LAA Presentation: CUR Decomposition

Akash Das
Anjan Mondal
Aniket Saha
Akshay Dada Thorave

Chennai Mathematical Institute

April 17, 2023

Table of Contents

CHENNAI MATHEMATICAL INSTITUTE

About This Project

Why a Matrix Decomposition Method?

One of the most critical problems in today's data driven world is to handle large amounts of data.

- Store the data.
- Carry out required analysis.

Now, a matrix is the most suitable to store numerical data.
We can easily store the data of m objects or units, where each of which is described by n features, in an $m \times n$ matrix.
Then while analyzing, we may face several issues.

- Missing values.
- Linear dependency among some variables.

CHENNAI MATHEMATICAL INSTITUTE

So we want to somehow ***'shrink'*** the data matrix.

- Note that we already have a few decomposition methods at our disposal, such as LU decomposition.

But we actually want to do a ***'low rank approximation'*** of the original matrix.
That is, we want to represent the data matrix using only some of the most ***influential*** variables.

- Now **'Singular Value Decomposition'** already exists.

So in this project, we will try to know how can this new CUR decomposition method help us.

CHENNAI MATHEMATICAL INSTITUTE

Introduction

- The CUR decomposition was proposed by **Michael W. Mahoney** and **Paul Drineas** in their paper *"CUR Matrix Decompositions for Improved Data Analysis"* published in the Proceedings of the National Academy of Sciences of the United States of America in 2009.
- CUR decomposition is a technique used for low rank approximation of a given matrix using three matrices C, U, R.

We will know what these C, U, R mean and how do we come up with such an idea next.

CHENNAI MATHEMATICAL INSTITUTE

About CUR

What is CUR?

- In CUR decomposition, we decompose a given matrix to get three matrices, one using the columns of the matrix (C), one using the rows of the matrix (R) and one constructed matrix (U).
- When the three new matrices are multiplied in a definite order (C×U×R), it produces a low rank matrix, which is a *'good'* approximation of the original matrix.
- Here we, based on a ***probability distribution***, sample some (say, k) columns (to form C) and rows (to form R) of the original matrix and then we carefully construct U based on some rules.

CHENNAI MATHEMATICAL INSTITUTE

About CUR

From the Authors

Here we will discuss what the authors have to say the CUR decomposition.

- CUR decomposition is a ***low rank matrix decomposition***.
- It is explicitly expressed by a small number of columns and/or rows of the original matrix.
- It is a **randomised algorithm**.
- It does not give the **best** low rank (say, k) approximation. However, in most cases, the deviation from the **best** k-rank approximation will be small.

Now we will discuss why do we need CUR in the next slides.

CHENNAI MATHEMATICAL INSTITUTE

- As we saw earlier, in data analysis, we often find missing values, correlation (linear dependency) among a few variables.
- So a fair amount of the data becomes useless to do any kind of analysis.
- So our key objective is to get a low rank matrix which will, *as much as possible, appropriately express* our original data matrix.
- Now, **Singular Value Decomposition** already gives us best low rank approximation of a given matrix.

But there are a few problems with using SVD in some cases. We will see them now.

Some of the limitations of SVD are:

- It is obtained by taking linear combinations of, at most, all the elements of our original data matrix.
  So it is not very much physically interpretable.
  For example, SVD is not interpretable when we are working with genome data, as we can not physically mix up some genes and come up with some physical explanation of that quantity.
- In worst case, it takes a lot of time to compute SVD.
- SVD also requires a lot of space.
- SVD can not handle missing values too well.
- SVD is affected by outliers in data.

Need of CUR

Our Objective

So we want an algorithm which is:

- Should have provable worst case optimality.
- Should have a natural statistical interpretation associated with it's construction.
- Should perform well in practice.

CHENNAI MATHEMATICAL INSTITUTE

Some Definitions

- Randomised algorithm
- Singular Value Decomposition
- Frobenius norm and Euclidean norm
- Statistical Leverage Scores
- Generalized inverse and Pseudoinverse
- Correlation among variables
- Semantically low rank matrices
- Importance Sampling

CHENNAI MATHEMATICAL INSTITUTE

Setup of CUR

- We have a $m \times n$ matrix A.
- We want to get k-rank approximation of A.
- We need to sample k columns of A based on some probability distribution.
  This will form the matrix C.
- The idea behind assigning probability to each column is based on their influence on the total variation which can be represented by *'Statistical Leverge Scores'*.
- Then we sample k rows from A based on some other probability distribution.
  This will form the matrix R.

CHENNAI MATHEMATICAL INSTITUTE

Setup of CUR

- Finally we construct U such that $\|A - C \times U \times R\|_F$ is optimized.
- The authors have claimed that:

$$\|A - C \times U \times R\|_F \leq (2 + \epsilon)\|A - A_k\|_F$$

for some error parameter ϵ, happens with 98% probability.
Here A_k is the best k-rank approximation of A, obtained from SVD.

