Data Science: Christmas Projects

These are optional projects that you may attempt after you have completed at least 4 past papers. Each project goes in more depth at a particular area of the Data Science course. You are encouraged to use the internet and books (and you can collaborate with others if you want). You can also email me for hints. Feel free to deviate from the exact instructions (or come up with a project of your own) if you find something that interests you (and you can also give partial answers).

I don't recommend spending time (during the university year) on more than one projects. The projects are not sorted in any order. If sufficiently many of you attempt a project, you can present during the revision session.

NumPy Projects

Project 1 [NumPy Internals]

1. Read parts of this page, this page or this chapter and answer the following questions:
    1. In which language are many of the NumPy operations implemented? How does this interact with python? For which operations is it efficient?
    2. Describe the memory structure of a NumPy ndarray.
    3. Describe the implementation of a typical NumPy universal function, e.g. numpy.dot (You may want to read this).
    4. Knowing how NumPy is implemented, how can you make your code more efficient?
    5. Describe a feature in the NumPy implementation that you found interesting.
2. Read a NumPy Enhancement proposal (see this list) and answer the following questions:
    1. Describe the change being proposed.
    2. Do you think it is needed? Why?
    3. What are the challenges in implementing it?
    4. How long do you think it would take to implement?

Project 2 [NumPy core library in C++]

In this project you will implement core parts of the NumPy library in C++.

1. Describe the representation for fixed size integer nd-arrays. (*Hint:* You can implement this using a 1D array and a shape array).
2. Implement the ones, zeros and reshape functionality.
3. Implement a function to set/get the i-th element. (Optional) You can also allow multidimensional indexing of `vector<int>` (`-1` indicates the last index of the dimension and `-2` could be equivalent to `:`).
4. Implement the sum, max and min functions.
5. Implement the addition, subtraction, multiplication, exponentiation, log, mean, sin and cos operations. (You may find it nicer to implement the operations using overloading and also support operations with scalar values). Also, you may find it useful to check for errors (i.e. adding two vectors of different dimensions).
6. Implement matrix multiplication operation.

7. Implement the quantile function.
8. (Optional) Implement linspace.
9. (Optional) Implement np.where and indexing based on a Boolean vector.
10. Implement one of the supervision exercises using your library. By executing both codes multiple times, determine which one if faster. (You can also run a hypothesis test if you are interested).
11. What is missing from your implementation? What would you implement differently if you had the time?
12. What are the advantages/disadvantages of using NumPy instead of using a C++ library?

Project 3 [Internals of numpy.random]

1. Read section 4.6 and section 6.3 about generating random variables in the extended lecture notes.
2. Assume you only have access to a generator of a uniform random variables:
    1. Implement sampling from a Bernoulli distribution.
    2. Implement sampling from a Binomial distribution.
    3. Implement sampling from a Poisson distribution.
    4. Implement sampling from an Exponential distribution.
    5. Implement selecting a random permutation of the given elements.
    6. Implement resampling (i.e. np.random.choice).
    7. Implement sampling from a discrete distribution given as $p_1, \ldots, p_n$. What is the time complexity of your algorithm?
    8. (optional) Read about the Alias method on how to implement this efficiently.
3. Sampling using inverse transform sampling and binary search.
    1. Given acess to the CDF of a continuous distribution as a python function `f(x)`, describe how you would use binary search to sample efficiently.
    2. How would you sample from the Beta distribution?
4. (optional) Read about the Box-Muller transform and sample from a Normal distribution.
5. (Puzzle) How would you sample uniformly from a biased coin for which you don't know how biased it is?

Markov Chains

Project 4 [Strongly Connected Components]

During the supervisions, we briefly discussed how you would algorithmically check if a given Markov Chain is irreducible. In this project you can research and implement such algorithms.

1. Implement an algorithm based on DFS or BFS that checks if for every vertex u you can reach every other vertex v following non-zero probability edges. What is the time complexity of your implementation?
2. Define a Strongly Connected Component (SCC).
3. Show that SCCs represent equivalence classes.
4. Create a directed graph with 3 strongly connected components.
5. (optional) Show that if you consider all vertices of an SCC as a single node, then they define a graph that is acyclic.
6. Show that a Markov Chain is irreducible if the graph induced by the non-zero transition probabilities has a single SCC.

1. Read about Kosaraju's SCC algorithm and write an exposition. (see p.615 in "Introduction to Algorithms" by Cormen et al)
2. (optional) Read about Tarjan's SCC algorithm and write an exposition. Which of the two algorithms do you prefer?
3. (optional) Implement an SCC algorithm. State your implementation complexity.

