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$, respectively.

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) \end{bmatrix}$$

where

$\mathrm{Cov}(X, X) = \mathrm{Var}(X)$

Example 16.1. *Multivariate Gaussian distribution* of uncorrelated random variables.

Let $\mathbf{X} = (X_1, X_2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$, a diagonal covariance matrix. Note the correlation between X_1 and X_2 is $\rho = 0$:

$$\begin{aligned} f_{X_1,X_2}(x_1, x_2) &= \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right)} \\ &= \underbrace{\frac{1}{\sigma_1\sqrt{2\pi}} e^{-(x_1-\mu_1)^2/(2\sigma_1^2)}}_{\text{involves } x_1} \underbrace{\frac{1}{\sigma_2\sqrt{2\pi}} e^{-(x_2-\mu_2)^2/(2\sigma_2^2)}}_{\text{involves } x_2} \end{aligned}$$

In other words, for Bivariate Normal RVs, if $\text{Cov}(X_1, X_2) = 0$, then X_1 and X_2 are independent. Wild!

Example 16.2. Gaussian blur using a *bivariate normal distribution* (i.e. *multivariate*) with a weight matrix.

Let's make a weight matrix used for Gaussian blur. In the weight matrix, each location in the weight matrix will be given a weight based on the probability density of the area covered by that grid square in a Bivariate Normal of independent X and Y, each zero mean with variance σ^2. For this example lets blur using $\sigma = 3$.

Each pixel is given a weight equal to the probability that X and Y are both within the pixel bounds. The center pixel covers the area where $-0.5 \leq x \leq 0.5$ and $-0.5 \leq y \leq 0.5$. What is the weight of the center pixel?

$$\begin{aligned} &P(-0.5 < X < 0.5, -0.5 < Y < 0.5) = \\ &\quad P(X < 0.5, Y < 0.5) - P(X < 0.5, Y < -0.5) \\ &\quad - P(X < -0.5, Y < 0.5) + P(X < -0.5, Y < -0.5) \\ &\quad = \phi\left(\frac{0.5}{3}\right) \cdot \phi\left(\frac{0.5}{3}\right) - 2\phi\left(\frac{0.5}{3}\right) \cdot \phi\left(\frac{-0.5}{3}\right) + \phi\left(\frac{-0.5}{3}\right) \cdot \phi\left(\frac{-0.5}{3}\right) \\ &\quad = 0.5662^2 - 2 \cdot 0.5662 \cdot 0.4338 + 0.4338^2 \approx 0.206 \end{aligned}$$

17 Continuous Joint Distributions II

From notes by Chris Piech and Lisa Yan.

17.1 Convolution: Sum of Independent Random Variables

Remember how deriving the sum of two independent Poisson random variables was tricky? When we move into integral land, the concept of convolution still carries over, and once you get a handle on notation, then computing the sum of two independent, jointly continuous random variables becomes fun. For some definition of fun...

17.2 Independent Normals

Let's start with one common case that has a nice form but a difficult derivation: For any two normal random variables $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ the sum of those two random variables is another normal: $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.

We won't derive the approach here, but it involves exponents, integrals, and completing the square (algebra throwback!).

17.3 General Independent Case

For two general independent random variables (i.e. cases of independent random variables that don't fit the above special situations) you can calculate the CDF or the PDF of the sum of two random variables using the following convolution formulas:

$$F_{X+Y}(a) = P(X + Y \leq a) = \int_{y=-\infty}^{\infty} F_X(a-y) F_Y(y) dy$$

$$f_{X+Y}(a) = \int_{y=-\infty}^{\infty} f_X(a-y) f_Y(y) dy$$

These is a direct analogy to the discrete case where you replace the integrals with sums and change notation for CDF and PDF.

17.4 Conditional Distributions (Continuous Case)

The conditional probability density function might look a bit wonky, but it works!

Continuous: The conditional probability density function (PDF) for the continuous case:

$$f_{X|Y}(x \mid y) = \frac{f_{X,Y}(x,y)}{f_Y(y)} \tag{17.1}$$

The conditional cumulative density function (CDF) for the continuous case:

$$F_{X|Y}(a \mid y) = P(X \leq a \mid Y = y) = \int_{-\infty}^{a} f_{X|Y}(x \mid y) dx \tag{17.2}$$

At first glance, the conditional density function seems to violate any notion of stoichiometric units of probability. Let us verify this with our understanding of discrete probability. Recall that for tiny epsilon ϵ, we can approximate:

$$P(|X - x| \leq \frac{\epsilon}{2}) = P(x - \frac{\epsilon}{2} \leq X \leq x + \frac{\epsilon}{2}) = \int_{x-\frac{\epsilon}{2}}^{x+\frac{\epsilon}{2}} f_X(a) da \approx f_X(x)\epsilon$$

This extends to the joint variable case: $P(|X - x| \leq \frac{\epsilon_X}{2}, |Y - y| \leq \frac{\epsilon_y}{2}) \approx f_{X,Y}(x,y)\epsilon_X \epsilon_Y$.

$$P\left(|X - x| \leq \frac{\epsilon_X}{2} \mid |Y - y| \leq \frac{\epsilon_Y}{2}\right) = \frac{P(|X - x| \leq \frac{\epsilon_X}{2}, |Y - y| \leq \frac{\epsilon_Y}{2})}{P(|Y - y| \leq \frac{\epsilon_Y}{2})} \quad \text{(def. cond. prob.)}$$

$$\approx \frac{f_{X,Y}(x,y)\epsilon_X \epsilon_Y}{f_Y(y)\epsilon_Y} = \frac{f_{X,Y}(x,y)}{f_Y(y)}\epsilon_X = f_{X|Y}(x \mid y)\epsilon_X$$

17.5 Conditional Expectation (Continuous Case)

Conditional expectation in the continuous case is a direct analogy to what we saw in the discrete case:

Let X and Y be jointly continuous random variables. We define the conditional expectation of X given $Y = y$ to be:

$$\mathbb{E}[X \mid Y = y] = \int_{-\infty}^{\infty} x f_{X|Y}(x \mid y) dx \tag{17.3}$$

Example 17.1. Convolution of a *uniform distribution*.

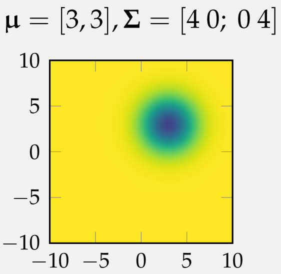
What is the PDF of $X + Y$ for independent uniform random variables $X \sim \text{Uni}(0, 1)$ and $Y \sim \text{Uni}(0, 1)$? First plug in the equation for general convolution of independent random variables:

$$f_{X+Y}(a) = \int_{y=0}^{1} f_X(a-y) f_Y(y) dy$$

$$f_{X+Y}(a) = \int_{y=0}^{1} f_X(a-y) dy \qquad \text{because } f_Y(y) = 1$$

It turns out that is not the easiest thing to integrate. By trying a few different values of a in the range $[0, 2]$ we can observe that the PDF we are trying to calculate is discontinuous at the point $a = 1$ and thus will be easier to think about as two cases: $a < 1$ and $a > 1$. If we calculate f_{X+Y} for both cases and correctly constrain the bounds of the integral we get simple closed forms for each case:

$$f_{X+Y}(a) = \begin{cases} a & \text{if } 0 < a \leq 1 \\ 2 - a & \text{if } 1 < a \leq 2 \\ 0 & \text{else} \end{cases}$$



Multivariate Example: In this example we are going to explore the problem of tracking an object in 2D space. The object exists at some (x, y) location, however we are not sure exactly where! Thus we are going to use random variables X and Y to represent location.

We have a prior belief about where the object is. In this example our prior both X and Y as normals which are independently distributed with mean 3 and variance 4. First let's write the prior belief as a joint probability density function:

$$\begin{aligned} f(X = x, Y = y) &= f(X = x) \cdot f(Y = y) && \text{(In the prior X and Y are independent)} \\ &= \frac{1}{\sqrt{2 \cdot 4 \cdot \pi}} \cdot e^{-\frac{(x-3)^2}{2 \cdot 4}} \cdot \frac{1}{\sqrt{2 \cdot 4 \cdot \pi}} \cdot e^{-\frac{(y-3)^2}{2 \cdot 4}} && \text{(Using the PDF equation for normals)} \\ &= K_1 \cdot e^{-\frac{(x-3)^2 + (y-3)^2}{8}} && \text{(All constants are put into K_1)} \end{aligned}$$

This combinations of normals is called a bivariate distribution. Here is a visualization of the PDF of our prior. The interesting part about tracking an object is the process of updating your belief about it's location based on an observation. Let's say that we get an instrument reading from a sonar that is sitting on the origin. The instrument reports that the object is 4 units away. Our instrument is not perfect: if the true distance was t units away, than the instrument will give a reading which is normally distributed with mean t and variance 1. Let's visualize the observation:

Based on this information about the noisiness of our prior, we can compute the conditional probability of seeing a particular distance reading D, given the true location of the object X, Y. If we knew the object was at location (x, y), we could calculate the true distance to the origin $\sqrt{x^2 + y^2}$ which would give us the mean for the instrument Gaussian:

$$\begin{aligned} f(D = d \mid X = x, Y = y) &= \frac{1}{\sqrt{2 \cdot 1 \cdot \pi}} \cdot e^{-\frac{\left(d - \sqrt{x^2 + y^2}\right)^2}{2 \cdot 1}} && \text{(Normal PDF where $\mu = \sqrt{x^2 + y^2}$)} \\ &= K_2 \cdot e^{-\frac{\left(d - \sqrt{x^2 + y^2}\right)^2}{2 \cdot 1}} && \text{(All constants are put into K_2)} \end{aligned}$$

How about we try this out on actual numbers. How much more likely is an instrument reading of 1 compared to 2, given that the location of the object is at $(1, 1)$?

$$
\begin{aligned}
\frac{f(D = 1 \mid X = 1, Y = 1)}{f(D = 2 \mid X = 1, Y = 1)} &= \frac{K_2 \cdot e^{-\frac{\left(1-\sqrt{1^2+1^2}\right)^2}{2\cdot 1}}}{K_2 \cdot e^{-\frac{\left(2-\sqrt{1^2+1^2}\right)^2}{2\cdot 1}}} && \text{(Substituting into the conditional PDF of } D\text{)} \\
&= \frac{e^0}{e^{-1/2}} \approx 1.65 && \text{(Notice how the } K_2 \text{ cancel out)}
\end{aligned}
$$

At this point we have a prior belief and we have an observation. We would like to compute an updated belief, given that observation. This is a classic Bayes' formula scenario. We are using joint continuous variables, but that doesn't change the math much, it just means we will be dealing with densities instead of probabilities:

$$
\begin{aligned}
f(X = x, Y = y \mid D = 4) &= \frac{f(D = 4 \mid X = x, Y = y) \cdot f(X = x, Y = y)}{f(D = 4)} && \text{(Bayes using densities)} \\
&= \frac{K_1 \cdot e^{-\frac{\left(4-\sqrt{x^2+y^2}\right)^2}{2}} \cdot K_2 \cdot e^{-\frac{\left[(x-3)^2+(y-3)^2\right]}{8}}}{f(D = 4)} && \text{(Substituting for prior and update)} \\
&= \frac{K_1 \cdot K2}{f(D = 4)} \cdot e^{-\left[\frac{\left(4-\sqrt{x^2+y^2}\right)^2}{2} + \frac{(x-3)^2+(y-3)^2}{8}\right]} \\
& && \text{(} f(D = 4) \text{ is a constant w.r.t. } (x, y)\text{)} \\
&= K_3 \cdot e^{-\left[\frac{\left(4-\sqrt{x^2+y^2}\right)^2}{2} + \frac{(x-3)^2+(y-3)^2}{8}\right]} && \text{(} K_3 \text{ is a new constant)}
\end{aligned}
$$

Wow! That looks like a pretty interesting function! You have successfully computed the updated belief. Let's see what it looks like. Here is a figure with our prior on the left and the posterior on the right: How beautiful is that! Its like a 2D normal distribution merged with a circle. But wait, what about that constant! We do not know the value of K_3 and that is not a problem for two reasons: the first reason is that if we ever want to calculate a relative probability of two locations, K_3 will cancel out. The second reason is that if we really wanted to know what K_3 was, we could solve for it.

This math is used every day in millions of applications. If there are multiple observations the equations can get truly complex (even worse than this one). To represent these complex functions often use an algorithm called particle filtering.

Example 17.2. Convolution of Poisson random variables.

Let's say we have two independent random Poisson variables for requests received at a web server in a day: X = # requests from humans/day, $X \sim \text{Poi}(\lambda_1)$ and Y = # requests from bots/day, $Y \sim \text{Poi}(\lambda_2)$. Since the convolution of Poisson random variables is also a Poisson we know that the total number of requests $(X+Y)$ is also a Poisson $(X+Y) \sim \text{Poi}(\lambda_1+\lambda_2)$. What is the probability of having k human requests on a particular day given that there were n total requests?

$$\begin{aligned}
P(X=k \mid X+Y=n) &= \frac{P(X=k, Y=n-k)}{P(X+Y=n)} = \frac{P(X=k)P(Y=n-k)}{P(X+Y=n} \\
&= \frac{e^{-\lambda_1}\lambda_1^k}{k!} \cdot \frac{e^{-\lambda_2}\lambda_2^{n-k}}{(n-k)!} \cdot \frac{n!}{e^{1(\lambda_1+\lambda_2)}(\lambda_1+\lambda_2)^n} \\
&= \binom{n}{k}\left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right)^k \left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right)^{n-k} \\
&\sim \text{Bin}\left(n, \frac{\lambda_2}{\lambda_1+\lambda_2}\right)
\end{aligned}$$

18 Central Limit Theorem

18.1 The Theory

From notes by Chris Piech and Lisa Yan.

The *central limit theorem* proves that the averages of samples from any distribution themselves must be normally distributed. Consider independent and indentically distributed (IID)[1] random variables $X_1, X_2, \ldots$ such that $\mathbb{E}[X_i] = \mu$ and $\text{Var}(X_i) = \sigma^2$. Let:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{18.1}$$

The central limit theorem states:

$$\bar{X} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right) \qquad \text{as } n \to \infty \tag{18.2}$$

It is sometimes expressed in terms of the standard normal, Z:

$$Z = \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} \qquad \text{as } n \to \infty \tag{18.3}$$

At this point you probably think that the central limit theorem is awesome. But it gets even better. With some algebraic manipulation we can show that if the sample mean of IID random variables is normal, it follows that the sum of equally weighted IID random variables must also be normal. Let's call the sum of IID random variables $\bar{Y}$:

$$\begin{aligned}
\bar{Y} = \sum_{i=1}^{n} X_i = n \cdot \bar{X} & \qquad \text{(If we define } \bar{Y} \text{ to be the sum of our variables)} \\
\sim \mathcal{N}\left(n\mu, n^2\frac{\sigma^2}{n}\right) & \qquad \text{(Since } \bar{X} \text{ is a normal and } n \text{ is a constant)} \\
\sim \mathcal{N}\left(n\mu, n\sigma^2\right) & \qquad \text{(By simplifying)}
\end{aligned}$$

In summary, the central limit theorem explains that both the sample mean of IID variables is normal (regardless of what distribution the IID variables came from) and that the sum of equally weighted IID random variables is normal (again, regardless of the underlying distribution).

[1] Random variables $X_1, \ldots, X_n$ are IID if $X_1, \ldots, X_n$ are independent and they all have the same PMF (if discrete) or PDF (if continuous), with $\mathbb{E}[X_i] = \mu$ and $\text{Var}(X_i) = \sigma^2$ for $i = 1, \ldots, n$.