CHENNAI MATHEMATICAL INSTITUTE

Key Idea of the Algorithm

If we follow the setup described in the previous two slides, we hope to manage to:

- Retain a lot of interpretability as the matrices are formed using explicitly the columns and rows of A, and no abstract quantities.
- Since it is a randomised algorithm, we hope to compute CUR much faster, on an average, than SVD.

CHENNAI MATHEMATICAL INSTITUTE

CUR Algorithm

Statistical Leverage Scores

Now we will discuss how do we calculate *statistical leverage scores*, how do we select columns and rows of A to form C and R, then how do we construct U.

- The idea behind sampling columns based on the statistical leverage scores comes from regression analysis.
- While selecting columns, we use the right singular vectors to compute the leverage score of a column.
- To start off, observe that one can write the j-th column of A (which has rank r) as:

$$A^j = \sum_{i=1}^{r} (\sigma_i u^i) v_j^i \approx \sum_{i=1}^{k} (\sigma_i u^i) v_j^i$$

Here σ_i is the i-th singular value, u^i is the i-th left singular vector, v_j^i is the j-th coordinate of the i-th right singular vector.

CHENNAI MATHEMATICAL INSTITUTE

- Now we define the normalized statistical leverage scores as follows:

$$\pi_j = \frac{1}{k}\sum_{i=1}^{k}(v_j^i)^2,\ \ j = 1, 2, ..., n$$

- Note that $\pi_j \geq 0,\ \forall\ j$.
- $\sum_{j=1}^{n} \pi_j = 1$

$$\because \sum_{j=1}^{n}\pi_j = \sum_{j=1}^{n}\frac{1}{k}\sum_{i=1}^{k}(v_j^i)^2 = \frac{1}{k}\sum_{j=1}^{n}\sum_{i=1}^{k}(v_j^i)^2 = \frac{1}{k}\sum_{i=1}^{k}(\sum_{j=1}^{n}(v_j^i)^2)$$

$$= \frac{1}{k}\sum_{i=1}^{k} 1 = 1$$

- The π_j's, up to scaling, equal to the diagonal elements of the projection matrix onto the span of the top k right singular vectors of A. As such, they have a natural statistical interpretation as a "leverage score" or "influence score" associated with each of the data points.
- They have been widely-used for outlier identification in diagnostic regression analysis.

Now we will take a brief look at importance sampling and then discuss about the algorithm.
So what is importance sampling?

- Importance sampling is a *Monte Carlo method* for evaluating properties of a particular distribution, while only having samples generated from a different distribution than the distribution of interest.

CHENNAI MATHEMATICAL INSTITUTE

CUR Algorithm

Choosing Columns

Now we will see how to get the sampled columns of A.

- The following algorithm takes as input:
  1. A $m \times n$ matrix A.
  2. A rank parameter k.
  3. An error parameter ϵ

  And outputs sampled columns of A.
- **ColumnSelect Algorithm:**
  - Compute $v^1,\ v^2,\ ,...,v^k$, the **top k right singular vectors of** A.
  - Compute the **normalized statistical leverage scores** π_j.
  - Select j-th column of A with probability

$$p_j = min(1, c.\pi_j)\ \forall j \in \{1, 2, ..., n\}$$

    where $c = O(\frac{1}{\epsilon^2} k.logk)$.
  - **Return the matrix C** consisting of the selected columns of A.

- Now that we have formed the matrix C using the columns of A, we need to form the matrix R now.
- We need to select rows of A to form R.
- We actually apply the same **ColumnSelect** algorithm to find R.
- We use ColumnSelect on A^T to select the rows.
- We select j-th row of A with probability

$$p_j^R = min(1, r.\pi_j^R) \ \forall j \in \{1, 2, ..., m\}$$

where $r = O(\frac{1}{\epsilon^2} k.logk)$.

We have found C and R using the columns and rows of A respectively.
Now we need to construct U.
We construct U as follows:

- Compute pseudoinverse (or Moore-Penrose generalized inverse) of C as C^+ and of R as R^+.
- Define U as:
$$U = C^+ \times A \times R^+$$
- **Note that,** the matrix multiplication is defined as C^+ has dimension $k \times m$, A has dimension $m \times n$ and R^+ has dimension $n \times k$.
- We get U as a $k \times k$ matrix.

So our final CUR algorithm is as follows:

- Run **ColumnSelect** on A with c = O($\frac{1}{\epsilon^2}k.logk$) to choose columns and construct the matrix C.
- Run **ColumnSelect** on A^T with r = O($\frac{1}{\epsilon^2}k.logk$) to choose rows and construct the matrix R.
- Define the matrix U as:

$$U = C^+ \times A \times R^+$$

Where C^+ and R^+ are the pseudoinverses (or Moore-Penrose generalized inverses) of C and R respectively.