Project 5 [Existence of stationary distributions in finite irreducible MCs]

In this project you will investigate the proof that the stationary distribution for a finite irreducible Markov chain exists. As a result you will produce a short exposition of the proof.

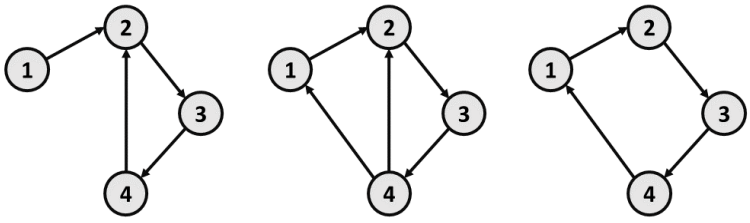
Read pages 11 and 12 from "Markov Chains and Mixing Times" book.

1. Write an exposition for Lemma 1.13. The following questions may guide you:
    1. Define the terms $P_z, E_z, \Omega, \tau_x, \tau_x^+$. (You may need to check a few of the previous pages)
    2. Why is it the case that $E[Y] = \sum_{y=0} \Pr(Y > y)$ for a non-negative integer random variable?
    3. Why does the term r appear in the last equation?
2. Write an exposition for Lemma 1.14. The following questions may guide you:
    1. Explain why $\overline{\pi}(y) \leq E_z[\tau_z^+]$.
    2. Explain the step at Equation (1.21).
    3. Explain the first equality at the top of page 13.
    4. Write in detail the steps for the case distinction of Equation (1.24).
3. (optionally) Read section 1.5.4 and write an exposition for why the stationary distribution is unique. (cf project related to matrix row and column rank).

Project 5 [Periodicity in Markov chains]:

1. Show how to compute the gcd of n integers when given a $\gcd$ function that takes two arguments.
2. Show that the Chicken McNugget theorem (see this video or this) implies that for $a, b \in \mathbb{Z}^+$ with $\gcd(a, b) = g$, there exists a number m, so that for all $m' \in \mathbb{Z}^+$ with $m' \cdot g > m$, $m' \cdot g$ is representable as a linear combination of a and b with non-negative coefficients.
3. Use (1) and (2) to show that for $x_1, \ldots, x_n \in \mathbb{Z}^+$ and $\gcd(x_1, \ldots, x_n)$ there is an integer m such that for all $m' \in \mathbb{Z}^+$ with $m' > m$, $m' \cdot g$ is representable as a linear combination of $x_1, \ldots, x_n$ with non-negative coefficients.

4. A graph consists of vertices and edges (with direction) between the vertices. A graph is *irreducible* if for every two vertices x, y, there exists a path of some length from x to y. For example, in the figure below, the graph on the left is not irreducible (since you cannot get from 1 to 3) and the other two graphs are.



We define for a vertex x, $T(x) = \{t \geq 1 : \text{there is a path of length } t \text{ from } x \text{ to } x\}$. The *period* $p(x)$ of a vertex x is the greatest common divisor of the elements of $T(x)$.

Show that if a graph is irreducible, then for all vertices x, y, $p(x) = p(y)$.

5. A graph is *aperiodic* if all vertices have period 1 (middle one is aperiodic). Show that for a graph that is aperiodic and irreducible, there is an integer r such that for all x and y there is a path from x to y of length r.
6. (optional) How would you check if a graph is aperiodic?

Project 6 [Sampling using Markov Chains]

This project is under construction, but you can still work on it

Chapter 29 from this book describes several uses of Markov Chains in sampling. Choose one or two methods and write an exposition. You may also get a better understanding by reproducing some of the diagrams. If you choose to write about importance sampling, you could read Section 5.3 of the extended notes.

MLEs

Project 7 [Sufficient statistics and some applications]

In this project you will learn about *sufficient statistics* and how they reveal something interesting about MLEs. Read Lecture 2 in these lecture notes and answer the following questions:

1. Define formally what a sufficient statistic is. Give a trivial sufficient staistic that works for any distribution.
2. Define formally what a minimal sufficient statistic is. (see here)
3. Find a minimal sufficient statistic (no need to argue that it is minimal) for:
    1. $X_1, \ldots, X_n$ being n independent observations with $X_i \sim \text{Po}(\lambda)$.
    2. $X_1, \ldots, X_n$ being n independent observations with $X_i \sim \mathcal{N}(\mu, \sigma)$.
    3. For random variables of setting E1Q6.
4. Is the sufficient statistic unique?
5. Prove that the MLE depends on the sufficient statistic.
6. Is this supported by the result you got in E1Q2, E1Q4 and E1Q6?
7. Show the posterior distribution depends only on the sufficient statistic.
8. Is this supported by the result you got in E2Q7(c)?
9. (+) State and prove the factorisation criterion for discrete random variables.

