Project Member Application
oasis
2026-27

Mathematics Club

Centre for Innovation
IIT Madras

Instructions

General Instructions

- Mention the following details at the start of your application.

Name:	Insti Nickname:
Roll No:	CGPA:
Phone (WA):	Email:

- Join the Aspiring Team Members WhatsApp group for further updates: Whatsapp Group
- The recommended font is a standard font size 11-13.
- The applications have to be submitted in PDF format, named as:
  `<First_name>_<Roll_Number>_Mathematics_Club_PM_OASIS.pdf`

  For example, `Mohit_CS24B032_Mathematics_Club_PM_OASIS.pdf`.
- You can upload the finished applications in this Google Form
- You may submit the completed application on or before **4th June, 2026**

Note:

- Do look up the resources mentioned at the end.
- It is fine even if you don't answer all the questions.
- Focus on the compulsory questions before attempting the bonus questions.
- If you have any queries, you can reach out to the Project Leads anytime you want:
  - B. R. Lahari (CS24B070): 9059071798
  - Mohit Rathi (CS24B032): 9770770772
- Feel free to use online resources for reading and reference while attempting the application. However, your submitted responses should reflect your own understanding and should not be AI-generated.

Contents

HR

0. Tell us a bit about yourself. Assume you are a master of flexing and proceed.
1. Why do you want to join this project? What skills or qualities do you possess that make you suitable for the work involved?
2. Mention all PoRs/activities you are planning to take this year. Weekly, how much time do you think you will be able to commit to this project? How much time will you commit to other PoRs and academics?
3. What would keep you motivated throughout the period of two semesters? How would you ensure that you are consistent in contributing to the project throughout the tenure?
4. **(10x Weightage)** What does the project name stand for?

Figure 1: Dwight speaking facts

§1 Statistical Learning: Big Data?

Statistical learning refers to a set of tools for modeling and understanding complex datasets. It helps us make sense of known data and derive insights from it which can be used to learn about nature/reality and consequently help us make predictions about variables of interest.

§1.1 The Statistical Mindset: Winning bets anol

Consider the following case: When 25-batch was going to enroll in the institute, Mohit and Lahari decided to wager a bet on who can best predict the average height of the incoming batch. They realized that, without any data, their estimations are just random guesses. So, they decide to collect data.
Because measuring thousands of incoming students is impossible, they were only able to collect a sample of $n = 100$ students who visited the campus early. Let this observed sample of heights be represented as a vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$. To turn this bet into a mathematical problem, they assume these heights are random variables generated by an underlying true population distribution—specifically, a normal distribution:

$$y_i \sim \mathcal{N}(\mu, \sigma^2)$$

Here, μ (the true average height) and σ^2 (the variance) are hidden parameters of nature. They exist, but they are completely unknown, there is no way me or you can know.
Lahari decides to just calculate the average of these observations and bets on it whereas Mohit uses the Maximum Likelihood Estimation (MLE) method to estimate the average for his bet.
Likelihood of a parameter μ given an i.i.d sample $\mathbf{y}$ is defined as

$$\mathcal{L}(\mu) = P(\mathbf{y}\ ;\ \mu) = \prod_{i=1}^{n} P(y^{(i)}\ ;\ \mu)$$

Intuitively, one can think of this as if μ is our parameter, how likely is it that we record a sample like $\mathbf{y}$. In MLE, we choose μ that maximizes likelihood.