You will roll a 6 sided dice 10 times. Let X be the total value of all 10 dice $= X_1 + X_2 + \cdots + X_{10}$. You win the game if $X \leq 25$ or $X \geq 45$. Use the central limit theorem to calculate the probability that you win.
Recall that $\mathbb{E}[X_i] = 3.5$ and $\text{Var}(X_i) = \frac{35}{12}$.

$$\begin{aligned} P(X \leq 25 \text{ or } X \geq 45) & \\ &= 1 - P(25.5 \leq X \leq 44.5) \\ &= 1 - P\left(\frac{25.5 - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \leq \frac{X - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \leq \frac{44.5 - 10(3.5)}{\sqrt{35/12}\sqrt{10}}\right) \\ &\approx 1 - (2\Phi(1.76) - 1) \approx 2(1 - 0.9608) = 0.0784 \end{aligned}$$

Example 18.1. Calculating probability of winning a dice game using the *central limit theorem*.

18.1.1 Gimbel Distribution

A more obscure theorem, the Fisher-Tippett-Gnedenko theorem, tells us about the *max* of IID random variables. It says that the max of IID exponential or normal random variables will be a *Gumbel* random variable:

$$Y \sim \text{Gumbel}(\mu, \beta) \qquad \text{(The max of IID variables)}$$

$$f(Y = l) = \frac{1}{\beta} e^{-(z + e^{-z})} \text{ where } z = \frac{k - \mu}{\beta} \qquad \text{(the Gumbel PDF)}$$

Example 18.2. Algorithm runtime using the *central limit theorem*.

You want to test the runtime of a new algorithm. You know the variance of the algorithm's runtime: $\sigma^2 = 4$ sec^2 but you want to estimate the mean: $\mu = t$ sec. You can run the algorithm repeatedly (IID trials). How many trials do you have to run so that your estimated runtime $= t \pm 0.5$ with 95% certainty? Let X_i be the run time of the i-th run (for $1 \le i \le n$).

$$0.95 = P(-0.5 \le \frac{\sum_{i=1}^{n} X_i}{n} - t \le 0.5)$$

By the central limit theorem, the standard normal Z must be equal to:

$$Z = \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} = \frac{(\sum_{i=1}^{n} X_i) - nt}{2\sqrt{n}}$$

Now we rewrite our probability inequality so that the central term is Z:

$$\begin{aligned}
0.95 &= P\left(-0.5 \le \frac{\sum_{i=1}^{n} X_i}{n} - t \le 0.5\right) \\
&= P\left(\frac{-0.5\sqrt{n}}{2} \le \frac{\sum_{i=1}^{n} X_i}{n} - t \le \frac{0.5\sqrt{n}}{2}\right) \\
&= P\left(\frac{-0.5\sqrt{n}}{2} \le \frac{\sqrt{n}}{2}\frac{\sum_{i=1}^{n} X_i}{n} - \frac{\sqrt{n}}{2}t \le \frac{0.5\sqrt{n}}{2}\right) \\
&= P\left(\frac{-0.5\sqrt{n}}{2} \le \frac{\sum_{i=1}^{n} X_i}{2\sqrt{n}} - \frac{\sqrt{n}}{\sqrt{n}}\frac{\sqrt{n}t}{2} \le \frac{0.5\sqrt{n}}{2}\right) \\
&= P\left(\frac{-0.5\sqrt{n}}{2} \le \frac{\sum_{i=1}^{n} X_i - nt}{2\sqrt{n}} \le \frac{0.5\sqrt{n}}{2}\right) \\
&= P\left(\frac{-0.5\sqrt{n}}{2} \le Z \le \frac{0.5\sqrt{n}}{2}\right)
\end{aligned}$$

And now we can find the value of n that makes this equation hold.

$$\begin{aligned}
0.95 &= \Phi\left(\frac{\sqrt{n}}{4}\right) - \Phi\left(-\frac{\sqrt{n}}{4}\right) = \Phi\left(\frac{\sqrt{n}}{4}\right) - \left(1 - \Phi\left(\frac{\sqrt{n}}{4}\right)\right) \\
&= 2\Phi\left(\frac{\sqrt{n}}{4}\right) - 1 \\
0.975 &= \Phi\left(\frac{\sqrt{n}}{4}\right) \quad \therefore \quad \Phi^{-1}(0.975) = 1.96 = \frac{\sqrt{n}}{4} \implies n = 61.4
\end{aligned}$$

Thus it takes 62 runs. If you are interested in how this extends to cases where the variance is unknown, look into variations of the students' t-test.

19 *Samples and the Bootstrap*

From notes by Chris Piech and Lisa Yan.

Let's say you are the king of Bhutan and you want to know the average happiness of the people in your country. You can't ask every single person, but you could ask a random subsample. In this next section we will consider principled claims that you can make based on a subsample. Assume we randomly sample 200 Bhutanese and ask them about their happiness. Our data looks like this: $72, 85, \ldots, 71$. You can also think of it as a collection of $n = 200$ IID (independent, identically distributed) random variables $X_1, X_2, \ldots, X_n$.

19.1 *Estimating Mean and Variance from samples*

We assume that the data we look at are IID from the same underlying distribution (F) with a true mean μ and a true variance σ^2. Since we can't talk to everyone in Bhutan, we have to rely on our sample to estimate the mean and variance. From our sample we can calculate a sample mean $\bar{X}$ and a sample variance S^2. These are the best guesses that we can make about the true mean and true variance.

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n} \qquad\qquad S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1}$$

The first question to ask is: are those unbiased estimates? Yes. Unbiased, means that if we were to repeat this sampling process many times, the expected value of our estimates should be equal to the true values we are trying to estimate. We will prove that that is the case for $\bar{X}$. The proof for S^2 is in the lecture slides.

$$\begin{aligned}
\mathbb{E}[\bar{X}] &= \mathbb{E}\left[\sum_{i=1}^{n} \frac{X_i}{n}\right] = \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[X_i] = \frac{1}{n}\sum_{i=1}^{n} \mu = \frac{1}{n} n\mu = \mu
\end{aligned}$$

The equation for sample mean seems like a reasonable way to calculate the expectation of the underlying distribution. The same could be said about sample variance except for the surprising $(n1)$ in the denominator of the equation. Why $(n1)$? That denominator is necessary to make sure that the $E[S^2] = \sigma^2$.

The intuition behind the proof is that sample variance calculates the distance of each sample to the sample mean, *not* the true mean. The sample mean itself varies, and we can show that its variance is also related to the true variance.

19.2 *Standard Error*

Okay, you convinced me that our estimates for mean and variance are not biased. But now I want to know how much my sample mean might vary relative to the true mean.

$$\begin{aligned}
\text{Var}(\bar{X}) &= \text{Var}\left(\sum_{i=1}^{n} \frac{X_i}{n}\right) = \left(\frac{1}{n}\right)^2 \text{Var}\left(\sum_{i=1}^{n} X_i\right) \\
&= \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \text{Var}(X_i) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \sigma^2 = \left(\frac{1}{n}\right)^2 n\sigma^2 = \frac{\sigma^2}{n} \\
&\approx \frac{S^2}{n} && \text{(Since } S \text{ is an unbiased estimate)} \\
\text{Std}(\bar{X}) &\approx \sqrt{\frac{S^2}{n}} && \text{(Since Std is the square root of Var)}
\end{aligned}$$

That $\text{Std}(\bar{X})$ term has a special name. It is called the *standard error* and its how you report uncertainty of estimates of means in scientific papers (and how you get error bars). Great! Now we can compute all these wonderful statistics for the Bhutanese people. But wait! You never told me how to calculate the $\text{Std}(S^2)$. True, that is outside the scope of CS109. You can find it on Wikipedia if you want.

Let's say we calculate the our sample of happiness has $n = 200$ people. The sample mean is $\bar{X} = 83$ (what is the unit here? happiness score?) and the sample variance is $S^2 = 450$. We can now calculate the standard error of our estimate of the mean to be 1.5. When we report our results we will say that the average happiness score in Bhutan is 83 ± 1.5 with variance 450.

Example 19.1. Sample mean and sample variance of the happiness in Bhutan problem.

19.3 *Bootstrap*

Bootstrap is a newly invented statistical technique for both understanding distributions of statistics and for calculating p-values.[1] It was invented here at Stanford in 1979 when mathematicians were just starting to understand how computers, and computer simulations, could be used to better understand probabilities.

The first key insight is that: if we had access to the underlying distribution F then answering almost any question we might have as to how accurate our statistics are becomes straightforward. For example, in the previous section we gave a formula for how you could calculate the sample variance from a sample of size n. We know that in expectation our sample variance is equal to the true variance. But what if we want to know the probability that the true variance is within a certain range of the number we calculated? That question might sound dry, but it is critical to evaluating scientific claims! If you knew the underlying distribution F you could simply repeat the experiment of drawing a sample of size n from F, calculate the sample variance from our new sample and test what portion fell within a certain range.

The next insight behind bootstrapping is that the best estimate that we can get for F is from our sample itself! The simplest way to estimate F (and the one we will use in this class) is to assume that $P(X = k)$ is simply the fraction of times that k showed up in the sample. Note that this defines the probability mass function of our estimate $\hat{F}$ of F.

To calculate $\text{Var}(S^2)$ we could calculate S_i^2 for each resample i and after 10,000 iterations, we could calculate the sample variance of all the S_i^2s. The bootstrap has strong theoretic grantees, and is accepted by the scientific community, when calculating any statistic. It breaks down when the underlying distribution has a "long tail" or if the samples are not IID.

[1] A p-value is the probability that a scientific claim is incorrect

```
def bootstrap(sample):
    n = number of elements in sample
    pmf = estimate the underlying pmf from the sample
    stats = []
    repeat 10,000 times:
        resample = draw n new samples from the pmf
        stat = calculate your stat on the resample
        stats.append(stat)
    now stats is used to estimate the distribution of the stat
```

20 *Maximum Likelihood Estimation*

From notes by Chris Piech and Lisa Yan.

We have learned many different distributions for random variables, and all of those distributions had *parameters*: the numbers that you provide as input when you define a random variable. So far when we were working with random variables, we either were explicitly told the values of the parameters, or we could divine the values by understanding the process that was generating the random variables.

What if we don't know the values of the parameters and we can't estimate them from our own expert knowledge? What if instead of knowing the random variables, we have a lot of examples of data generated with the same underlying distribution? In this chapter we are going to learn formal ways of estimating parameters from data.

These ideas are critical for artificial intelligence. Almost all modern machine learning algorithms work like this: (1) Specify a probabilistic model that has parameters. (2) Learn the value of those parameters from data.

20.1 *Parameters*

Before we dive into parameter estimation, first let's revisit the concept of parameters. Given a model, the parameters are the numbers that yield the actual distribution. In the case of a Bernoulli random variable, the single parameter was the value p. In the case of a Uniform random variable, the parameters are the a and b values that define the min and max value. Here is a list of random variables and the corresponding parameters. From now on, we are going to use the notation θ to be a vector of all the parameters:

In the real world often you don't know the ''true'' parameters, but you get to observe data. Next up, we will explore how we can use data to estimate the model parameters.

Distribution	**Parameters**
Bernoulli(p)	$\theta = p$
Poisson(λ)	$\theta = \lambda$
Uniform(a, b)	$\theta = (a, b)$
Normal(μ, σ^2)	$\theta = (\mu, \sigma^2)$
$Y = mX + b$	$\theta = (m, b)$

Table 20.1. Probability distribution parameters θ.

It turns out there isn't just one way to estimate the value of parameters. There are two main approaches: *Maximum Likelihood Estimation* (MLE) and *Maximum A Posteriori* (MAP). Both of these approaches assume that your data are IID samples: $X_1, X_2, \ldots, X_n$ where all X_i are independent and have the same distribution.

20.2 *Maximum Likelihood*

Our first algorithm for estimating parameters is called *maximum likelihood estimation* (MLE). The central idea behind MLE is to select the parameters θ that make the observed data the most likely.

The data that we are going to use to estimate the parameters are going to be n independent and identically distributed (IID) samples: $X_1, X_2, \ldots, X_n$.

20.2.1 *Likelihood*

We made the assumption that our data are identically distributed. This means that they must have either the same probability mass function (if the data are discrete) or the same probability density function (if the data are continuous). To simplify our conversation about parameter estimation, we are going to use the notation $f(X \mid \theta)$ to refer to this shared PMF or PDF. Our new notation is interesting in two ways. First, we have now included a conditional on θ which is our way of indicating that the likelihood of different values of X depends on the values of our parameters. Second, we are going to use the same symbol f for both discrete and continuous distributions.

What does likelihood mean and how is "likelihood" different than "probability"? In the case of discrete distributions, likelihood is a synonym for the joint probability of your data. In the case of continuous distribution, likelihood refers to the joint probability density of your data.

Since we assumed each data point is independent, the likelihood of all our data is the product of the likelihood of each data point. Mathematically, the likelihood of our data given parameters θ is:

$$L(\theta) = \prod_{i=1}^{n} f(X_i \mid \theta) \tag{20.1}$$

For different values of parameters, the likelihood of our data will be different. If we have correct parameters, our data will be much more probable than if we have incorrect parameters. For that reason we write likelihood as a function of our parameters (θ).

20.2.2 *Maximization*

In maximum likelihood estimation (MLE) our goal is to chose values of our parameters (θ) that maximizes the likelihood function from the previous section. We are going to use the notation $\hat{\theta}$ to represent the best choice of values for our parameters. Formally, MLE assumes that:

$$\hat{\theta} = \arg\max_{\theta} L(\theta) \tag{20.2}$$

''Arg max'' is short for argument of the maximum. The arg max of a function is the value of the domain at which the function is maximized. It applies for domains of any dimension.

A cool property of arg max is that since log is a monotonic function, the arg max of a function is the same as the arg max of the log of the function! That's nice because logs make the math simpler.

If we find the arg max of the log of likelihood, it will be equal to the arg max of the likelihood. Therefore, for MLE, we first write the *log likelihood* function (LL):

$$LL(\theta) = \log L(\theta) = \log \prod_{i=1}^{n} f(X_i \mid \theta) = \sum_{i=1}^{n} \log f(X_i \mid \theta) \tag{20.3}$$

To use a maximum likelihood estimator, first write the log likelihood of the data given your parameters. Then chose the value of parameters that maximize the log likelihood function. Argmax can be computed in many ways. All of the methods that we cover in this class require computing the first derivative of the function.

20.2.3 Bernoulli MLE Estimation

For our first example, we are going to use MLE to estimate the p parameter of a Bernoulli distribution. We are going to make our estimate based on n data points which we will refer to as IID random variables $X_1, X_2, \ldots, X_n$. Every one of these random variables is assumed to be a sample from the same Bernoulli, with the same p, namely $X_i \sim Ber(p)$. We want to find out what that p is.

Step one of MLE is to write the likelihood of a Bernoulli as a function that we can maximize. Since a Bernoulli is a discrete distribution, the likelihood is the probability mass function. You may not have realized before that the probability mass function of a Bernoulli X can be written as:

$$f(X) = p^X(1-p)^{1-X} \tag{20.4}$$

Interesting! Where did that come from? It's an equation that allows us to say that the probability that $X = 1$ is p and the probability that $X = 0$ is $1 - p$. Convince yourself that when $X_i = 0$ and $X_i = 1$ the PMF returns the right probabilities. We write the PMF this way because it is differentiable.

Let's do some maximum likelihood estimation:

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i} && \text{(first write the likelihood function)} \\
LL(\theta) &= \sum_{i=1}^{n} \log p^{X_i}(1-p)^{1-X_i} && \text{(then take the log)} \\
&= \sum_{i=1}^{n} X_i(\log p) + (1-X_i)\log(1-p) \\
&= Y \log p + (n-Y)\log(1-p) && \text{(where } Y = \textstyle\sum_{i=1}^{n} X_i\text{)}
\end{aligned}$$

We have a formula for the log likelihood. Now we simply need to chose the value of p that maximizes our log likelihood. As your calculus teacher probably taught you, one way to find the value which maximizes a function that is to find the first derivative of the function and set it equal to 0.

$$\frac{\partial LL(p)}{\partial p} = Y\frac{1}{p} + (n-Y)\frac{-1}{1-p} = 0$$
$$\hat{p} = \frac{Y}{n} = \frac{\sum_{i=1}^{n} X_i}{n}$$

All that work to find out that the maximum likelihood estimate is simply the sample mean...