Project 8 [Fisher Information and its relation to MLE]

In this project, you will investigate the Fisher information and prove the Cramer-Rao lower bound. Optionally, you can empirically verify that the MLE is asymptotically normal.

1. Read parts of this article to answer the following questions:
    1. Define the score functions L_1 and L_2.

    2. Show that if you can interchange $d/d\theta$ with E_θ, then for $h(X)$ a statistic with $E_\theta[h(X)^2]$,

$$\frac{d}{d\theta}E_\theta[h(x)] = E_\theta[h(X)L_1(X,\theta)]$$

3. Show that $E_\theta[L_1(X,\theta)] = 0$ for $\theta \in \Theta$.
4. Show that $\mathrm{Cov}_\theta(h(X), L_1(X,\theta)) = \frac{d}{d\theta}E_\theta[h(X)]$.
5. Show that $\mathrm{Var}_\theta(L_1(X,\theta)) = E_\theta[L_1^2(X,\theta)]$.
6. Recall from Part IA Probability, the Cauchy-Schwarz inequality for random variables A and B with finite second-order moments, $\mathrm{Var}(A)\mathrm{Var}(B) \geq (\mathrm{Cov}(A,B))^2$.
7. Prove the Cramer-Rao lower bound,

$$\mathrm{Var}_\theta(h(X)) \geq \frac{\left(\frac{d}{d\theta}E_\theta[h(X)]\right)^2}{E_\theta[L_1^2(X,\theta)]}$$

8. Define the Fisher information $\mathcal{I}(\theta)$.

2. Assume $h(X)$ is an unbiased estimator for parameter θ. Simplify the above expression for the Cramer-Rao lower bound.
3. (optional) Show that when there are multiple i.i.d. samples, $X_1, \ldots, X_n$, $E_\theta[L_1^2(X_1,\ldots,X_n;\theta)] = nE_\theta[L_1^2(X,\theta)]$.
4. Now, we will look into the Cramer-Rao lower is applied to prove that an estimator is optimal in some sense.
    1. Define what *Uniformly Minimum Variance Unbiased Estimator (UMVUE)* is.
    2. Show that the MLE estimator for the Bernoulli distribution with multiple samples is a UMVUE. *Hint:* Show that the CR bound for the Bernoulli distribution is $p(1-p)/n$ and show that the MLE matches this.
    3. Show that the MLE estimator for the Poisson distribution with multiple samples is a UMVUE. *Hint:* Show that the CR bound is λ/n, using the fact that $E[X^2] = \lambda(\lambda+1)$.
5. (optional) There is a theorem which states that under some assumption about the underlying distribution, for large number of samples n, the MLE follows a Normal distribution $\Theta_{\mathrm{MLE}} \sim \mathcal{N}(\theta, (n\mathcal{I}(\theta))^{-1})$ (see p.21 here for a formal statement, but it will not be needed for the exercise). In the rest of the exercise, you will empirically verify this theorem for the Poisson and the Bernoulli distribution.
    1. For the Bernoulli distribution:
        1. Pick a value of $p \in (0,1)$.
        2. Repeat K times the following steps: Take n (some sample size) samples of Bernoulli r.vs. (use the Binomial distribution)
        3. Collect the K samples for the MLE and plot the empirical distribution.
        4. Estimate the variance of the empirical distribution and verify that it is close to the Fisher information.
    2. Repeat for the Poisson distribution.

Project 9 [Family of Exponential distributions]

1. Define what is meant by an exponential family? (see e.g. wikipedia)
2. Several of the distributions you have come across belong to this family:
    1. Show that the Bernoulli distribution forms an exponential family.
    2. Show that the Poisson distribution forms an exponential family.
    3. Show that the Gaussian distribution forms an exponential family.
    4. Show that the Multinomial distribution forms an exponential family.

    5. Show that the Pareto distribution forms an exponential family.
3. Show that $E[t(X)] = \nabla_\eta A(\eta)$.

4. Now you will show that every distribution in the exponential family has a conjugate prior (prior and posterior belong to the same distribution).