Questions

1. An estimator is a formula or a mathematical rule that helps you estimate something. One can even think of it as an algorithm. Get to know what an estimator is and briefly explain it with a small example. Can there exist multiple estimators for the same variable?
2. Explain what MLE is and apply it in the above case assuming data is gaussian distributed.
3. Explain what Bias, Variance and MSE are. How are these used in deciding which estimator is better?
4. Let us say a sleepy Lahari notes someone's height as 1750cm instead of 175cm. What would go wrong in using MLE with a gaussian assumption and why? Further, use MLE assuming the data is laplace-distributed and show that $\hat{\mu}$ is the median of $\mathbf{y}$ for this assumption. Is this a more robust approach? Why or why not?
5. **The Best of Both Worlds (Huber Loss):** L2 loss is computationally fast but fragile. L1 loss is

robust but non-differentiable at zero. The **Huber Loss** function combines them: it uses squared error for small residuals and absolute error for large residuals. Write down the piece-wise derivative of the Huber Loss function. Explain why this derivative acts as a "clipping" mechanism that mathematically prevents any single outlier from exerting infinite pull.

6. *Bonus:* Explain what Maximum A Posteriori (MAP) estimation is and apply it to the height bet. Assume Mohit has a prior belief that the true average height μ is normally distributed around a historical mean of 170 cm with a variance of σ_0^2:

$$\mu \sim \mathcal{N}(170, \sigma_0^2)$$

Derive the MAP estimator $\hat{\mu}_{\text{MAP}}$. How does this prior belief change Mohit's guess compared to his original MLE guess? What happens to the estimator as the sample size $n \to \infty$?

§1.2 Maximized Likelihood so Everyone Likes Him

Mohit, and you too by now, must've realized that he is going to tie on this bet (since the MLE for a normal distribution's mean is exactly the sample average!). Being greedy and wanting a competitive edge, he decides that predicting a single global average is too naive. What if he could predict the exact height of a *specific* student dynamically?

Having maximized likelihood amongst the 25-Batch, he gathers more data. For each student, instead of just measuring their height, he also records their parents' average height (x_1), their age (x_2), and a genetic marker score (x_3). He hypothesizes that a student's height y is no longer just a static average μ, but a baseline plus a weighted sum of these specific features:

$$y = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + \varepsilon$$

This makes a lot of sense intuitively right? Let us formalize this.
Let us represent our input dataset (predictors) as an $n \times p$ matrix X and the corresponding heights (response) as an $n \times 1$ column vector Y:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

We assume that there exists an underlying relationship between the predictors and the response,

$$Y = f(X) + \varepsilon,$$

where f is an unknown function (which Mohit assumed was linear) and ε represents random noise with mean 0 and variance σ^2.

As f is not perfectly known, the primary objective of statistical learning is to *estimate* the function f (i.e., find the best weights w) as accurately and efficiently as possible. This is what we call "fitting a model". However, in practical large-scale settings, several challenges arise:

- Large number of features (p) and observations (n).
- High computation costs for estimating complex models.
- Datasets may contain outliers or corrupted samples which significantly affect estimation quality.
- Test data can shift significantly compared to train data.

In this project, **OASIS**, we aim to explore and implement several algorithms with a strong emphasis on efficiently tackling the above problems.

§2 Tug of War: Bias vs. Variance

Estimation of function f can be done by two methods , parametric and non-parametric.

- Parametric: This method involves assumption of some initial function and finding the parameters. We can infer that this approach is less flexible i.e., more restrictive. Eg: Linear regression where we assume the function follows a linear trend over all features.
- Non-parametric: This method does not explicitly assume any functional format but tries to estimate f that gets as close to the data points as possible. This method is more flexible as there is no assumption that is made while taking the function. Eg: Spline estimation.

To quantify the extent to which predicted response is close to the original one, we use Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

Our goal is to estimate function $\hat{f}$ with the training data for which MSE is less with one of the above methods. As we have seen, the Non-parametric methods are more flexible which will result in less training MSE. But in reality we mostly use parametric methods for datasets with large features. This is because MSE of training data is an underestimate of MSE of test data.

Questions

7. Can you explain the reason for the above sentence with the terms bias and variance?
8. *Bonus:* Simulate linear regression and spline regression across a dataset and show why parametric (less flexible) methods are better.