20.2.4 Poisson MLE Estimation

Practice is key. Let us estimate the best parameter values for a Poisson distribution. Like before, suppose we have n samples from our Poisson, which we represent as random variables $X_1, X_2, \ldots, X_n$. We assume that for all i, X_i are IID and $X_i \sim \text{Poi}(\lambda)$. Our parameter is therefore $\theta = \lambda$. The PMF of a Poisson is:

$$f(X \mid \lambda) = e^{-\lambda}\frac{\lambda^X}{X!} \tag{20.5}$$

Let's write the log-likelihood function first:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{X_i}}{X_i!} && \text{(likelihood function)} \\ LL(\theta) &= \sum_{i=1}^{n} -\lambda \log e + X_i \log \lambda - \log(X_i!) && \text{(log-likelihood function)} \\ &= \sum_{i=1}^{n} -n\lambda + \log \lambda \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \log(X_i!) && \text{(use log with base } e\text{)} \end{aligned}$$

Then, we differentiate with respect to our parameter λ and set it equal to 0. Note that $\sum_{i=1}^{n} \log(X_i)$ is a constant with respect to λ:

$$\frac{\partial LL(\theta)}{\partial \lambda} = -n + \frac{1}{\lambda}\sum_{i=1}^{n} X_i = 0$$

Finally, we solve and find that $\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} X_i$. Yup, it's the sample mean again!

20.2.5 Normal MLE Estimation

Let's keep practicing. Next, we will estimate the best parameter values for a normal distribution. All we have access to are n samples from our normal, which we represent as IID random variables $X_1, X_2, \ldots, X_n$. We assume that for all i, $X_i \sim \mathcal{N}(\mu = \theta_0, \sigma^2 = \theta_1)$. This example seems trickier because a normal has two parameters that we have to estimate. In this case, θ is a vector with two values. The first is the mean (μ) parameter, and the second is the variance (σ^2) parameter.

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(X_i \mid \theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\theta_1}} e^{-\frac{(X_i-\theta_0)^2}{2\theta_1}} && \text{(likelihood of a continuous variable is the PDF)} \\
LL(\theta) &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi\theta_1}} e^{-\frac{(X_i-\theta_0)^2}{2\theta_1}} && \text{(we want to calculate log likelihood)} \\
&= \sum_{i=1}^{n} \left[-\log\left(\sqrt{2\pi\theta_1}\right) - \frac{1}{2\theta_1}(X_i - \theta_0)^2 \right]
\end{aligned}$$

Again, the last step of MLE is to choose values of θ that maximize the log likelihood function. In this case, we can calculate the partial derivative of the LL function with respect to both θ_0 and θ_1, set both equations to equal 0, and then solve for the values of θ. Doing so results in the equations for the values $\hat{\mu} = \hat{\theta}_0$ and $\hat{\sigma}^2 = \hat{\theta}_1$ that maximize likelihood. The result is: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2$. Note that $\hat{\sigma}^2$ is biased because it relies on the unbiased $\hat{\mu}$.

21 The Beta Distribution

From notes by Chris Piech and Lisa Yan.

In this chapter we are going to have a very meta discussion about how we represent probabilities. Until now probabilities have just been numbers in the range 0 to 1. However, if we have uncertainty about our probability, it would make sense to represent our probabilities as random variables (and thus articulate the relative likelihood of our belief).

21.1 Mixing Discrete and Continuous

In order to characterize probabilities as random variables (recall that according to Axiom 1, probabilities are real values between 0 and 1) in the context of discrete experiment outcomes (e.g., number of heads in a certain number of coin flips), we must introduce one more concept: Bayes' theorem that mixes discrete PMFs with continuous PDFs. These equations are straightforward once you have your head around the notation for probability density functions $f_X(x)$ and probability mass functions $p_X(x)$.

Let X be continuous random variable and let N be a discrete random variable. The conditional probabilities of X given N and N given X respectively are:

$$f_{X|N}(x \mid n) = \frac{p_{N|X}(n \mid x) f_X(x)}{p_N(n)} \qquad p_{N|X}(n \mid x) = \frac{f_{X|N}(x \mid n) p_N(n)}{f_X(x)}$$

21.2 Estimating Probabilities

Imagine we have a coin and we would like to know its probability of coming up heads (p). We flip the coin $(n+m)$ times and it comes up head n times. One way to calculate the probability is to assume that it is exactly $p = \frac{n}{n+m}$. That number, however, is a coarse estimate, especially if $n+m$ is small. Intuitively it doesn't capture our uncertainty about the value of p. Just like with other random variables, it often makes sense to hold a distributed belief about the value of p.

To formalize the idea that we want a distribution for p we are going to use a random variable X to represent the probability of the coin coming up heads. Before flipping the coin, we could say that our belief about the coin's success probability is uniform: $X \sim \text{Uni}(0,1)$.

If we let N be the number of heads that came up, given that the coin flips are independent, $(N \mid X) \sim \text{Bin}(n+m, x)$. We want to calculate the probability density function for $X \mid N$. We can start by applying Bayes Theorem:

$$\begin{aligned}
f(X=x \mid N=n) &= \frac{P(N=n \mid X=x) f(X=x)}{P(N=n)} && \text{(Bayes Theorem)} \\
&= \frac{\binom{n+m}{n} x^n (1-x)^m}{P(N=n)} && \text{(Binomial PMF, Uniform PDF)} \\
&= \frac{\binom{n+m}{n}}{P(N=n)} x^n (1-x)^m && \text{(Moving terms around)} \\
&= \frac{1}{c} \cdot x^n (1-x)^m && \text{(where } c = \textstyle\int_0^1 x^n (1-x)^m dx\text{)}
\end{aligned}$$

21.3 *Beta Distribution*

The equation that we arrived at when using a Bayesian approach to estimating our probability defines a probability density function and thus a random variable. The random variable is called a *Beta distribution*, and it is defined as follows:

Support for *Beta*: $(0,1)$

The probability density function (PDF) for a Beta $X \sim \text{Beta}(a,b)$ is:

$$f(X=x) = \begin{cases} \frac{1}{B(a,b)} x^{a-1} (1-x)^{b-1} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

where $B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1} dx$ is a normalization constant.

A Beta distribution has the following statistics:

$\text{mode}(X) = \frac{a-1}{a+b-2}$

$$\begin{aligned}
\mathbb{E}[X] &= \frac{a}{a+b} \\
\text{Var}(X) &= \frac{ab}{(a+b)^2(a+b+1)}
\end{aligned}$$

All modern programming languages have a package for calculating Beta CDFs. You will not be expected to compute the CDF by hand in CS109.

To model our estimate of the probability of a coin coming up heads as a beta set $a = n + 1$ and $b = m + 1$. Beta is used as a random variable to represent a belief distribution of probabilities in contexts beyond estimating coin flips. It has many desirable properties: it has a support range that is exactly $(0, 1)$, matching the values that probabilities can take on and it has the expressive capacity to capture many different forms of belief distributions. Let's imagine that we had observed $n = 4$ heads and $m = 2$ tails. The probability density function for $X \sim \text{Beta}(5, 3)$ is shown in figure 21.1.

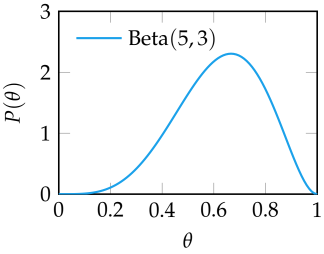


Figure 21.1. A Beta distribution for $\text{Beta}(5, 3)$.

Notice how the most likely belief for the probability of our coin is when the random variable, which represents the probability of getting a heads, is $4/6$, the fraction of heads observed. This distribution shows that we hold a non-zero belief that the probability could be something other than $4/6$. It is unlikely that the probability is 0.01 or 0.09, but reasonably likely that it could be 0.5.

It works out that $\text{Beta}(1, 1) = \text{Uni}(0, 1)$. As a result the distribution of our belief about p before (''prior'') and after (''posterior'') can both be represented using a Beta distribution. When that happens we call Beta a ''conjugate'' distribution. Practically, conjugate means easy update.

21.3.1 *Beta as a Prior*

You can set $X \sim \text{Beta}(a, b)$ as a prior to reflect how biased you think the coin is apriori to flipping it. This is a subjective judgment that represent $a + b - 2$ ''imaginary'' trials with $a - 1$ heads and $b - 1$ tails. If you then observe $n + m$ real trials with n heads you can update your belief. Your new belief would be, $X \mid (n \text{ heads in } n + m \text{ trials}) \sim \text{Beta}(a + n, b + m)$. Using the prior $\text{Beta}(1, 1) = \text{Uni}(0, 1)$ is the same as saying we haven't seen any ''imaginary'' trials, so apriori we know nothing about the coin. This form of thinking about probabilities is representative of the ''Bayesian'' field of thought where computer scientists explicitly represent probabilities as distributions (with prior beliefs). That school of thought is separate from the ''Frequentist'' school which tries to calculate probabilities as single numbers evaluated by the ratio of successes to experiments.

Example 21.1. Beta distribution to model student grades.

Assignment Example: In one particular iteration of this course, we talked about reasons why grade distributions might be well suited to be described as a Beta distribution. Let's say that we are given a set of student grades for a single exam and we find that it is best fit by a Beta distribution: $X \sim \text{Beta}(a = 8.28, b = 3.16)$. What is the probability that a student is below the mean (i.e. expectation)?

The answer to this question requires two steps. First calculate the mean of the distribution, then calculate the probability that the random variable takes on a value less than the expectation.

$$\mathbb{E}[X] = \frac{a}{a+b} = \frac{8.28}{8.28+3.16} \approx 0.7238$$

Now we need to calculate $P(X < \mathbb{E}[X])$. That is exactly the CDF of X evaluated at $\mathbb{E}[X]$. We don't have a formula for the CDF of a Beta distribution but all modern programming languages will have a Beta CDF function. In Python using the scipy stats library we can execute `stats.beta.cdf` which takes the x parameter first followed by the alpha and beta parameters of your Beta distribution.

$$\begin{aligned} P(X < \mathbb{E}[X]) &= F_X(0.7238) \\ &= \texttt{stats.beta.cdf(0.7238, 8.28, 3.16)} \approx 0.46 \end{aligned}$$

This can also be done in Julia using the `Distributions` package:

```
julia> using Distributions
julia> B = Beta(8.28, 3.16);
julia> cdf(B, 0.7238)
0.4602742456226714
```

22 Maximum A Posteriori

22.1 Maximum A Posteriori Estimation

From notes by Chris Piech and Lisa Yan.

MLE is great, but it is not the only way to estimate parameters! This section introduces an alternate algorithm, *Maximum A Posteriori* (MAP). The paradigm of MAP is that we should choose the value for our parameters that is the most likely given the data. At first blush this might seem the same as MLE; however, remember that MLE chooses the value of parameters that makes the *data* most likely.

One of the disadvantages of MLE is that it best explains data we have seen and makes no attempt to generalize to unseen data. In MAP, we incorporate *prior* belief about our parameters, and then we update our posterior belief of the parameters based on the data we have seen.

Formally, for IID random variables $X_1, \dots, X_n$:

$$\theta_{\text{MAP}} = \arg\max_{\theta} f(\theta \mid X_1, X_2, \dots, X_n)$$

In the equation above we trying to calculate the conditional probability of unobserved random variables given observed random variables. When that is the case, think Bayes' Theorem! Expand the function f using the continuous version of Bayes' Theorem:

$$\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_{\theta} f(\theta \mid X_1, X_2, \dots, X_n) \\
&= \arg\max_{\theta} \frac{f(X_1, X_2, \dots, X_n \mid \theta) g(\theta)}{h(X_1, X_2, \dots, X_n)} && \text{(by Bayes' Theorem)}
\end{aligned}$$

Note that f, g and h are all probability densities. We used different symbols to make it explicit that they may have different functions. Now we are going to leverage two observations. First, the data is assumed to be IID so we can decompose the density of the data given θ. Second, the denominator is a constant with respect to θ. As such, its value does not affect the arg max, and we can drop that term. Mathematically:

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} \frac{\prod_{i=1}^{n} f(X_i \mid \theta) g(\theta)}{h(X_1, X_2, \ldots, X_n)} && \text{(Since the samples are IID)} \\ &= \arg\max_{\theta} \prod_{i=1}^{n} f(X_i \mid \theta) g(\theta) && \text{(Since } h \text{ is constant with respect to } \theta\text{)}\end{aligned}$$

As before, it will be more convenient to find the arg max of the log of the MAP function, which gives us the final form for MAP estimation of parameters.

$$\theta_{MAP} = \arg\max_{\theta} \left(\log(g(\theta)) + \sum_{i=1}^{n} \log(f(X_i \mid \theta)) \right) \tag{22.1}$$

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f(X_i \mid \theta)$$

$$\theta_{MAP} = \arg\max_{\theta} \prod_{i=1}^{n} f(X_i \mid \theta) g(\theta)$$

Using Bayesian terminology, the MAP estimate is the mode of the ''posterior'' distribution for θ. If you look at this equation side by side with the MLE equation you will notice that MAP is the arg max of the exact same function plus a term for the log of the prior.

Maximum A Posteriori maximizes:

log-likelihood + log-prior

22.1.1 Parameter Priors

In order to get ready for the world of MAP estimation, we are going to need to brush up on our distributions. We will need reasonable distributions for each of our different parameters. For example, if you are predicting a Poisson distribution, what is the right random variable type for the prior of λ?

A desiderata for prior distributions is that the resulting posterior distribution has the same functional form. We call these ''conjugate'' priors. In the case where you are updating your belief many times, conjugate priors makes programming in the math equations much easier.

Here is a list of different parameters and the distributions most often used for their priors: We won't cover the inverse gamma distribution in this class. The remaining two, Dirichlet and Gamma, you will not be required to know, but details for them are included below for completeness.

Parameters	Distribution
Bernoulli p	Beta
Binomial p	Beta
Poisson λ	Gamma
Exponential λ	Gamma
Multinomial p_i	Dirichlet
Normal μ	Normal
Normal σ^2	Inverse Gamma

Table 22.1. List of distributions often used as priors.

The distributions used to represent your "prior" belief about a random variable will often have their own parameters. For example, a Beta distribution is defined using two parameters (a, b). Do we have to use parameter estimation to evaluate a and b too? No. Those parameters are called "hyperparameters". That is a term we reserve for parameters in our model that we fix before running parameter estimate. Before you run MAP you decide on the values of (a, b).

22.1.2 *Beta*

We've covered that Beta is a conjugate distribution for Bernoulli. The MAP of a Bernoulli distribution with a Beta prior is the *mode* of the Beta posterior. The *mode* of a distribution is the value that maximizes the probability mass function (if discrete) or probability density function (if continuous).

If $X \sim \text{Beta}(a, b)$, where a, b are integers where $a + b > 2$, the mode is $\arg\max_x f(x) = \frac{a-1}{a+b-2}$, where $f(x)$ is the PDF of X.

Flip $n + m$ coins and observe n heads. If we assume a prior on p of $\text{Beta}(n_{\text{imag}} + 1, m_{\text{imag}} + 1)$, the posterior on the parameter p is $\text{Beta}(n + n_{\text{imag}} + 1, m + m_{\text{imag}} + 1)$. The MAP estimator is therefore the mode of this distribution:

$$\frac{n + n_{\text{imag}}}{n + n_{\text{imag}} + m + m_{\text{imag}}}$$

22.1.3 *Dirichlet*

The Dirichlet distribution generalizes beta in the same way multinomial generalizes Bernoulli. A random variable X that is Dirichlet is parametrized as $X \sim \text{Dir}(a_1, a_2, \ldots, a_m)$. The PDF of the distribution is:

$$f(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m) = K \prod_{i=1}^{m} x_i^{a_i - 1}$$

Where K is a normalizing constant.

You can intuitively understand the hyperparameters of a Dirichlet distribution: imagine you have seen $\sum_{i=1}^{m} a_i - m$ imaginary trials. In those trials you had $(a_i - 1)$ outcomes of value i. As an example, consider estimating the probability of getting different numbers on a six-sided ''skewed die'' (where each side is a different shape). We will estimate the probabilities of rolling each side of this die by repeatedly rolling the die n times. This will produce n IID samples. For the MAP paradigm, we are going to need a prior on our belief of each of the parameters $p_1, \ldots, p_6$. We want to express that we lightly believe that each roll is equally likely.