   Assume that we draw independent samples x_i with

$$\Pr(x_i \mid \eta) = h_1(x_i)\exp(\eta^T t(x_i) - A_1(\eta))$$
$$\Pr(\eta; \lambda) = h_2(\eta)\exp([\lambda_1; \lambda_2]^T [\eta; -A_1(\eta)] - A_2(\lambda))$$

   Show that the posterior distribution belongs in the same exponential family as the prior.

5. Read about MLEs for exponential families here and here.

Project 10 [Another cardinality estimation algorithm]:

This project is under construction, but you can still work on it

Read about another approach for cardinality estimation (see this overview). One such example is HyperLogLog (wikipedia, paper). Write an exposition about how it works and write an implementation or investigate its theoretical guarantees.

Distributions

Project 11 [Basic properties of Gamma, Beta and Dirichlet]

In this project, you will inverstigate the Gamma and Beta distribution in more depth. You will also see the relation between the two and how Dirichlet distribution generalises the Beta distribution. You may find the links (Gamma, Beta, Dirichlet) useful to answer the following questions.

1. Define the function $\Gamma(x)$.
2. Prove the following properties of the $\Gamma(x)$ function:
    1. Show that $\Gamma(x+1) = x\Gamma(x)$ for $x \in (0, \infty)$.
    2. Show that $\Gamma(x+1) = x!$ for $x \in \mathbb{N}$.
    3. By writing $\Gamma(k) = \int_0^1 x^{k-1}e^{-x}\mathrm{dx} + \int_1^\infty x^{k-1}e^{-x}\mathrm{dx}$, show that the two addends are finite. (This shows that the integral is exists and is finite) *Hint:* See the first link.
    4. Deduce that the $\Gamma(\alpha, \beta)$ distribution is indeed a distribution.
    5. Find the mean of the $\Gamma(a, b)$ distribution.
    6. Find the variance of the $\Gamma(a, b)$ distribution.
3. Define the $B(a, b)$ function.
4. Prove the following properties of the $B(a, b)$ function.
    1. Show that $B(a, 1) = \frac{1}{a}$ for $a \in (0, \infty)$.
    2. Show that $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$. *Hint:* See the second link.
    3. Show that $B(a, b) = \frac{(a-1)!(b-1)!}{(a+b-1)!}$ for $a, b \in \mathbb{N}$ and $a, b \geq 1$.
    4. Find the mean of the $B(a, b)$ distribution.
    5. Find the variance of the $B(a, b)$ distribution.
5. Let X_1, X_2 be independent r.vs. with $X_i \sim \Gamma(a_i, 1)$, then $Y_i = X_i/(X_1 + X_2) \sim B(a_i, 1)$. (See Proposition 25 here).

6. (optional) Define the Dirichlet distribution.
7. (optional) Demonstrate the connection between Dirichlet distribution and Gamma distribution. (see here)

Project 12 [Basics of Mathematical Analysis and the Riemann integral]

In this project you will investigate the basics of mathematical analysis that are essential to the formalisation of probability theory for the continuous case. In particular we go over some definitions for what it means to converge and use it to formally define the Riemann integral. Then we demonstrate why this is not sufficient for representing continuous random variables.

You can see these as questions that guide you to the formal definition of the Riemann integral as defined in the 2014 lecture notes of the maths department. To get a thorough understanding of analysis you would need to read the entire lecture notes, so it is expected that you consult the notes for all questions.

1. Define what it means for a sequence to converge. (see here)
2. Demonstrate that the following sequences converge: (i) $a_n = 1/n$, (ii) $1/2^n$ and $\frac{n+5}{n+8}$.
3. Prove that if $a_n \to a$ and $b_n \to b$, then $a_n + b_n \to a + b$ and $c \cdot a_n \to c \cdot a$ for constant c.
4. Define what the limit of a function is. (see here)
5. Determine the following limits: (i) $\lim_{x \to 3} x + 7$, (ii) $\lim_{x \to a} \frac{x^2 - a^2}{x - a}$ and (iii) $\frac{3x^2}{4 + x^2}$.
6. What does it mean for a function to be continuous? Prove that x^3 is continuous. (see here)
7. What does it mean for a function to be differentiable? (see here)
8. Read about the Intermmediate Value Theorem here (can you think of a binary search proof?). Why does the function need to be continuous?
9. (optional Read about Rolle's Theorem and the Mean Value Theorem (see here).
10. Now we will investigate Riemann integrals. (see here and you may also find this chapter interesting)
    1. Define the lower and upper integral.
    2. What does it mean for a function to be Riemann integrable?
    3. Integrate from first principles $f(x) = x$.
    4. Show that the function f is not Riemann integrable in $[0, 1]$:

$$f(x) = \begin{cases} \frac{1}{\sqrt{x}} & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x = 0 \end{cases}$$

    However, if you take the limit of $a \to 0$ of $g(a) = \int_a^1 f(x) \mathrm{dx}$, then $g(a) = 2$.