§3 When in Doubt, Fit a Line

Let us assume that our model is linear with all features i.e,

$$Y \approx \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

here $\beta_0, \beta_1, \cdots, \beta_p$ are parameters that determine the function f. To estimate these parameters, we try to minimize a metric called Sum of Squares of Residuals (SSR) and this method is called 'least squares approach' .

$$\text{SSR} = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2 = (\mathbf{y} - X\boldsymbol{\beta})^\top (\mathbf{y} - X\boldsymbol{\beta})$$

After making appropriate assumptions called "OLS Assumptions" and finding the minima of SSR for all range of β_i values we get optimal values as

$$\hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \mathbf{y}$$

The above equation is called the Normal Equation. In most of the cases, few features won't have much significance in the response. So the β value for such features should ideally be zero. But due to the irreducible error ε and least square optimization, it would be a non-zero value. So to predict if a feature has significance in the response, we compute t-statistics on the parameter.

Questions

9. What are the OLS assumptions? Show a derivation of the equation

$$\hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \mathbf{y}$$

(Hint: you can use either projection, matrix calculus, or your own method)

10. Uhh why did we choose to minimize SSR i.e. Sum of **Squares** of Residuals? We very well could have selected:
    - Sum of Residuals
    - Sum of Absolute Values of Residuals
    - Sum of 4th powers of Residuals

    (a) Explain briefly why Sum of residuals would fail and (b) justify the minimization of sum of **squares** of residuals. (Hint: Likelihood!)

11. Explain what t-statistic is and take this dataset containing 5 features and tabulate each feature's t-value.

§4 How Close Did We Get, Really?

§4.1 T-Test: The Truth Teller or Just a Show

The t-statistic is useful for assessing the significance of individual features in a dataset. However, when multiple features are insignificant simultaneously, relying solely on individual t-tests can increase the risk of error. In this case, we will use F-statistic to predict if there is more than one feature which is insignificant in the response.

Questions

12. Explain the reason why we can't rely solely on t-statistics to assess the significance of multiple insignificant features, and how does the F-statistic address this limitation?

§4.2 The Art of Counting Errors

After training a model on a dataset, we often assess its accuracy using measures such as Residual Standard Error (RSE) and Correlation $(Y,\hat{Y})$ which is also referred as R^2.

Questions

13. What does each of these terms represent, and how do they estimate the model's error? What do the extreme values of the R^2 term indicate?

§5 Subsetting the Data

While estimating the parameters for linear regression we used least squares approach. Although this method works perfectly for cases where $n >> p$, it will have a high variance in the case of $n \approx p$ or $p > n$. So to overcome this issue, we select a subset from the set of variables and use them to train our data. If we brute force all possible combinations, we will have 2^p models to estimate which is computationally expensive.

Algorithm 5.1 Subset Selection

1. Start with a null model, $M_0 : (\beta_1 = \beta_2 = \cdots = \beta_p = 0)$.
2. For $k = 0, \ldots, p-1$:
   a) Consider all $p-k$ models that augment the predictors in M_k with one additional predictor.
   b) Choose the best among these $p-k$ models, and call it M_{k+1}.
   Here, **best** is defined as having the smallest Residual Sum of Squares (RSS) or the highest R^2.

The above algorithm reduces the number of comparisons from 2^p to just $p+1$. Now we need to choose the best from these $p+1$ models. We cannot use RSS or R^2 as measures because the number of features are different

for these $p+1$ models.

Questions

14. Why would estimating parameters using the least squares approach have high variance in the case of $n \approx p$ or $p > n$?
15. What are the various metrics used to evaluate the performance of these $p+1$ models, and how do they differ from one another?

§6 Thinking Outside the (Least) Square

We have used least squares approach to estimate the parameters however, it has two major disadvantages.

- It does not work when number of features is comparable to that of observations.
- It does not shrink any parameter exactly to zero even if the feature does not have any significance in the response.