Before you roll, let's imagine you had rolled the die six times and had gotten one of each possible value. Thus, the ''prior'' distribution would be $\text{Dir}(2, 2, 2, 2, 2, 2)$. After observing $n_1 + n_2 + \cdots + n_6$ new trials with n_i results of outcome i, the ''posterior'' distribution is $\text{Dir}(2 + n_1, \cdots, 2 + n_6)$.

Using a prior which represents one imagined observation of each outcome is called ''Laplace smoothing'' and it guarantees that none of your probabilities are 0 or 1. The Laplace estimate for a Multinomial RV is $p_i = \frac{X_i + 1}{n + m}$ for $i = 1, \ldots, m$, where X_i is the number of times you saw the outcome, and n is the number of actual trials in your observed experiment.

22.1.4 *Gamma*

The $\text{Gamma}(k, \theta)$ distribution is the conjugate prior for the λ parameter of the Poisson distribution. (It is also the conjugate for the λ in the exponential, but we won't cover that here.)

The hyperparameters can be interpreted as: you saw k total imaginary events during θ imaginary time periods. After observing n events during the next t time periods the posterior distribution is $\text{Gamma}(k + n, \theta + t)$.

As an example, $\text{Gamma}(10, 5)$ would represent having seen 10 imaginary events in 5 time periods. It is like imagining a rate of 2 with some degree of confidence. If we start with that Gamma as a prior and then see 11 events in the next 2 time periods our posterior is $\text{Gamma}(21, 7)$, which is equivalent to an updated rate of 3.

23 *Naïve Bayes*

From notes by Chris Piech and Lisa Yan.

Naïve Bayes is a type of machine learning algorithm called a classifier. It is used to predict the probability of a discrete *label* random variable Y based on the state of *feature* random variables $\mathbf{X}$. We are going to learn all necessary parameters for the probabilistic relationship between $\mathbf{X}$ and Y from data. Naïve Bayes is a *supervised classification* Machine Learning algorithm.

23.1 *Machine Learning: Classification*

In *supervised* machine learning, your job is to use training data with feature/label pairs $(\mathbf{I}, Y)$ in order to estimate a label-predicting function $\hat{Y} = g(X)$. This function can then be used to make future predictions. A *classification* task is one where Y takes on one of a *discrete* number of values. Often in classification, $g(X) = \arg\max_y \hat{P}(Y = y \mid X = x)$.

To learn all parameters required to calculate $g(\mathbf{X})$, you are given n different training pairs known as training data: $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$. $\mathbf{X}^{(j)}$ is a vector of m discrete features for the i-th training example, where $\mathbf{X}^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)})$. $y^{(j)}$ is the discrete label for the i-th training example. This symbolic description of classification hides the fact that prediction is applied to interesting real life problems:

1. Predicting heart disease Y, based on a set of m observations from a heart scan $\mathbf{X}$.

2. Predicting ancestry Y, based on m DNA states $\mathbf{X}$.

3. Predicting if a user will like a movie Y given whether or not they like a set of m movies $\mathbf{X}$.

In this section we are going to assume that all random variables are binary. While this is not a necessary assumption (Naïve Bayes can work for non-binary data), it makes it much easier to learn the core concepts. Specifically, we assume that all labels are binary $y \in \{0, 1\}$, and all features are binary $x_j \in \{0, 1\}, \forall j = 1, \ldots, m$.

23.2 *Naïve Bayes algorithm*

Here is the Naïve Bayes algorithm. After presenting the algorithm we are going to show the theory behind it.

23.2.1 *Prediction*

For an example with $\mathbf{X} = [x_1, x_2, \ldots, x_m]$, we can make a corresponding prediction for Y. We use hats (e.g., $\hat{P}$ or $\hat{Y}$) to symbolize values which are estimated.

$$\begin{aligned}
\hat{Y} = g(\mathbf{x}) &= \underset{y \in \{0,1\}}{\arg\max}\, \hat{P}(Y)\hat{P}(X \mid Y) \qquad \text{(This is equal to } \arg\max \hat{P}(Y = y \mid \mathbf{X})\text{)} \\
&= \underset{y \in \{0,1\}}{\arg\max}\, \hat{P}(Y = y) \prod_{i=1}^{m} \hat{P}(X_i = x_i \mid Y = y) \\
&\hfill \text{(Naïve Bayes assumption)} \\
&= \underset{y \in \{0,1\}}{\arg\max}\, \log \hat{P}(Y = y) + \sum_{i=1}^{m} \log \hat{P}(X_i = x_i \mid Y = y) \\
&\hfill \text{(Log version for numerical stability)}
\end{aligned}$$

In order to calculate this expression, we are going to need to learn the estimates $\hat{P}(Y = y)$ and $\hat{P}(X_i = x_i \mid Y = y)$ from data using a process called *training*.

23.2.2 *Training*

The objective in training is to estimate the probabilities $P(Y)$ and $P(X_i \mid Y)$ for all $0 < i \leq m$ features. Using an MLE estimate:

$$\hat{P}(X_i = x_i \mid Y = y) = \frac{(\text{\# training examples where } X_i = x_i \text{ and } Y = y)}{(\text{training examples where } Y = y)}$$

Using a Laplace MAP estimate:

$$\hat{P}(X_i = x_i \mid Y = y) = \frac{(\text{\# training examples where } X_i = x_i \text{ and } Y = y) + 1}{(\text{training examples where } Y = y) + 2}$$

Estimating $P(Y = y)$ is also straightforward. Using MLE estimation:

$$\hat{P}(Y = y) = \frac{(\text{\# training examples where } Y = y)}{(\text{training examples})}$$

23.3 Theory

Now that you have the algorithm spelled out, let's go over the theory of how we got there. To do so, we will first explore an algorithm which doesn't work, called ''Brute Force Bayes.'' Then, we introduce the Naïve Bayes Assumption, which will make our calculations possible.

23.3.1 Brute Force Bayes

We can solve the classification task using a brute force solution. To do so we will learn the full joint distribution $\hat{P}(Y, \mathbf{X})$.

In the world of classification, when we make a prediction, we want to chose the value of y that maximizes $P(Y = y \mid X = x)$. If we can only estimate $\hat{P}(Y = y \mid X = x)$, then we want to find a function $g(\mathbf{X}) = \arg\max_y \hat{P}(Y \mid \mathbf{X})$.

$$\begin{aligned}
\hat{y} = g(x) &= \arg\max_{y \in \{0,1\}} \hat{P}(Y \mid \mathbf{X}) && \text{(Our objective)} \\
&= \arg\max_{y \in \{0,1\}} \frac{\hat{P}(\mathbf{X} \mid Y)\hat{P}(Y)}{\hat{P}(\mathbf{X})} && \text{(By Bayes' Theorem)} \\
&= \arg\max_{y \in \{0,1\}} \hat{P}(\mathbf{X} \mid Y)\hat{P}(Y) && \text{(Since } \hat{P}(\mathbf{X}) \text{ is constant with respect to } Y\text{)}
\end{aligned}$$

Using our training data, we could interpret the joint distribution of X and Y as one giant Multinomial with a different parameter for every combination of $\mathbf{X} = \mathbf{x}$ and $Y = y$. If for example, the input vectors are only length one (i.e., $|\mathbf{X}| = 1$) and the number of values that x and y can take on are small—say, binary—this is a totally reasonable approach. We could estimate the multinomial using MLE or MAP estimators and then calculate argmax over a few lookups in our table.

The bad times hit when the number of features becomes large. Recall that our multinomial needs to estimate a parameter for every unique combination of assignments to the vector $\mathbf{X}$ and the value Y. If there are $|\mathbf{X}| = m$ binary features then this strategy is going to take order $O(2^m)$ space and there will likely be many parameters that are estimated without any training data that matches the corresponding assignment.

23.3.2 *Naïve Bayes Assumption*

The Naïve Bayes Assumption is that each feature of $\mathbf{X}$ is conditionally independent of one another given Y. That assumption is naïve (and often wrong), but useful. This assumption allows us to make predictions using space and data which is linear with respect to the size of the features: $O(m) if |\mathbf{x}| = m$. That allows us to train and make predictions for huge feature spaces, such as one which has an indicator for every word on the internet. Using this assumption the prediction algorithm can be simplified:

$$\begin{aligned}
\hat{y} = g(x) &= \underset{y \in \{0,1\}}{\arg\max}\, \hat{P}(\mathbf{X}, Y) && \text{(As we last left off)} \\
&= \underset{y \in \{0,1\}}{\arg\max}\, \hat{P}(Y)\hat{P}(\mathbf{X} \mid Y) && \text{(By chain rule)} \\
&= \underset{y \in \{0,1\}}{\arg\max}\, \hat{P}(Y) \prod_{i=1}^{m} \hat{P}(X_i \mid Y) && \text{(Using the Naïve Bayes assumption)} \\
&= \underset{y \in \{0,1\}}{\arg\max}\, \log \hat{P}(Y) + \sum_{i=1}^{m} \log \hat{P}(X_i \mid Y) \\
& && \text{(Log version for numerical stability)}
\end{aligned}$$

This algorithm is fast and stable both when training and making predictions.

Let us consider a particular feature, the i-th feature X_i . How should we represent $\hat{P}(X_i = x_i \mid Y = y)$? For a particular event $Y = y$ that we condition on, X_i can take on one of k discrete values . Thus for each particular y, we can model the likelihood of X_i taking on values as a Multinomial random variable with k parameters. We can then find MLE and MAP estimators for the parameters of that Multinomial. Recall that the MLE to estimate parameter p_i for a Multinomial is just counting, whereas the MAP estimator (with Laplace prior) to estimate parameter p_i imagines one extra example of each outcome:

$$\hat{p}_{i,\text{MLE}} = \frac{n_i}{n} \quad \text{and} \quad \hat{p}_{i,\text{MAP}} = \frac{n_i + 1}{n + k},$$

where n is the number of observations, n_i is the number of observations with outcome i, and k is the total possible number of outcomes k.

Note that in the version of classification we are using in CS109, X_i is binary (technically, a Multinomial with 2 parameters) and therefore $k = 2$. We used the Multinomial derivation to help you understand how one would handle a feature X_i that takes on multiple discrete values.

Naïve Bayes is a simple form of a Bayesian Network where the label Y is the only variable which directly influences the likelihood of each feature variable X_i. It is a simple model from a field of machine learning called *probabilistic graphical models*. In that field you make a graph of how your variables are related to one another and you come up with conditional independence assumptions that make it computationally tractable to estimate the joint distribution.

Say we have thirty examples of people's preferences (like or not) for Star Wars, Harry Potter and Pokemon. Each training example has X_1 , X_2 and Y where X_1 is whether or not the user liked Star Wars, X_2 is whether or not the user liked Harry Potter and Y is whether or not the user liked Pokemon. For the 30 training examples, the MAP and MLE estimates are as follows:

For a new user who likes Star Wars ($X_1 = 1$) but not Harry Potter ($X_2 = 0$), do you predict that they will like Pokemon? Yes! $Y = 1$ leads to a larger value in the argmax term:

$$\begin{aligned} &\text{if } Y = 0: \\ &\quad \hat{P}(X_1 = 1 \mid Y = 0)\hat{P}(X_2 = 0 \mid Y = 0)\hat{P}(Y = 0) = (0.77)(0.38)(0.43) \approx 0.126 \\ &\text{if } Y = 1: \\ &\quad \hat{P}(X_1 = 1 \mid Y = 1)\hat{P}(X_2 = 0 \mid Y = 1)\hat{P}(Y = 1) = (0.76)(0.41)(0.57) \approx 0.178 \end{aligned}$$

24 *Linear Regression and Gradient Ascent*

From notes by Chris Piech and Lisa Yan.

24.1 *Regression*

Regression is a second category of machine learning prediction algorithms. You have a prediction function $\hat{Y} = g(\mathbf{X})$ as before, but you would like to predict a Y that takes on a *continuous* number.

We won't elaborate on the regression task too much, because classification (with discrete Y) already has a plethora of modern computer science applications—image recognition, sentiment analysis of text, and text authorship, to name a few. However, we will explore *linear regression* (where we model g as a linear function) and learn a truly valuable iterative optimization algorithm (the "butter" to machine learning's "bread," if you will) called *gradient ascent*.

24.2 *Gradient Ascent Optimization*

In many cases we can't solve for argmax mathematically. Instead we use a computer. To do so we employ an algorithm called gradient ascent (a classic in optimization theory). The idea behind gradient ascent is that if you continuously take small steps in the direction of your gradient, you will eventually make it to a local maxima.

Start with theta as any initial value (often 0). Then take many small steps towards a local maxima. The new theta after each small step can be calculated as:

$$\theta_j^{\text{new}} = \theta_j^{\text{old}} + \eta \cdot \frac{\partial LL(\theta^{\text{old}})}{\partial \theta_j}$$

Where "eta" (η) is the magnitude of the step size that we take. If you keep updating θ using the equation above you will (often) converge on good values of θ. As a general rule of thumb, use a small value of η to start. If ever you find that the function value (for the function you are trying to argmax) is decreasing, your choice of η was too large. Here is the gradient ascent algorithm in pseudocode:

```
function GradientAscent
    Initialize $\theta_j = 0$ for all $0 \leq j \leq m$
    for many iterations do:
        gradient[j] $= 0$ for all $0 \leq j \leq m$
        Calculate all gradient[j]'s based on data and current data setting
        $\theta_j \leftarrow \theta_j + \eta *$ gradient[j] for all $0 \leq j \leq m$
```

Algorithm 24.1. Gradient ascent.

24.3 Linear Regression

Suppose we are working with 1-dimensional observations, i.e., $\mathbf{X} =< X_1 >= X$. Linear Regression assumes the following linear model for prediction, which has two parameters, a and b:

$$\hat{Y} = g(\mathbf{X}) = aX + b$$

Using this model, we would like to determine the optimal parameters according to some optimization objective. We discuss two approaches: an analytical approach that minimizes mean squared error, and a computational approach that maximizes training data likelihood. With one important assumption (which we'll get to later), the two approaches are equivalent.

24.3.1 Analytical Solution with Mean Squared Error

For regression tasks, we usually decide a prediction $\hat{Y} = g(X)$ that minimizes the mean squared error (MSE) "loss" function:

$$\begin{aligned} \theta_{MSE} &= \arg\min_{\theta} \mathbb{E}[(Y - \hat{Y})^2] \\ &= \arg\min_{\theta} \mathbb{E}[(Y - g(\mathbf{X}))^2] \\ &= \arg\min_{\theta} \mathbb{E}[(Y - aX - b)^2] \end{aligned}$$

With our linear prediction model, we determine $\theta_{MSE} = (a_{MSE}, b_{MSE})$ by differentiating the mean squared error with respect to a and b:

$$\begin{aligned}
\frac{\partial}{\partial a}\mathbb{E}[(Y - aX - b)^2] &= \mathbb{E}\left[\frac{\partial}{\partial a}(Y - aX - b)^2\right] \\
&= \mathbb{E}[-2(Y - aX - b)X] \\
&= -2\mathbb{E}[XY] + 2a\mathbb{E}[X^2] + 2b\mathbb{E}[X] \\
\frac{\partial}{\partial b}\mathbb{E}[(Y - aX - b)^2] &= \mathbb{E}\left[\frac{\partial}{\partial b}(Y - aX - b)^2\right] \\
&= \mathbb{E}[-2(Y - aX - b)] \\
&= -2\mathbb{E}[Y] + 2a\mathbb{E}[X] + 2b
\end{aligned}$$

Setting derivatives to 0 and solving for simultaneous equations:

$$\begin{aligned}
a_{MSE} &= \frac{\mathbb{E}[XY] - \mathbb{E}[X]E[Y]}{E[X^2] - (E[X])^2} = \frac{\mathrm{Cov}(X, Y)}{\mathrm{Var}(X)} = \rho(X, Y)\frac{\sigma_X}{\sigma_Y} \\
b_{MSE} &= \mathbb{E}[Y] - a\mathbb{E}[X] = \mu_Y - a_{MSE}\mu_X \\
Y &= \rho(X, Y)\frac{\sigma_Y}{\sigma_X}(X - \mu_X) + \mu_Y
\end{aligned}$$

Wait, those are our best parameters? But we don't know the distributions of X and Y, and therefore we don't know true statistics on X and Y. We estimate these statistics based on our observed training data. Our model is therefore as follows (where $\bar{X}$ and $\bar{Y}$ are the sample means computed from the training data:

$$\begin{aligned}
\hat{Y} = g(X = x) = \hat{\rho}(X, Y)\frac{\hat{\sigma}_Y}{\hat{\sigma}_X}(x - \bar{X}) + \bar{Y} \\
\hat{a}_{MSE} &= \frac{\sum_{i=1}^{n}(x^{(i)} - \bar{X})(y^{(i)} - \bar{Y})}{\sum_{i=1}^{n}(x^{(i)} - \bar{X})^2} = \hat{\rho}(X, Y)\frac{S_Y}{S_X} \\
\hat{b}_{MSE} &= \bar{Y} - \hat{a}_{MSE}\bar{X}
\end{aligned}$$

24.3.2 *Computational Solution with Maximum Likelihood*

That seemed somewhat anticlimactic: we had this optimal prediction function, but we had to estimate the parameters of the prediction function from the training data. Let's borrow an idea from our parameter estimation unit by maximizing the likelihood of our training data!