    5. You can read more disadvantages of the Riemann integral in Chapter 1B.
11. In case you are interested in learning more about the formal foundations of measure theory you would need to read (NOT during term-time of course): (1) the remaining of mathematical analysis, (2) Metric and Topological Spaces (notes) and (3) measure theory (notes).

Vectors and Matrices

Project 13 [Basics of Linear Algebra]

1. Define what is a *vector space*. (see here)

2. What is a *spanning set*? What is a *basis*? What is the *dimension* of a vector space? (see here)
3. Define a *vector subspace*. (see here)
4. Define a *linear map*. Show that the image and kernel is a vector subspace. (see here)
5. State and prove the rank-nullity theorem. (see here)
6. Show that any *linear map* can be represented as a matrix. (see here)
7. Read these notes and show that row and column rank is equal.

Project 14 [Gaussian Elimination]

This project is under construction, but you can still work on it

This project walks you through Gaussian elimination and some applications in Computer Science.

1. Read about the Gaussian elimination.
2. Show how to use the Gaussian Elimination to solve the following systems of equations. (see here)
3. (optional) Implement Gaussian in Python using NumPy that finds a solution to a system of equations (provided there exists a unique solution). (see here)
4. Show how the Gaussian elimination can be used to find the inverse of a matrix. (see here)
5. In this part, we will show how to compute the determinant using Gaussian eliminiation.
    1. Read about the recursive definition of the determinant.
    2. Show that adding two rows does not change the determinant value.
    3. Show that the determinant of an upper triangular matrix is the product of the diagonal entries.
    4. Show how to compute the determinant of a square matrix using Gaussian elimination.

Linear regression

Project 15 [Linear regression in Tensorflow]

Tensorflow is a very popular framework for automatic differentiation. It is primarily used for implementing, training and evaluating neural networks. Linear regression can be seen as a very simple neural network and it can be trained (i.e. fitted) using gradient descend. In this project, you will learn more about this and implement it from scratch in Tensorflow.

1. Read Chapter 3.1 from the book “Deep into Deep Learning”.
2. Read Chapter 3.2 and implement using Tensorflow a gradient descend-based linear regression algorithm.

Project 16 [More theoretical properties of linear regression]

This project is under construction, but you can still work on it

See Lectures 13-15 from the maths Part IB Statistics lecture notes and write an exposition for something that you find interesting.

Hypothesis testing/Confidence intervals

Project 17 [Chi-squared and t-student distributions]

This project is under construction, but you can still work on it

Read Lectures 8, 10 and 11 from the maths Part IB Statistics lecture notes and write an exposition for how chi-squared and t-student distributions relate to the Normal distribution, and how they are used for confidence intervals and hypothesis testing.

Project 18 [Neyman-Pearson Hypothesis testing framework]

This project is under construction, but you can still work on it

Read Lecture 6 from the maths Part IB Statistics lecture notes and write an exposition for Neyman-Pearson hypothesis testing framework.

More models and datasets

Project 19 [A neural probabilistic model]

1. Read about Recurrent Neural Networks from Section 10.6 in the extended lecture notes. (If you want to look at an example of a RNN gate, read e.g. this).
2. Follow this Jupyter notebook to implement an RNN that generates text.

Project 20 [Creating your own dataset]

In this project, you can create a dataset and perform statistical inference on that.

1. Choose a quantity to measure. Here are some examples:
    1. Measure how much time it takes for you to cover a distance of $10m$. Create a dataset timing yourself. Also, time yourself covering distances of various lengths.
    2. Use this Android app or any similar app (or the Arduino) to collect data with your sensors. For example,
        1. Collect light sensitivity to estimate at what time the sun sets down (and maybe estimate the tilt of the earth).
        2. Collect barometric pressure or temperature to predict the temperature in a few hours.
    3. Collect articles from various sites and perform statistical analysis on occurrences of n-grams.
2. Choose a Data Science task and using the techniques you learnt in the course, attempt to solve it.
3. What problems did you encounter (if any) when creating your dataset?