There are two other methods that we use to overcome this issue, Ridge regression and the Lasso. In Ridge regression we try to minimize ,

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

and while using Lasso, we try to minimise,

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

where λ is a tunable parameter.

Questions

16. What are the key differences between these two methods, how do they improve upon the least squares approach? A bonus if you can explain pictorially too.
17. What role does the parameter λ play in the optimization process?
18. *Bonus:*
    - Recall the derivation of normal equation you did in question 3. Derive a similar equation, but for Ridge Regression.
    - Recall the MAP you did in section 1. Show the similarity between Ridge and MAP.

§7 Ghost RoBusters

Data gathered cannot be fully trusted and can have outliers or highly incorrect values. Since such values are rare, the data can still be used - we must build robust estimators; estimators smart enough to learn the true trend while actively ignoring the garbage.
Suppose you fit an OLS regression line to n clean data points, resulting in a slope $\hat{\beta}_{OLS}$. Now, an adversary adds a single new corrupted point (x_{new}, y_{new}) to your dataset.

Questions

19. Suppose Lahari fit an OLS regression line to n clean data points, resulting in a slope $\hat{\beta}_{OLS}$. Mohit adds a single new corrupted point (x_{new}, y_{new}) to her dataset.
    - If x_{new} is kept at the sample mean $\bar{x}$, but $y_{new} \to \infty$, what happens to the slope?
    - If $y_{new} = 0$, but $x_{new} \to \infty$, what happens to the slope? Explain the mathematical mechanism behind this collapse.
20. **Bonus:** Many classic robust estimators require the model to be *affine equivariant* (the estimator yields the same fundamental result even if the data is rotated or scaled). However, a mathematical theorem states that the breakdown point of any affine equivariant estimator drops exactly to zero if the number of features p is greater than the number of observations n. Explain geometrically why n points in a p-dimensional space (where $p > n$) are impossible to protect from outliers. (Hint: Think about hyperplanes).

§8 The Handyman and His Tools

Now that you have gained a conceptual understanding of statistics, let's familiarize ourselves with the tools required to execute this efficiently. Every sub-part of this section will introduce you stages in the pipeline of the project.

§8.1 Getting started with R

To begin with, we must install R. Depending on the OS in use you can look this up, it will provide you with the step by step instructions required to install R.

- You can check if R has been successfully installed by running `help.start()` from the terminal. This should lead you to a web-page consisting of detailed documentation of the language, feel free to play around with it!
- Find a simple data set of your choice. Copy the data in a spread sheet in google sheets (remember to give access to the mathematics club account!)
- Load this data in R, write a simple script to perform linear regression on the data. Keep all that you have learnt from the previous section in mind whilst doing so!

Questions

21. Perform a simple linear regression on a data set of your choice (in R). Make sure to give us access to the script and data set.

§8.2 Need For Speed

At its core, we are looking at an optimisation problem. The mathematical aspect of optimisation is looked into in the previous section, here we will look into code optimisation. That is to write code that will run fast and efficiently. Let's switch gears to C++ for a while. Take a look at this function:

```cpp
#include <vector>

std::vector<std::vector<int>> multiply(const std::vector<std::vector<int>>& A,
    const std::vector<std::vector<int>>& B) {
    int r1 = A.size();
    int c1 = A[0].size();
    int c2 = B[0].size();
    std::vector<std::vector<int>> result(r1, std::vector<int>(c2, 0));

    for (int i = 0; i < r1; ++i) {
        for (int j = 0; j < c2; ++j) {
            for (int k = 0; k < c1; ++k) {
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    return result;
}
```

- While the logic for this looks intact, the cost of such computation is quite high.
- A function such as matrix multiplication will be called multiple times, large-scale simulations require millions of matrix multiplications. Even a 1-second delay per multiplication adds up to days or weeks of total runtime.
- This task involves optimising the process as much as possible. Here are a few hints: It is useful to think about two things: namely, how loops are structured and how memory is accessed. (Some techniques to look up: loop unrolling, cache friendly code and loop blocking)
- The idea behind parallel computation: breaking down a large task into smaller sub tasks that can be implemented simultaneously.
- The aim of this task is to conceptually understand the pipeline of code, and introduce you to the possibility of writing code that runs efficiently rather than just doing the bare minimum job. This exercise should

be an exploration of how code really works. It is completely alright if you face difficulty in execution of the same.