Recall that our training data has n datapoints

$$((x^{(1)}, y^{(1)}), ((x^{(2)}, y^{(2)}), \dots, ((x^{(n)}, y^{(n)}),$$

generated IID according to the joint distribution of X and Y, $f(X, Y \mid \theta)$. We can model this joint distribution by incorporating our regression model: $Y = \hat{Y} + Z = aX + b + Z$, where $\hat{Y} = g(X) = aX + b$ is our prediction and Z is our error (i.e., noise) between our prediction $\hat{Y}$ and the actual Y.

We approach the problem of finding a and b that maximize the likelihood of our train data by first finding a distribution involving Y, X, and $\theta = (a, b)$. We then find the value of θ that maximizes the log-likelihood function.

If we assume $Z \sim \mathcal{N}(0, \sigma^2)$ and X follows some unknown distribution, then we can calculate the conditional distribution of Y given X is some number x and we have some parameter values $\theta = (a, b)$ as simply $Y = ax + b + Z$. This is just the sum of a Gaussian and a number, thereby implying that $Y \mid X, \theta \sim \mathcal{N}(aX + b, \sigma^2)$, which has PDF:

$$f(Y = y \mid X = x, \theta = (a, b)) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-ax-b)^2}{2\sigma^2}}$$

Now we are ready to write the likelihood function, then take its log to get the log likelihood function:

Optimize the log conditional likelihood:

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log f(y^{(i)} \mid x^{(i)}, \theta)$$

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(y^{(i)}, x^{(i)} \mid \theta) && \text{(Let's break up this joint)} \\
&= \prod_{i=1}^{n} f(y^{(i)} \mid x^{(i)}, \theta) f(x^{(i)}) && \text{(Chain rule, } f_X(x^{(i)}) \text{ is independent of } \theta\text{)} \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-(y^{(i)} - ax(i) + b)^2/(2\sigma^2)} \cdot f(x^{(i)}) \\
& && \text{(Substitute in the conditional distribution of } Y \mid X, \theta\text{)} \\
LL(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-(y^{(i)} - ax(i) - b)^2/(2\sigma^2)} f(x^{(i)}) && \text{(Substitute in } L(\theta)\text{)} \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-(y^{(i)} - ax(i) - b)^2/(2\sigma^2)} + \sum_{i=1}^{n} \log f(x^{(i)}) \\
& && \text{(Log of a product is the sum of logs)} \\
&= n \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y^{(i)} - ax(i) - b)^2 + \sum_{i=1}^{n} \log f(x^{(i)})
\end{aligned}$$

Our goal is to find parameters a, b that maximize likelihood. Remember that argmax is invariant of logarithmic transformations and positive scalar constants, and additive constants. Let's remove positive constant multipliers and terms that don't include θ. We are left with trying to find a value of θ that maximizes:

$$\hat{\theta} = \arg\max_{\theta} \left[-\sum_{i=1}^{n} (y^{(i)} - ax(i) - b)^2 \right]$$

To solve this argmax we are going to use Gradient Ascent. In order to do so we first need to find the derivative of the function we want to argmax with respect to both parameters in θ:

$$\begin{aligned}
\frac{\partial}{\partial a} \left[-\sum_{i=1}^{n} (y^{(i)} - ax(i) - b)^2 \right] &= -\sum_{i=1}^{n} \frac{\partial}{\partial a} (y^{(i)} - ax(i) - b)^2 \\
&= -\sum_{i=1}^{n} 2(y^{(i)} - ax(i) - b)(-x^{(i)}) \\
&= 2\sum_{i=1}^{n} (y^{(i)} - ax(i) - b)(x^{(i)}) \\
\frac{\partial}{\partial b} \left[-\sum_{i=1}^{n} (y^{(i)} - ax(i) - b)^2 \right] &= 2\sum_{i=1}^{n} (y^{(i)} - ax(i) - b)
\end{aligned}$$

This first derivative can be plugged into gradient ascent to give our final algorithm:

```
a, b = 0, 0  # initialize θ
repeat many times:
    gradient_a, gradient_b = 0, 0
    for each training example (x, y):
        diff = y - (a * x + b)
        gradient_a += 2 * diff * x
        gradient_b += 2 * diff
    a += η * gradient_a  # θ += η * gradient
    b += η * gradient_b
```

If you run gradient ascent for enough training (i.e., update) steps, you will find that for linear regression, the maximum likelihood estimators (assuming zero-mean, normally distributed noise between predicted $\hat{Y}$ and actual Y) is equivalent to the mean squared error estimators. Cool!!

25 Logistic Regression

From notes by Chris Piech and Lisa Yan.

Before we get started, I want to familiarize you with some notation:

$$\theta^\top \mathbf{X} = \theta \cdot \mathbf{X} = \sum_{i=1}^{n} \theta_i X_i = \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n \qquad \text{(weighted sum)}$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \text{(sigmoid function)}$$

25.1 Logistic Regression Overview

Classification is the task of choosing a value of y that maximizes $P(Y \mid \mathbf{X})$. Naïve Bayes worked by approximating that probability using the naïve assumption that each feature was independent given the class label.

For all classification algorithms you are given n IID training datapoints

$$(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})$$

where each "feature" vector $\mathbf{x}^{(i)}$ has $m = |\mathbf{x}^{(i)}|$ features.

Logistic regression classifier:

$$\hat{Y} = \underset{y \in \{0,1\}}{\arg\max}\, P(Y \mid \mathbf{X})$$

$$P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \sigma\left(\sum_{j=0}^{m} \theta_j x_j\right) = \sigma(\theta^\top \mathbf{x})$$

25.1.1 Logistic Regression Assumption

Logistic Regression is a classification algorithm (I know, terrible name) that works by trying to learn a function that approximates $P(Y \mid \mathbf{X})$. It makes the central assumption that $P(Y \mid \mathbf{X})$ can be approximated as a sigmoid function applied to a linear combination of input features. Mathematically, for a single training datapoint $(\mathbf{x}, y)$ Logistic Regression assumes:

$$P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \sigma(z) \text{ where } z = \theta_0 + \sum_{i=1}^{m} \theta_i x_i$$

This assumption is often written in the equivalent forms:

$$P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \sigma(\theta^\top \mathbf{x}) \qquad \text{(where we always set } x_0 \text{ to be 1)}$$
$$P(Y = 0 \mid \mathbf{X} = \mathbf{x}) = 1 - \sigma(\theta^\top \mathbf{x}) \qquad \text{(by total law of probability)}$$

Using these equations for probability of $Y \mid \mathbf{X}$, we can create an algorithm that select values of θ that maximize that probability for all data. I am first going to state the log probability function and partial derivatives with respect to θ. Then later we will (a) show an algorithm that can chose optimal values of θ and (b) show how the equations were derived.

An important realization is that given the best values for the parameters (θ), logistic regression often can do a great job of estimating the probability of different class labels. However, given bad—or even random—values of θ, it does a poor job. The amount of ''intelligence'' that your logistic machine laerning algorithm has is dependent on having good values of θ.

25.1.2 Log Likelihood

In order to choose values for the parameters of logistic regression, we use Maximum Likelihood Estimation (MLE). As a result, we will have two steps: (1) Write the log-likelihood function, and (2) find the values of θ that maximize the log-likelihood function.

The labels that we are predicting are binary, and the output of our logistic regression function is $P(Y = 1 \mid \mathbf{X} = \mathbf{x})$. This means that we can (and should) interpret the Logistic Regression model as a Bernoulli random variable: $Y \mid \mathbf{X} = \mathbf{x} \sim \text{Ber}(p)$, where $p = \sigma(\theta^\top \mathbf{x})$.

To start, here is a super slick way of writing the probability of one datapoint (recall that this is the non-piecewise form of writing the probability mass function of a Bernoulli random variable):

$$P(Y = y \mid \mathbf{X} = \mathbf{x}) = \sigma(\theta^\top \mathbf{x})^y \cdot [1 - \sigma(\theta^\top \mathbf{x})]^{(1-y)}$$

$$P(Y = y \mid \mathbf{X} = \mathbf{x}) = \begin{cases} \sigma(\theta^\top \mathbf{x}) & \text{if } y = 1 \\ 1 - \sigma(\theta^\top \mathbf{x}) & \text{if } y = 0 \end{cases}$$

Now that we know the probability mass function of a single datapoint, we can write the conditional likelihood of all the data, where each datapoint is independent:

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} P(Y = y^{(i)} \mid \mathbf{X} = \mathbf{x}^{(i)}) \\
&= \prod_{i=1}^{n} \sigma(\theta^\top \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left[1 - \sigma(\theta^\top \mathbf{x}^{(i)})\right]^{(1-y^{(i)})}
\end{aligned}$$

And if you take the log of this function, you get the log-conditional likelihood of the training dataset for Logistic Regression.

Side Note: While we calculate conditional likelihood here, it is worthwhile noting that maximizing log-likelihood and log-conditional likelihood are equivalent:

$$\begin{aligned}
LL(\theta) &= \sum_{i=1}^{n} \log(f(\mathbf{x}^{(i)}, y^{(i)} \mid \theta)) = \sum_{i=1}^{n} \log(f(\mathbf{x}^{(i)} \mid \theta) P(y^{(i)} \mid \mathbf{x}^{(i)}, \theta)) && \text{(Chain rule)} \\
&= \sum_{i=1}^{n} \log(f(\mathbf{x}^{(i)}) f(y^{(i)} \mid \mathbf{x}^{(i)}, \theta)) && (\mathbf{X}, \theta \text{ independent}) \\
\theta_{MLE} = \arg\max_{\theta} LL(\theta) &= \arg\max_{\theta} \left(\sum_{i=1}^{n} \log f(\mathbf{x}^{(i)}) + \log f(y^{(i)} \mid \mathbf{x}^{(i)}, \theta) \right) && \text{(Log of products)} \\
&= \arg\max_{\theta} \left(\sum_{i=1}^{n} \log f(y^{(i)} \mid \mathbf{x}^{(i)}, \theta) \right) && (\text{Constants w.r.t. } \theta)
\end{aligned}$$

25.1.3 Gradient of Log Likelihood

In MLE, now that we have a function for log-likelihood, we simply need to chose the values of θ that maximize it. We can find the best values of θ by using an optimization algorithm. However, in order to use an optimization algorithm, we first need to know the partial derivative of log-likelihood with respect to each parameter. First I am going to give you the partial derivative (so you can see how it is used). Then I am going to show you how to derive it.

The partial derivative of log-likelihood with respect to each parameter θ_j is:

$$\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^\top \mathbf{x}^{(i)})\right] x_j^{(i)}$$

25.2 Parameter Estimation using Gradient Ascent Optimization

Once we have an equation for Log Likelihood, we chose the values for our parameters (θ) that maximize said function. In the case of logistic regression we can't solve for θ mathematically. Instead we use a computer to chose θ.

To do so we employ an algorithm called gradient ascent. That algorithms claims that if you continuously take small steps in the direction of your gradient, you will eventually make it to a local maxima. In the case of Logistic Regression you can prove that the result will always be a global maxima.

The small step that we continually take given the training dataset can be calculated as follows:

$$\begin{aligned}\theta_j^{\text{new}} &= \theta_j^{\text{old}} + \eta \cdot \frac{\partial LL(\theta^{\text{old}})}{\partial \theta_j^{\text{old}}} \\ &= \theta_j^{\text{old}} + \eta \cdot \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^{\text{old}^\top} \mathbf{x}^{(i)}) \right] x_j^{(i)},\end{aligned}$$

where η is the magnitude of the step size that we take. If you keep updating θ using the equation above you will converge on the best values of θ!

Pro-tip: Don't forget that in order to learn the value of θ_0 , you can simply define $\mathbf{x}_0 = 1$ for all datapoints.

25.3 Derivations

In this section we provide the mathematical derivations for the log-likelihood function and the gradient. The derivations are worth knowing because these ideas are heavily used in Artificial Neural Networks.

Our goal is to calculate the derivative of the log likelihood with respect to each theta. To start, here is the definition for the derivative of sigma with respect to its inputs:

$$\frac{\partial}{\partial z}\sigma(z) = \sigma(z)[1 - \sigma(z)] \qquad \text{(to get the derivative with respect to } \theta\text{, use the chain rule)}$$

Derivative of gradient for one datapoint $(\mathbf{x}, y)$:

$$
\begin{aligned}
\frac{\partial LL(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} y \log \sigma(\theta^\top \mathbf{x}) + \frac{\partial}{\partial \theta_j} (1-y) \log[1-\sigma(\theta^\top \mathbf{x})] && \text{(derivative of sum of terms)} \\
&= \left[\frac{y}{\sigma(\theta^\top \mathbf{x})} - \frac{1-y}{1-\sigma(\theta^\top \mathbf{x})}\right] \frac{\partial}{\partial \theta_j} \sigma(\theta^\top \mathbf{x}) && \text{(derivative of } \log f(x)\text{)} \\
&= \left[\frac{y}{\sigma(\theta^\top \mathbf{x})} - \frac{1-y}{1-\sigma(\theta^\top \mathbf{x})}\right] \sigma(\theta^\top \mathbf{x})[1-\sigma(\theta^\top \mathbf{x})] x_j && \text{(chain rule + derivative of sigma)} \\
&= \left[\frac{y-\sigma(\theta^\top \mathbf{x})}{\sigma(\theta^\top \mathbf{x})[1-\sigma(\theta^\top \mathbf{x})]}\right] \sigma(\theta^\top \mathbf{x})[1-\sigma(\theta^\top \mathbf{x})] x_j && \text{(algebraic manipulation)} \\
&= [y-\sigma(\theta^\top \mathbf{x})] x_j && \text{(cancelling terms)}
\end{aligned}
$$

Because the derivative of sums is the sum of derivatives, the gradient of theta is simply the sum of this term for each training datapoint.

$$
\sum_{i=1}^{n} [y^{(i)} - \sigma(\theta^\top \mathbf{x}^{(i)})] x_j^{(i)}
$$

A Review

This section organizes the concepts to help navigate the material.

$$\text{PMF} = P(X = x) = p_X(x)$$
$$\text{PDF} = f(x)$$
$$\text{CDF} = F_X(x)$$

COUNTING	
Sum Rule	$\lvert A\rvert + \lvert B\rvert = m + n$
Product Rule	$\lvert A\rvert\lvert B\rvert = mn$
Inclusion-Exclusion	$\lvert A \cup B\rvert = \lvert A\rvert + \lvert B\rvert - \lvert A \cap B\rvert$
Floor and Ceiling	$\lfloor 1.9 \rfloor = 1$ and $\lceil 1.9 \rceil = 2$
The Pigeonhole Principle	$\lceil m/n \rceil$

COMBINATORICS	
Permutations	$n!$
Combinations (Binomial)	$\dfrac{n!}{r!(n-r)!} = \dbinom{n}{r}$
Bucketing (Multinomial)	$\dfrac{n!}{n_1!n_2!\ldots n_r!} = \dbinom{n}{n_1, n_2, \ldots, n_r}$
Divider Method	$\dfrac{(n+r-1)!}{n!(r-1)!} = \dbinom{n+r-1}{n}$

PROBABILITY	
PMF	$p_X(x) = P(X = x)$
Expectation	$\mathbb{E}[X] = \sum_{x \in X} x \cdot p_X(x)$
Variance	$\text{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

	number of successes	time until success	
one trial	$\text{Ber}(p)$	$\text{Geo}(p)$	one success
several trials	$\text{Bin}(n, p)$	$\text{NegBin}(r, p)$	several successes
interval of time	$\text{Poi}(\lambda)$	$\text{Exp}(\lambda)$	interval of time to first success

```
using Distributions

# Probability mass functions (PMFs)
Ber = Distributions.Bernoulli
Bin = Distributions.Binomial
Poi = Distributions.Poisson
Geo = Distributions.Geometric
NegBin = Distributions.NegativeBinomial

# Probability density functions (PDFs)
Uni = Distributions.Uniform
Exp = Distributions.Exponential

# Expectation
𝔼(X::Bernoulli) = X.p
𝔼(X::Binomial) = X.n*X.p
𝔼(X::Poisson) = X.λ
𝔼(X::Geometric) = 1/X.p
𝔼(X::NegativeBinomial) = X.r/X.p
𝔼(X::Uniform) = (X.a + X.b)/2
𝔼(X::Exponential) = 1/X.θ

# Variance
Var(X::Bernoulli) = X.p*(1-X.p)
Var(X::Binomial) = X.n*X.p*(1-X.p)
Var(X::Poisson) = X.λ
Var(X::Geometric) = (1-X.p)/X.p^2
Var(X::NegativeBinomial) = X.r*(1-X.p)/X.p^2
Var(X::Uniform) = (X.b-X.a)^2/12
Var(X::Exponential) = 1/X.θ^2
```

Algorithm A.1. Aliases for distributions, expectation, and variance using the `Distributions` package.

B Calculation Reference

B.1 Summation Identities

Here are some useful identities and rules related to working with summations. In the rules below, f and g are arbitrary real-valued functions.

From notes by Mehran Sahami and Lisa Yan.

Pulling a constant out of a summation:

$$\sum_{n=s}^{t} C \cdot f(n) = C \cdot \sum_{n=2}^{n} f(n), \text{ where } C \text{ is a constant.} \tag{B.1}$$

Eliminating the summation by summing over the elements:

$$\sum_{i=1}^{n} x = nx \tag{B.2}$$

$$\sum_{i=m}^{n} x = (n - m + 1)x \tag{B.3}$$

$$\sum_{i=s}^{n} f(C) = (n - s + 1) f(C), \text{ where } C \text{ is a constant.} \tag{B.4}$$

Combining related summations:

$$\sum_{n=s}^{j} f(n) + \sum_{n=j+1}^{t} f(n) = \sum_{n=s}^{t} f(n) \tag{B.5}$$

$$\sum_{n=s}^{t} f(n) + \sum_{n=s}^{t} g(n) = \sum_{n=s}^{t} [f(n) + g(n)] \tag{B.6}$$

Changing the bounds on the summation:

$$\sum_{n=s}^{t} f(n) = \sum_{n=s+p}^{t+p} f(n - p) \tag{B.7}$$

“Reversing” the order of the summation:

$$\sum_{n=a}^{b} f(n) = \sum_{n=b}^{a} f(n) \tag{B.8}$$

Arithmetic series:

$$\sum_{i=0}^{n} i = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{B.9}$$

$$\sum_{i=m}^{n} i = \frac{(n-m+1)(n+m)}{2} \tag{B.10}$$

Arithmetic series involving higher order polynomials:

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{n^3}{3} + \frac{n^2}{2} + n/6 \tag{B.11}$$

$$\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 = \frac{n^4}{4} + \frac{n^3}{2} + \frac{n^2}{4} = \left[\sum_{i=1}^{n} i\right]^2 \tag{B.12}$$

Geometric series:

$$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x} \tag{B.13}$$

$$\sum_{i=m}^{n} x^i = \frac{x^{n+1}-x^m}{x-1} \tag{B.14}$$

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \text{ if } |x| < 1 \tag{B.15}$$

More exotic geometric series:

$$\sum_{i=0}^{n} i2^i = 2 + 2^{n+1}(n-1) \tag{B.16}$$

$$\sum_{i=0}^{n} \frac{i}{2^i} = \frac{2^{n+1}-n-2}{2^n} \tag{B.17}$$

Taylor expansion of exponential function:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \tag{B.18}$$

Binomial coefficient:

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n \tag{B.19}$$

B.2 Growth Rates of Summations

Besides solving a summation explicitly, it is also worthwhile to know some general growth rates on sums, so you can (tightly) bound a sum if you are trying to prove something in the big-Oh/Theta world. If you're not familiar with big-Theta (Θ) notation, you can think of it like big-Oh notation, but it actually provides a "tight" bound. Namely, big-Theta means that the function grows *no more quickly* and *no more slowly* than the function specified, up to constant factors, so it's actually more informative than big-Oh.

Here are some useful bounds:

$$\sum_{i=1}^{n} i^c = \Theta(n^{c+1}), \text{ for } c \geq 0 \tag{B.20}$$

$$\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n) \tag{B.21}$$

$$\sum_{i=1}^{n} c^i = \Theta(c^n), \text{ for } c \geq 2 \tag{B.22}$$

B.3 Identities of Products

Recall that the mathematical symbol Π represents a product of terms (analogous to Σ representing a sum of terms). Below, we give some useful identities related to products.

Definition of factorial:[1]

$$\prod_{i=1}^{n} i = n! \tag{B.23}$$

Stirling's approximation for $n!$ is given below. This approximation is useful when computing $n!$ for large values of n (particularly when $n > 30$).

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{B.24}$$

$$\approx \sqrt{2\pi n}^{\left(n+\frac{1}{2}\right)} e^{-n} \tag{B.25}$$

[1] By definition $0! = 1$.

Eliminating the product by multiplying over the elements:

$$\prod_{i=1}^{n} C = C^n, \text{ where } C \text{ is a constant.} \tag{B.26}$$

Combining products:

$$\prod_{i=1}^{n} f(i) \cdot \prod_{i=1}^{n} g(i) = \prod_{i=1}^{n} f(i) \cdot g(i) \tag{B.27}$$

Turning products into summations (using logarithms, assuming $f(i) > 0$ for all i):

$$\log\left(\prod_{i=1}^{n} f(i)\right) = \sum_{i=1}^{n} \log f(i) \tag{B.28}$$

B.4 Calculus Review

Product Rule for derivatives:

$$d(u \cdot v) = du \cdot v + u \cdot dv \tag{B.29}$$

Derivative of exponential function:

$$\frac{d}{dx} e^u = e^u \frac{du}{dx} \tag{B.30}$$

Integral of exponential function:

$$\int e^u du = e^u \tag{B.31}$$

$$\int Ae^{Au} du = e^{Au} \tag{B.32}$$

$$\int e^{Cu} du = \frac{1}{C} e^{Cu} \tag{B.33}$$

Derivative of natural logarithm:

$$\frac{d}{dx} \ln(x) = \frac{1}{x} \tag{B.34}$$

Integral of $\frac{1}{x}$:

$$\int \frac{1}{x} dx = \ln(x) \tag{B.35}$$

Integration by parts (everyone's favorite!):

Choose a suitable u and dv to decompose the integral of interest:

$$\int u \cdot dv = u \cdot v - \int v \cdot du \tag{B.36}$$

Here's the underlying rule that integration by parts is derived from:

$$\int d(u \cdot v) = u \cdot v = \int du \cdot v + \int u \cdot dv \tag{B.37}$$

B.5 Computing Permutations and Combinations

For your problem set solutions it is fine for your answers to include factorials, exponentials, or combinations; you don't need to calculate those all out to get a single numeric answer. However, if you'd like to work with those in Python or Julia, here are a few functions you may find useful.

Computes	Python*	Julia
$n!$	`math.factorial(n)`	`factorial(n)`
$\binom{n}{k}$	`scipy.special.binom(n, k)`	`binomial(n, k)`
e^n	`math.exp(n)`	`exp(n)`
n^m	`n ** m`	`n^m`

* Names to the left of the dots (`.`) are modules that need to be imported before being used: `import math, scipy.special`.

http://en.wikipedia.org/wiki/Summation

Problem Set 1, Combinatorics

PSET1 Q1. How many ways can 10 people be seated in a row if

a. there are no restrictions on the seating arrangement?

b. persons A and B must sit next to each other?

c. there are 5 adults and 5 children, and no two adults nor two children can sit next to each other?

d. there are 5 married couples and each couple must sit together?

Answer.

a.

b.

c.

d.

PSET1 Q2. At the local zoo, a new exhibit consisting of 3 different species of birds and 3 different species of reptiles is to be formed from a pool of 8 bird species and 6 reptile species. How many exhibits are possible if

a. there are no additional restrictions on which species can be selected?

b. 2 particular bird species cannot be placed together (e.g., they have a predator-prey relationship)?

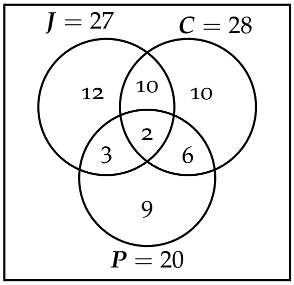
c. 1 particular bird species and 1 particular reptile species cannot be placed together?

Answer.

a.

b.

c.



PSET1 Q3. A university is offering 3 programming classes: one in Java, one in C++, and one in Python. The classes are open to any of the 100 students at the university. There are: a total of 27 students in the Java class; a total of 28 students in the C++ class; a total of 20 students in the Python class; 12 students in both the Java and C++ classes (note: these students are also counted as being in each class in the numbers above); 5 students in both the Java and Python classes; 8 students in both the C++ and Python classes; and 2 students in all three classes (note: these students are also counted as being in each pair of classes).

a. If a student is chosen randomly at the university, what is the probability that the student is not in any of the 3 programming classes?

b. If a student is chosen randomly at the university, what is the probability that the student is taking *exactly one* of the three programming classes?

c. If two different students are chosen randomly at the university, what is the probability that at least one of the chosen students is taking at least one of the programming classes?

Answer.

a.

b.

c.

NAME: EMAIL:

PSET1 Q4. Say you have \$20 million that must be invested among 4 possible companies. Each investment must be in integral units of \$1 million, and there are minimal investments that need to be made if one is to invest in these companies. The minimal investments are \$1, \$2, \$3, and \$4 million dollars, respectively for company 1, 2, 3, and 4. How many different investment strategies are available if

a. an investment must be made in each company?

b. investments must be made in at least 3 of the 4 companies?

c. Now assume that we do not have a minimal investment in any of the companies. How many different investment strategies are available if we must invest less than or equal to \$$k$ million dollars total among the 4 companies, where k is a non-negative integer less than or equal to 20 (i.e. $0 \leq kle20$)? Note that you can think of k as a constant that can be used in your answer.

Answer.

a.

b.

c.

PSET1 Q5. Consider an array x of integers with k elements (e.g., `int x[k]`), where each entry in the array has a distinct integer value between 1 and n, inclusive, and the array is sorted in increasing order. In other words, $1 \leq$ `x[i]` $\leq n$, for all $i = 0, 1, 2, \ldots, k-1$, and the array is sorted, so `x[0]` $<$ `x[1]` $< \ldots <$ `x[k-1]`. How many such sorted arrays are possible?

Answer.

PSET1 Q6. If we assume that all possible poker hands (comprised of 5 cards from a standard 52 card deck) are equally likely, what is the probability of being dealt:

a. a flush? (A hand is said to be a flush if all 5 cards are of the same suit. Note that this definition means that *straight flushes* (five cards of the same suit in numeric sequence) are also considered flushes.)

b. two pairs? (This occurs when the cards have numeric values a, a, b, b, c, where a, b and c are all distinct.)

c. four of a kind? (This occurs when the cards have numeric values a, a, a, a, b, where a and b are distinct.)

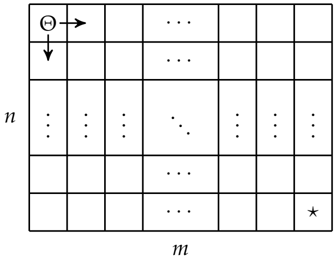
Answer.

a.

b.

c.

NAME: EMAIL:



PSET1 Q7. Imagine you have a robot (Θ) that lives on an $n \times m$ grid (it has n rows and m columns):

The robot starts in cell $(1, 1)$ and can take steps either to the right or down (**no left or up steps**). How many distinct paths can the robot take to the destination ($\star$) in cell (n, m) if

a. there are no additional constraints?

b. the robot must start by moving to the right?

c. the robot changes direction exactly 3 times? Moving down two times in a row is not changing directions, but switching from moving down to moving right is. For example, moving [down, right, right, down] would count as having two direction changes.

Answer.

a.

b.

c.

PSET1 Q8. Say we roll a six-sided die six times. What is the probability that

a. we will roll two different numbers *thrice* (three times) each?

b. we will roll *exactly one* number *exactly* three times? Hint: Be careful of overcounting.

Answer.

a.

b.

PSET1 Q9. A binary string containing M `0`'s and N `1`'s (in arbitrary order, where all orderings are equally likely) is sent over a network. What is the probability that the first r bits of the received message contain exactly k `1`'s?

Answer.

PSET1 Q10. Say we send out a total of 20 distinguishable emails to 12 distinct users, where each email we send is equally likely to go to any of the 12 users (note that it is possible that some users may not actually receive any email from us). What is the probability that the 20 emails are distributed such that there are 4 users who receive exactly 2 emails each from us and 3 users who receive exactly 4 emails each from us?

Answer.

PSET1 Q11. Say a hacker has a list of n distinct password candidates, only one of which will successfully log her into a secure system.

a. If she tries passwords from the list at random, deleting those passwords that do not work, what is the probability that her first successful login will be (exactly) on her k-th try?

b. Now say the hacker tries passwords from the list at random, but does **not** delete previously tried passwords from the list. She stops after her first successful login attempt. What is the probability that her first successful login will be (exactly) on her k-th try?

Answer.

a.

b.

PSET1 Q12. Suppose that m strings are hashed (randomly) into N buckets, assuming that all N^m arrangements are equally likely. Find the probability that exactly k strings are hashed to the first bucket.

Answer.

PSET1 Q13. [Extra credit] To get good performance when working with binary search trees (BST), we must consider the probability of producing completely degenerate BSTs (where each node in the BST has at most one child). See Lecture Notes #2, Example 2, for more details on binary search trees.

a. If the integers 1 through n are inserted in arbitrary order into a BST (where each possible order is equally likely), what is the probability (as an expression in terms of n) that the resulting BST will have completely degenerate structure?

b. Using your expression from part (a), determine the smallest value of n for which the probability of forming a completely degenerate BST is less than 0.001 (i.e., 0.1%).

Answer.

NAME: EMAIL:

PSET1 Q14. [Coding] Consider a game, which uses a generator that produces independent random integers between 1 and 100, inclusive. The game starts with a sum $S = 0$. The first player adds random numbers from the generator to S until $S > 100$, at which point they record their last random number `x`. The second player continues by adding random numbers from the generator to S until $S > 200$, at which point they record their last random number `y`. The player with the highest number wins; e.g., if `y` $>$ `x`, the second player wins. Write a Python 3 program to simulate 100,000 games and output the estimated probability that the second player wins.

Answer.