Questions

22. Find ways in which the above code can be optimised. In addition to parallel computing write about your understanding for any two techniques (you are always free to explore more than required). How much computation time is saved with each optimisation method?
23. Implement these three techniques (in C++) and analyse their performance (of all three combined as well as each technique individually).

§8.3 Go, get the data first

Being a project based on statistics, there is one more area of interest and that is data loading. Here's how to go about it. You can find a dataset of your choice, do note that it should have a minimum of 10^6 data points. Remember to quote the source of the data set in your application.

Questions

Steps for data loading

24.
- Open the contents of the file using `ifstream`. Print the contents of line number 2025 and line 3106.
- Split each line into the required fields.
- Find an appropriate data structure for storing these values.
- *Bonus:* Identify and briefly explain "bottlenecks" and optimize this process if possible! (Hint: Parallelize the process, optimize each step, minimize bottlenecks, etc..)

§8.4 The Simulation Situation

Now that you have learned how to load the dataset, lets work on a simulation in C++.

Questions

25. Use the dataset provided here to simulate linear regression with the Lasso method in C++.
26. You are encouraged to use the Eigen library for efficient matrix operations and linear algebra support.
27. *Bonus:* Why is Eigen efficient?

§8.5 It's a Wrap!

Time has come to link R and C++, we are using C++ for speed, but building the package in R. The next obvious step is to make the functions written in C++ callable in R. This is achieved through something called **wrappers**. Packages such as `Rcpp` allow you to do the same.

Questions

28. Call the matrix multiplication function that you have optimised in C++ in R.

§9 Bonus Section: Who's da iboss?

This section is designed to help you implement a basic subdata selection method. **I**nformation **B**ased **O**ptimal **S**ubdata **S**election (or **IBOSS** for short) is a very intuitive algorithm that selects subdata based on information, as the name suggests. The algorithm has a proof, but we focus on the implementation here.
Refer to the IBOSS paper here
The algorithm is as follows:

```
                              IBOSS(Z, y, k)
Input:
    Z : N × p covariate matrix
    y : response vector
    k : desired subdata size (k ≤ N)
Output:
    β̂ : OLS estimate using selected subdata

Steps:
1. r ← k / (2p)
2. S ← Φ

3. for j ← 1 to p
4.       Remove rows already in S
5.       Select:
             r rows with smallest z_j values
             r rows with largest z_j values
6.       Add selected row indices to S

7. Construct subdata:
         Z* ← rows of Z indexed by S
         y* ← rows of y indexed by S
8. Form design matrix:
         X* ← [1  Z*]
9. Compute OLS estimate β̂ using the new design matrix and normal equation.
10. return β̂
```

Questions

29. Implement the above algorithm in either R or C++.
30. Understand that:

$$T_{subdata-selection} + T_{fitting\ model\ on\ subdata} < T_{fitting\ model\ on\ full-data}.$$

Hence, show that the above holds in your code and also show if and by how much your subdata-fitted model is worse than a model fitted on the full-data. The choice of showing this "worse-ness" is upto you :). You can use this dataset: Here

Resources

1. Statistical Inference - Casella, Berger
2. Statistical Learning with R - Stanford School of Humanities and Sciences
3. The Elements of Statistical Learning - Trevor Hastie, Robert Tibshirani, Jerome Friedman
4. An Introduction to Statistical Learning - Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
5. Loop Unrolling, Cache friendly code and loop blocking - Description of a few optimisation methods
6. Wrapper for C++ in R - Help with `Rcpp`