Problem Set 2, Probability

PSET2 Q1. Say in Silicon Valley, 35% of engineers program in Java and 28% of the engineers who program in Java also program in C++. Furthermore, 40% of engineers program in C++.

a. What is the probability that a randomly selected engineer programs in Java and C++?

b. What is the conditional probability that a randomly selected engineer programs in Java given that they program in C++?

Answer.

a.

b.

PSET2 Q2. A website wants to detect if a visitor is a robot or a human. They give the visitor five CAPTCHA tests that are hard for robots but easy for humans. If the visitor fails one of the tests, they are flagged as a robot. The probability that a human succeeds at a single test is 0.95, while a robot only succeeds with probability 0.3. Assume all tests are independent. The percentage of visitors on this website that are robots is 5%; all other visitors are human.

a. If a visitor is actually a robot, what is the probability they get flagged (the probability they fail at least one test)?

b. If a visitor is human, what is the probability they get flagged?

c. Suppose a visitor gets flagged. Using your answers from part (a) and (b), what is the probability that the visitor is a robot?

d. If a visitor is human, what is the probability that they pass exactly three of the five tests?

e. Building off of your answer from part (d), what is the probability that a visitor with unknown identity passes exactly three of the five tests?

Answer.

a.

b.

c.

d.

e.

PSET2 Q3. Say all computers either run operating system W or X. A computer running operating system W is twice as likely to get infected with a virus as a computer running operating system X. If 70% of all computers are running operating system W, what percentage of computers infected with a virus are running operating system W?

Answer.

PSET2 Q4. The Superbowl institutes a new way to determine which team receives the kickoff first. The referee chooses with equal probability one of three coins. Although the coins look identical, they have probability of heads 0.1, 0.5 and 0.9, respectively. Then the referee tosses the chosen coin 3 times. If more than half the tosses come up heads, one team will kick off; otherwise, the other team will kick off. If the tosses resulted in the sequence H, T, H, what is the probability that the fair coin was actually used?

Answer.

PSET2 Q5. After a long night of programming, you have built a powerful, but slightly buggy, email spam filter. When you don't encounter the bug, the filter works very well, always marking a spam email as **SPAM** and always marking a non-spam email as **GOOD**. Unfortunately, your code contains a bug that is encountered 10% of the time when the filter is run on an email. When the bug is encountered, the filter always marks the email as **GOOD**. As a result, emails that are actually spam will be erroneously marked as **GOOD** when the bug is encountered. Let p denote the probability that an email is actually non-spam, and let q denote the conditional probability that an email is non-spam given that it is marked as **GOOD** by the filter.

a. Determine q in terms of p.

b. Using your answer from part (a), explain mathematically whether q or p is greater. Also, provide an intuitive justification for your answer.

Answer.

a.

b.

PSET2 Q6. Two cards are randomly chosen without replacement from an ordinary deck of 52 cards. Let E be the event that both cards are **Aces**. Let F be the event that the **Ace of Spades** is one of the chosen cards, and let G be the event that at least one **Ace** is chosen.

a. Compute $P(E \mid F)$.

b. Are E and F independent? Justify your answer using your response to part (a).

c. Compute $P(E \mid G)$.

Answer.

a.

b.

c.

PSET2 Q7. Your colleagues in a comp-bio lab have sequenced DNA from a large population in order to understand how a gene (G) influences two particular traits $(T_1$ and $T_2)$. They find that $P(G) = 0.6$, $P(T_1 \mid G) = 0.7$, and $P(T_2 \mid G) = 0.9$. They also observe that if a subject does not have the gene G, they express neither T_1 nor T_2. The probability of a patient having both T_1 and T_2 given that they have the gene G is 0.63.

a. Are T_1 and T_2 conditionally independent given G?

b. Are T_1 and T_2 conditionally independent given G^c?

c. What is $P(T_1)$?

d. What is $P(T_2)$?

e. Are T_1 and T_2 independent?

Answer.

a.

b.

c.

d.

e.

PSET2 Q8. The color of a person's eyes is determined by a pair of eye-color genes, as follows:

- if both of the eye-color genes are **blue**-eyed genes, then the person will have **blue** eyes
- if one or more of the genes is a **brown**-eyed gene, then the person will have **brown** eyes

A newborn child independently receives one eye-color gene from each of its parents, and the gene it receives from a parent is equally likely to be either of the two eye-color genes of that parent. Suppose William and both of his parents have **brown** eyes, but William's sister (Claire) has **blue** eyes. (We assume that **blue** and **brown** are the only eye-color genes.)

a. What is the probability that William possesses a **blue**-eyed gene?

b. Suppose that William's wife has **blue** eyes. What is the probability that their first child will have **blue** eyes?

Answer.

a.

b.

PSET2 Q9. Consider the following algorithm for betting in roulette. At each round (''spin''), you bet \$1 on a color (''**red**'' or ''**black**''). If that color comes up on the wheel, you keep your bet AND win \$1; otherwise, you lose your bet.

i. Bet \$1 on ''**red**''

ii. If ''**red**'' comes up on the wheel (with probability 18/38), then you win \$1 (and keep your original \$1 bet) and you **immediately** quit (i.e., you do not do step (iii) below).

iii. If ''**red**'' did not come up on the wheel (with probability 20/38), then you lose your initial \$1 bet. But, then you bet \$1 on ''**red**'' on *each* of the next **two** spins of the wheel. After those two spins, you quit (no matter what the outcome of the next two spins).

Let X denote your ''winnings'' when you quit, i.e., the total amount of money won minus any amounts lost while playing. This value may be negative.

a. Determine $P(X > 0)$.

b. Determine $\mathbb{E}[X]$. (Rhetorical question: Would you play this game?)

Answer.

a.

b.

PSET2 Q10.

c. For each gene i, decide whether or not you think that is would be reasonable to assume that G_i is independent of T. Support your argument with numbers. Remember that our probabilities are based on 100,000 bats, not infinite bats, and are therefore only estimates of the true probabilities.

d. Give your best interpretation of the results from (a) to (c).

e. **[Extra Credit]** Try and find conditional independence relationships between the genes and the trait. Incorporate this information to improve your hypothesis of how the five genes relate to whether or not a bat can carry Ebola.

Answer.

c.

d.

e.

PSET2 Q11. [Extra Credit] Suppose we want to write an algorithm `fairRandom` for randomly generating a `0` or a `1` with equal probability ($= 0.5$). Unfortunately, all we have available to us is a function:

```
unknownRandom()::Int
```

that randomly generates bits, where on each call a `1` is returned with some unknown probability p that need not be equal to 0.5 (and a `0` is returned with probability $1 - p$).

Consider the following algorithms for `fairRandom` and `simpleRandom`:

```
function fairRandom()
    $r_1$, $r_2$ = 0, 0
    while true
        $r_1$ = unknownRandom()
        $r_2$ = unknownRandom()
        if ($r_1 \neq r_2$) break; end
    end
    return $r_2$
end
```

```
function simpleRandom()
    $r_1$, $r_2$ = 0, 0
    $r_1$ = unknownRandom()
    while true
        $r_2$ = unknownRandom()
        if ($r_1 \neq r_2$) break; end
    end
    return $r_2$
end
```

a. Show mathematically that `fairRandom` does indeed return a `0` or a `1` with equal probability.

b. Say we want to simplify the function, so we write the `simpleRandom` function. Would the `simpleRandom` function also generate `0`'s and `1`'s with equal probability? Explain why or why not. In addition, determine $P($`simpleRandom` returns `1`$)$ in terms of p.

Answer.

a.

b.

Problem Set 3, Random Variables

PSET3 Q2. Lyft line gets 2 requests every 5 minutes, on average, for a particular route. A user requests the route and Lyft commits a car to take her. All users who request the route in the next five minutes will be added to the car as long as the car has space. The car can fit up to three users. Lyft will make $7 for each user in the car (the revenue) minus $9 (the operating cost).

a. How much does Lyft expect to make from this trip?

b. Lyft has one space left in the car and wants to wait to get another user. What is the probability that another user will make a request in the next 30 seconds?

Answer.

a.

b.

PSET3 Q3. Suppose it takes at least 9 votes from a 12-member jury to convict a defendant. Suppose also that the probability that a juror votes that an actually guilty person is innocent is 0.25, whereas the probability that the juror votes that an actually innocent person is guilty is 0.15. If each juror acts independently and if 70% of defendants are actually guilty, find the probability that the jury renders a correct decision. Also determine the percentage of defendants found guilty by the jury.

Answer.

PSET3 Q4. To determine whether they have measles, 1000 people have their blood tested. However, rather than testing each individual separately (1000 tests is quite costly), it is decided to use a *group testing* procedure:

- Phase 1: First, place people into groups of 5. The blood samples of the 5 people in each group will be pooled and analyzed together. If the test is positive (at least one person in the pool has measles), continue to Phase 2. Otherwise send the group home. 200 of these pooled tests are performed.
- Phase 2: Individually test each of the 5 people in the group. 5 of these individual tests are performed per group in Phase 2.

Suppose that the probability that a person has measles is 5% for all people, independently of others, and that the test has a 100% true positive rate and 0% false positive rate (note that this is unrealistic). Using this strategy, compute the expected total number of blood tests (individual and pooled) that we will have to do across Phases 1 and 2.

Answer.

PSET3 Q5. Let X be a continuous random variable with probability density function:

$$f(x) = \begin{cases} c(2-2x^2) & -1 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

a. What is the value of c?

b. What is the cumulative distribution function (CDF) of X?

c. What is $\mathbb{E}[X]$?

Answer.

a.

b.

c.

PSET3 Q6. The number of times a person's computer crashes in a month is a Poisson random variable with $\lambda = 7$. Suppose that a new operating system patch is released that reduces the Poisson parameter to $\lambda = 2$ for 80% of computers, and for the other 20% of computers the patch has no effect on the rate of crashes. If a person installs the patch, and has their computer crash 4 times in the month thereafter, how likely is it that the patch has had an effect on the user's computer (i.e., it is one of the 80% of computers that the patch reduces crashes on)?

Answer.

PSET3 Q7. Say there are k buckets in a hash table. Each new string added to the table is hashed into bucket i with probability p_i, where $\sum_{i=1}^{k} p_i = 1$. If n strings are hashed into the table, find the expected number of buckets that have at least one string hashed to them. (Hint: Let X_i be a binary variable that has the value 1 when there is at least one string hashed into bucket i after the n strings are added to the table (and 0 otherwise). Compute $\mathbb{E}\left[\sum_{i=1}^{k} X_i\right]$.)

Answer.

NAME: EMAIL:

PSET3 Q8. You are testing software and discover that your program has a non-deterministic bug that causes catastrophic failure (aka a ''hindenbug''). Your program was tested for 400 hours and the bug occurred **twice**.

a. Each user uses your program to complete a three hour long task. If the hindenbug manifests they will immediately stop their work. What is the probability that the bug manifests for a given user?

b. Your program is used by one million users. Use a Normal approximation to estimate the probability that more than 10,000 users experience the bug. Use your answer from part (a). Provide a numeric answer for this part.

Answer.

a.

b.

PSET3 Q9. Say the lifetimes of computer chips produced by a certain manufacturer are normally distributed with parameters $\mu = 1.5 \times 10^6$ hours and $\sigma = 9 \times 10^5$ hours. The lifetime of each chip is independent of the other chips produced.

a. What is the approximate probability that a batch of 100 chips will contain at least 6 whose lifetimes are more than 3.0×10^6 hours?

b. What is the approximate probability that a batch of 100 chips will contain at least 65 whose lifetimes are less than 1.9×10^6 hours? Provide a numeric answer for this part.

Answer.

a.

b.

PSET3 Q10. A Bloom filter is a probabilistic implementation of the *set* data structure, an unordered collection of unique objects. In this problem we are going to look at it theoretically. Our Bloom filter uses 3 different independent hash functions H_1, H_2, H_3 that each take any string as input and each return an index into a bit-array of length n. Each index is equally likely for each hash function.

To add a string into the set, feed it to each of the 3 hash functions to get 3 array positions. Set the bits at all these positions to 1. For example, initially all values in the bit-array are zero. In this example $n = 10$:

Index:	0	1	2	3	4	5	6	7	8	9
Value:	0	0	0	0	0	0	0	0	0	0

After adding a string "pie", where $H_1(\text{"pie"}) = 4$, $H_2(\text{"pie"}) = 7$, and $H_3(\text{"pie"}) = 8$:

Index:	0	1	2	3	4	5	6	7	8	9
Value:	0	0	0	0	1	0	0	1	1	0

Bits are never switched back to 0. Consider a Bloom filter with $n = 9{,}000$ buckets. You have added $m = 1{,}000$ strings to the Bloom filter. Provide a **numerical answer** for all questions.

a. What is the (approximated) probability that the first bucket has 0 strings hashed to it?

To *check* whether a string is in the set, feed it to each of the 3 hash functions to get 3 array positions. If any of the bits at these positions is 0, the element is not in the set. If all bits at these positions are 1, the string *may* be in the set; but it could be that those bits are 1 because some of the other strings hashed to the same values. You may assume that the value of one bucket is independent of the value of all others.

Answer.

a.

b. What is the probability that a string which has *not* previously been added to the set will be misidentified as in the set? That is, what is the probability that the bits at all of its hash positions are already 1? Use approximations where appropriate.

Answer.

b.

c. Our Bloom filter uses three hash functions. Was that necessary? Repeat your calculation in (b) assuming that we only use a single hash function (not 3).

Answer.

c.

(Chrome uses a Bloom filter to keep track of malicious URLs. Questions such as this allow us to compute appropriate sizes for hash tables in order to get good performance with high probability in applications where we have a ballpark idea of the number of elements that will be hashed into the table.)

PSET3 Q11. Last summer (May 2019) the concentration of CO_2 in the atmosphere was 414 parts per million (ppm) which is substantially higher than the pre-industrial concentration: 275 ppm. CO_2 is a greenhouse gas and as such increased CO_2 corresponds to a warmer planet.

Absent some pretty significant policy changes, we will reach a point within the next 50 years (i.e., well within your lifetime) where the CO_2 in the atmosphere will be double the pre-industrial level. In this problem we are going to explore the following question: What will happen to the global temperature if atmospheric CO_2 doubles?

The measure, in degrees Celsius, of how much the global average surface temperature will change (at the point of equilibrium) after a doubling of atmospheric CO_2 is called "Climate Sensitivity." Since the earth is a complicated ecosystem climate scientists model Climate Sensitivity as a random variable, S. The IPPC Fourth Assessment Report had a summary of 10 scientific studies that estimated the PDF of S:

In this problem we are going to treat S as part-discrete and part-continuous. For values of S less than 7.5, we are going to model sensitivity as a discrete random variable with PMF based on the average of estimates from the studies in the IPCC report. Here is the PMF for S in the range 0 through 7.5:

Sensitivity, S (degrees C)	0	1	2	3	4	5	6	7
Expert Probability	0.00	0.11	0.26	0.22	0.16	0.09	0.06	0.04

The IPCC fifth assessment report notes that there is a non-negligible chance of S being greater than 7.5 degrees but didn't go into detail about probabilities. In the paper "Fat-Tailed Uncertainty in the Economics of Catastrophic Climate Change" Martin Weitzman discusses how different models for the PDF of Climate Sensitivity (S) for large values of S have wildly different policy implications.

For values of S greater than or equal to 7.5 degrees Celsius, we are going to model S as a continuous random variable. Consider two different assumptions for S when it is at least 7.5 degrees Celsius: a fat tailed distribution (f_1) and a thin tailed distribution (f_2):

$$f_1(x) = \frac{K}{x} \text{ s.t. } 7.5 \leq x < 30$$

$$f_2(x) = \frac{K}{x^3} \text{ s.t. } 7.5 \leq x < 30$$

For this problem assume that the probability that S is greater than 30 degrees Celsius is 0.

a. Compute the probability that Climate Sensitivity is at least 7.5 degrees Celsius.

> ***Answer.***
>
> a.

b. Calculate the value of K for both f_1 and f_2.

> ***Answer.***
>
> b.

c. It is estimated that if temperatures rise more than 10 degrees Celsius, all the ice on Greenland will melt. Estimate the probability that S is greater than 10 under both the f_1 and f_2 assumptions.

> ***Answer.***
>
> c.

d. Calculate the expectation of S under both the f_1 and f_2 assumptions.

> ***Answer.***
>
> d.

e. Let $R = S^2$ be a crude approximation of the cost to society that results from S. Calculate $\mathbb{E}[R]$ under both the f_1 and f_2 assumptions.

> ***Answer.***
>
> e.

Notes: (1) Both f_1 and f_2 are ''power law distributions''. (2) Calculating expectations for a variable that is part discrete and part continuous is as simple as: use the discrete formula for the discrete part and the continuous formula for the continuous part.

PSET3 Q12. **[Extra Credit]** Say we have a cable of length n. We select a point (chosen uniformly randomly) along the cable, at which we cut the cable into two pieces. What is the probability that the shorter of the two pieces of the cable is less than 1/3 the size of the longer of the two pieces? Explain formally how you derived your answer.

Answer.

Problem Set 4, Distributions

PSET4 Q1. The **median** of a continuous random variable having cumulative distribution function F is the value m such that $F(m) = 0.5$. That is, a random variable is just as likely to be larger than its median as it is to be smaller. Find the median of X (in terms of the respective distribution parameters) in each case below.

a. $X \sim \text{Uni}(a, b)$

b. $X \sim \mathcal{N}(\mu, \sigma^2)$

c. $X \sim \text{Exp}(\lambda)$

Answer.

a.

b.

c.

PSET4 Q2. Users independently sign up for two online social networking sites, Lookbook and Quickgram. On average, 7.5 users sign up for Lookbook each minute, while on average 5.5 users sign up for Quickgram each minute. The number of users signing up for Lookbook and for Quickgram each minute are independent. A new user is defined as a new account, i.e., the same person signing up for both social networking sites will count as two new users.

a. What is the probability that more than 10 new users will sign up for the Lookbook social networking site in the next minute?

b. What is the probability that more than 13 new users will sign up for the Quickgram social networking site in the next 2 minutes?

c. What is the probability that the company will get a combined total of more than 40 new users across both websites in the next 2 minutes?

Answer.

a.

b.

c.

PSET4 Q3. Say that of all the students who will attend Stanford, each will buy at most one laptop computer when they first arrive at school. 40% of students will purchase a PC, 30% will purchase a Mac, 10% will purchase a Linux machine and the remaining 20% will not buy any laptop at all. If 15 students are asked which, if any, laptop they purchased, what is the probability that exactly 6 students will have purchased a PC, 4 will have purchased a Mac, 2 will have purchased a Linux machine, and the remaining 3 students will have not purchased any laptop?

Answer.

PSET4 Q4. Say we have two independent variables X and Y, such that $X \sim \text{Geo}(p)$ and $Y \sim \text{Geo}(p)$. Mathematically derive an expression for $P(X = k \mid X + Y = n)$, where k and n are non-negative integers.

Answer.

PSET4 Q5. Choose a number X at random from the set of numbers $\{1, 2, 3, 4, 5\}$. Now choose a number at random from the subset no larger than X, that is from $\{1, \ldots, X\}$. Let Y denote the second number chosen.

a. Determine the joint probability mass function of X and Y.

b. Determine the conditional mass function of X given $Y = i$. Do this for $i = 1, 2, 3, 4, 5$.

c. Are X and Y independent? Justify your answer.

Answer.

a.

b.

c.

PSET4 Q6. Let $X_1, X_2, \ldots$ be a series of independent random variables which all have the same mean μ and the same variance σ^2. Let $Y_n = X_n + X_{n+1}$. For $j = 0, 1,$ and 2, determine $\text{Cov}(Y_n, Y_{n+j})$. Note that you may have different cases for your answer depending on the value of j.

Answer.

PSET4 Q7. Our ability to fight contagious diseases depends on our ability to model them. One person is exposed to llama-flu. The method below models the number of individuals who will get infected.

```
from scipy import stats
"""
Return number of people infected by one individual.
"""
def num_infected():
  # most people are immune to llama flu.
  # stats.bernoulli(p).rvs() returns 1 w.p. p (0 otherwise)
  immune = stats.bernoulli(p = 0.99).rvs()
  if immune: return 0

  # people who are not immune spread the disease far by
  # making contact with k people (up to 100).
  spread = 0
  # returns random # of successes in n trials w.p. p of success
  k = stats.binom(n = 100, p = 0.25).rvs()
  for i in range(k):
    spread += num_infected()

  # total infections will include this individual
  return spread + 1
```

What is the expected return value of `numInfected()`?

Answer.

PSET4 Q8. In class, we considered the following recursive function:

```
def recurse():
    x = np.random.choice([1,2,3])   # equally likely values 1,2,3
    if (x == 1): return 3
    elif (x == 2): return (5 + recurse())
    else: return (7 + recurse())
```

Let Y = the value returned by `recurse()`. We previously computed $\mathbb{E}[Y] = 15$. What is $\text{Var}(Y)$?

Answer.

PSET4 Q9. You go on a camping trip with two friends who each have a mobile phone. Since you are out in the wilderness, mobile phone reception isn't very good. One friend's phone will independently drop calls with 20% probability. Your other friend's phone will independently drop calls with 30% probability. Say you need to make 6 phone calls, so you randomly choose one of the two phones and you will use that *same* phone to make all your calls (but you don't know which has a 20% versus 30% chance of dropping calls). Of the first 3 (out of 6) calls you make, one of them is dropped. What is the conditional expected number of dropped calls in the 6 total calls you make (conditioned on having already had one of the first three calls dropped)?

Answer.

PSET4 Q11. [Extra Credit] Consider a bit string of length n, where each bit is independently generated and has probability p of being a 1. We say that a *bit switch* occurs whenever a bit differs from the one preceding it in the string (if there is a preceding bit). For example, if $n = 5$ and we have the bit string `11010`, then there are 3 bit switches. Find the expected number of bit switches in a string of length n. (Hint: You might find it helpful to use a set of indicator (Bernoulli) variables that are defined in terms of whether a bit switch occurred in each *position* of the string. And in case you're wondering why we care about bit switches, the number of bit switches in a string can be one indicator of how compressible that string might be—for example, if the bit string represented a file that we were trying to ZIP.)

Answer.

Problem Set 5, Sampling

PSET5 Q1. The joint probability density function of continuous random variables X and Y is given by:

$$f_{X,Y}(x,y) = c\frac{y}{x} \qquad \text{where } 0 < y < x < 1$$

a. What is the value of c in order for $f_{X,Y}(x,y)$ to be a valid probability density function?

Answer.

a.

Parts (c,b,d,e) on next pages...

...part (a) on previous page.

b. Are X and Y independent? Explain why or why not.

Answer.

b.

Parts (c,d,e) on next pages...

...parts (a,b) on previous pages.

c. What is the marginal density function of X?

Answer.

c.

d. What is the marginal density function of Y?

Answer.

d.

e. What is $\mathbb{E}[X]$?

Answer.

e.

PSET5 Q2. A robot is located at the *center* of a square world that is 10 kilometers on each side. A package is dropped off in the robot's world at a point (x, y) that is uniformly (continuously) distributed in the square. If the robot's starting location is designated to be $(0, 0)$ and the robot can only move up/down/left/right parallel to the sides of the square, the distance the robot must travel to get to the package at point (x, y) is $|x| + |y|$. Let $D =$ the distance the robot travels to get to the package. Compute $\mathbb{E}[D]$.

Answer.

PSET5 Q3. Let X, Y, and Z be independent random variables, where $X \sim \mathcal{N}(\mu_1, {\sigma_1}^2)$, $Y \sim \mathcal{N}(\mu_2, {\sigma_2}^2)$, and $Z \sim \mathcal{N}(\mu_3, {\sigma_3}^2)$.

a. Let $A = X + Y$. What is the distribution (along with parameter values) of A?

> ***Answer.***
>
> a.

b. Let $B = 4X + 3$. What is the *joint* distribution (along with parameter values) of B and Z? (Hint: Bivariate Normal)

> ***Answer.***
>
> b.

Part (c) on next page...

...parts (a,b) on previous page.

c. Let $C = aX - b^2Y + cZ$, where a, b, and c are real-valued constants. What is the distribution (along with parameter values) of C? Show how you derived your answer.

Answer.

c.

PSET5 Q4. A fair 6-sided die is repeatedly rolled until the total sum of all the rolls exceeds 300. Approximate (using the Central Limit Theorem) the probability that *at least* 80 rolls are necessary to reach a sum that exceeds 300.

Answer.

NAME: EMAIL:

PSET5 Q5. Program A will run 20 algorithms in sequence, with the running time for each algorithm being independent random variables with mean = 46 seconds and variance = 100 seconds^2. Program B will run 20 algorithms in sequence, with the running time for each algorithm being independent random variables with mean = 48 seconds and variance = 200 seconds^2.

a. What is the approximate probability that Program A completes in less than 900 seconds?

Answer.

a.

Parts (b,c) on next page...

...part (a) on previous page.

b. What is the approximate probability that Program B completes in less than 900 seconds?

Answer.

b.

c. What is the approximate probability that Program A completes in less time than Program B?

Answer.

c.

PSET5 Q6.

c. **[Written]** Use your answers to part (a) and (b) and approximate X and Y as Normal random variables with mean and variance that match their biometric data. Report both distributions.

d. **[Written]** Calculate the ratio of the probability that user A wrote the email over the probability that user B wrote the email. You do not need to submit code, but you should include the formula that you attempted to calculate and a short description (a few sentences) of how your code works.

Answer.

c.

d.

PSET5 Q7.

d. **[Written]** Would you use the mean or the median of 5 peer grades to assign scores in the online version of Stanford's HCI class? Hint: it might help to visualize the scores. Feel free to write code to help you answer this question, but for this question we'll solely evaluate your written answer in the PDF that you upload to Gradescope.

Answer.

d.

PSET5 Q8.

c. **[Written]** For each of the three backgrounds, calculate a difference in means in learning outcome between `activity1` and `activity2`, and the p-value of that difference.

d. **[Written]** Your manager at Coursera is concerned that you have been "p-hacking," which is also known as data dredging: `https://en.wikipedia.org/wiki/Data_dredging`. In one sentence, explain why your results in part (c) are not the result of p-hacking.

Answer.

c.

d.

Quiz 1

I acknowledge and accept the letter and spirit of the Honor Code: ____________________

Q1. You have just been elected social chair for the student organization PRoB (Probability Revolution or Bust) for the coming academic year 2020–2021. As new social chair, you would like to hold 10 (indistinct) socials over the next 3 quarters (Autumn, Winter, and Spring). Each social is equally likely to be assigned to any of the quarters, and it is possible that a quarter has no socials. Order of socials within a quarter doesn't matter.

a. (5 points) In how many distinct ways can the 10 (indistinct) socials be allocated to the 3 quarters?

> ***Answer.***
>
> a.

b. (5 points) In how many distinct ways can the 10 (indistinct) socials be allocated to the 3 quarters if you must hold at least 2 socials each quarter?

> ***Answer.***
>
> b.

c. (3 points) Let your answer to part (a) be n. Consider the event where you hold 3 socials in Autumn, 3 socials in Winter, and 4 socials in Spring. Is the probability of this event $\frac{1}{n}$? Briefly explain in a few sentences why or why not.

> ***Answer.***
>
> c.

d. (12 points) You are also planning your courseload for next year. You have 10 courses to schedule over 3 quarters (Autumn, Winter, and Spring). All of the courses are distinct and offered every quarter; order of courses within a quarter doesn't matter. In how many distinct ways can you allocate 10 (distinct) courses to the 3 quarters if you can take at most 4 courses in any quarter?

> ***Answer.***
>
> d.

Q2. A home robot has two different sensors for motion detection. If there is a moving object, sensor V (video camera) will detect motion with probability 0.95, and sensor L (laser) will detect motion with probability 0.8. If there is no moving object, there is a 0.1 probability that sensor V will detect motion (even though there is no object), and a 0.05 probability that sensor L will detect motion.

Based on empirical evidence, the probability that there is a moving object is 0.7. Note that these sensors use independent detection algorithms to identify motion, so that **conditioned** on there being a moving object (or not), the events of detecting motion (or not) for each sensor is **independent**.

a. (3 points) Given that there is a moving object and that sensor V does not detect motion, what is the probability that sensor L detects motion? Give a numerical answer.

Answer.

a.

b. (5 points) Given that there is a moving object, what is the probability that **at least one** of the two sensors detects motion? Give a numerical answer.

Note: You can use WolframAlpha as a calculator (it also accepts LaTeX equations). Example: `https://www.wolframalpha.com/input/?i=sum+x%5Ek%2Fk%21%2C+k%3D0+to+infinity`

Answer.

b.

c. (5 points) Given that there is a moving object, what is the probability that **exactly one** of the two sensors detects motion? Give a numerical answer.

Answer.

c.

d. (8 points) What is the probability that there is a moving object given that **both** sensors detect motion? Give a numerical answer.

Answer.

d.

e. (4 points) The probabilities that sensor V detects motion given a moving object and that sensor V detects motion given no moving object do not sum up to 1. Briefly explain in a few sentences why this is okay.

Answer.

e.

Q3. You have 8 pairs of mittens, each a different pattern. Left and right mittens are also distinct. Suppose that you are fostering kittens, and you leave them alone for a few hours with your mittens. When you return, you discover that they have hidden 4 mittens! Suppose that your kittens are equally likely to hide any 4 of your 16 distinct mittens. Let X be the number of complete, distinct ***pairs*** of mittens that you have left.

a. (15 points) Compute the probability mass function of X, $p_X(x)$. (Hint: Note the support of X is $\{4, 5, 6\}$)

Answer.

a.

b. (5 points) Compute $\mathbb{E}[X]$ using the definition of expectation and your answer to (a).

Answer.

b.

c. (10 points) Define the random variable X_i to be 1 if your i^{th} pair of mittens is complete after the kitten fiasco, and 0 otherwise. Using this definition of X_i for $i = 1, \ldots, 8$ and the linearity of expectation, compute $\mathbb{E}[X]$ again. **Do not use your answer to part (a).**

Answer.

c.

Q4. A food takeout and delivery company, DashDoor (like DoorDash, but better) wants to understand how busy their employees are each month. In metropolitan areas, employees receive on average 8 customer requests each day. Regardless of where they work, each employee spends exactly 0.5 hours to make deliveries for each request (one at a time); for example, an employee who receives exactly 4 requests on a particular day will spend exactly 2 hours making deliveries.

a. (4 points) What is the variance of the number of hours that a metropolitan employee spends on making deliveries in a day?

Answer.

a.

b. (6 points) What is the probability that a metropolitan employee has a ''very busy day,'' defined as spending at least 5 hours making deliveries from requests received that day? Give a numerical answer.

Note: You can use WolframAlpha as a calculator (it also accepts LaTeX equations). Example: `https://www.wolframalpha.com/input/?i=sum+x%5Ek%2Fk%21%2C+k%3D0+to+infinity`

Answer.

b.

c. (6 points) The company estimates that 0.6 of their employees work in metropolitan areas, and the rest in suburban areas. Suburbs have different customer demand: in suburban areas, employees receive on average 6 customer requests each day. What is the probability that a randomly chosen employee has a very busy day?

Answer.

c.

d. (14 points) Consider a metropolitan employee. The event that they have a very busy day on a particular day is independent of their business on other days. Let p be your answer from part (b), the probability that they have a very busy day.

i. (6 points) What is the probability that this employee has more than 10 busy days in the next 20 working days? Leave your answer in terms of p.

Answer.

d.i.

ii. (4 points) What is the expected number of very busy days that this employee has over the next 20 working days? Leave your answer in terms of p.

Answer.
d.ii.

iii. (4 points) Would it be reasonable to approximate the probability you computed in part (d)(i) using a Poisson random variable? Briefly explain in a few sentences why or why not.

Answer.
d.iii.

References

1. M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization*. MIT Press, 2019.
2. K. Rosen, *Discrete Mathematics and Its Applications*. 2007, vol. 6.
3. S. M. Ross et al., *A First Course in Probability*. 2006, vol. 7.