Error Analysis

Now we will see how much error our CUR makes in comparison to SVD. In other words, we will see how much distant it is from the best k-rank approximation obtained by SVD.
The error made by CUR is:

$$\|A - CUR\|_F \leq \|A - P_C A\|_F + \|A - P_R A\|_F$$

$$\leq 2 \times max(\|A - P_C A\|_F,\ \|A - P_R A\|_F)$$

$$\leq 2 \times (1 + \frac{\epsilon}{2})\|A - A_k\|_F$$

$$= (2 + \epsilon)\|A - A_k\|_F$$

As claimed before.
Here P_C and P_R are two projection matrices with $P_C = CC^+$ and $P_R = R^+R$.

CHENNAI MATHEMATICAL INSTITUTE

What We Achieved

- So we have found C, U, R as described.
- these matrices are explicitly constructed from the original data matrix A, so they retain the interpretability of the columns of A.
- We randomly select the columns and rows of A, so it is computationally less costly than SVD.
- CUR also, in most cases, doesn't make much more error in comparison with SVD. It depends on the error parameter ϵ.

CHENNAI MATHEMATICAL INSTITUTE

Comparison of SVD and CUR

Now we want to visualize what exactly happens with the errors versus rank (k) while doing SVD and CUR on the same given matrix, for a fixed error parameter ϵ for CUR.

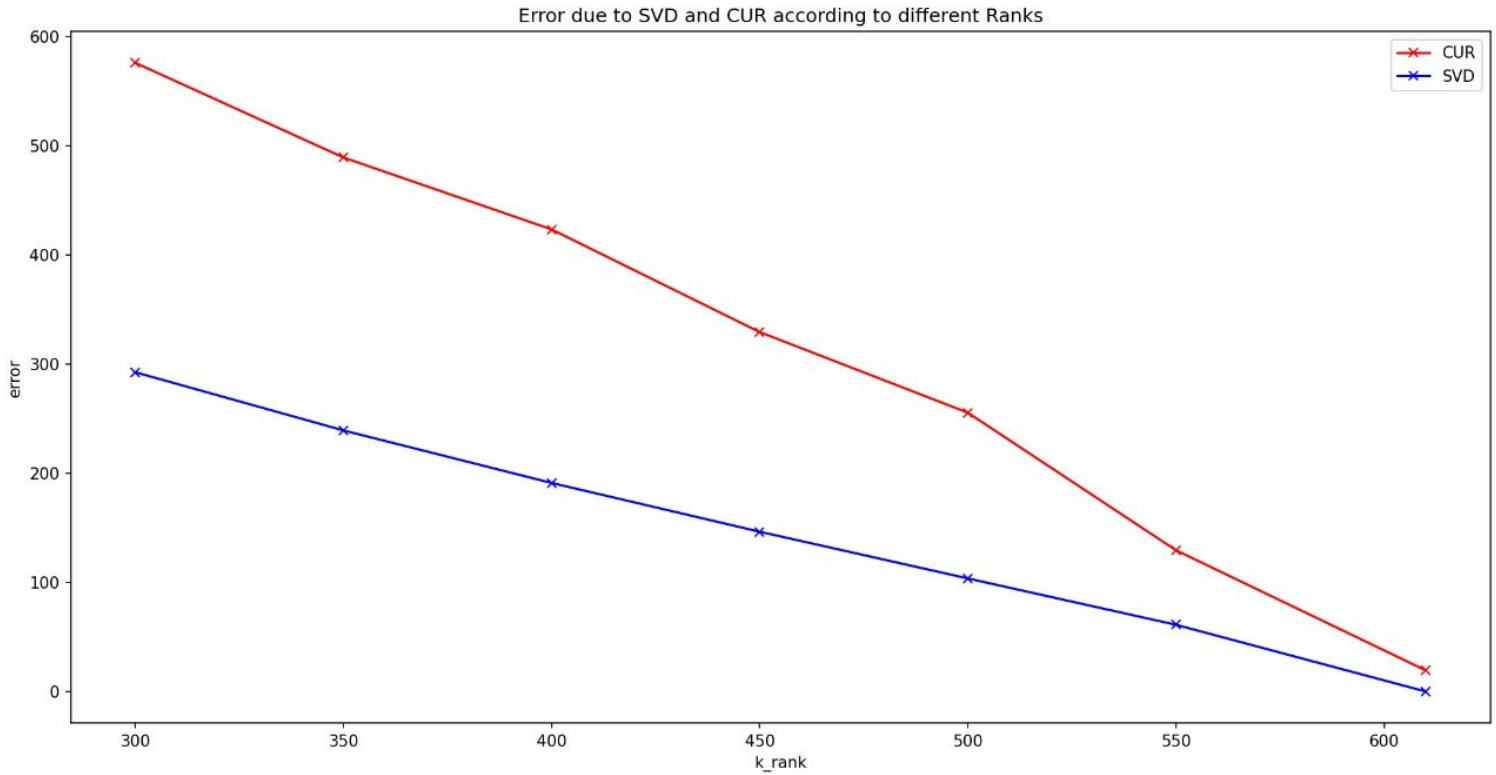

Figure 1: Comparison of Errors of SVD & CUR

CHENNAI MATHEMATICAL INSTITUTE

Comparison of SVD and CUR

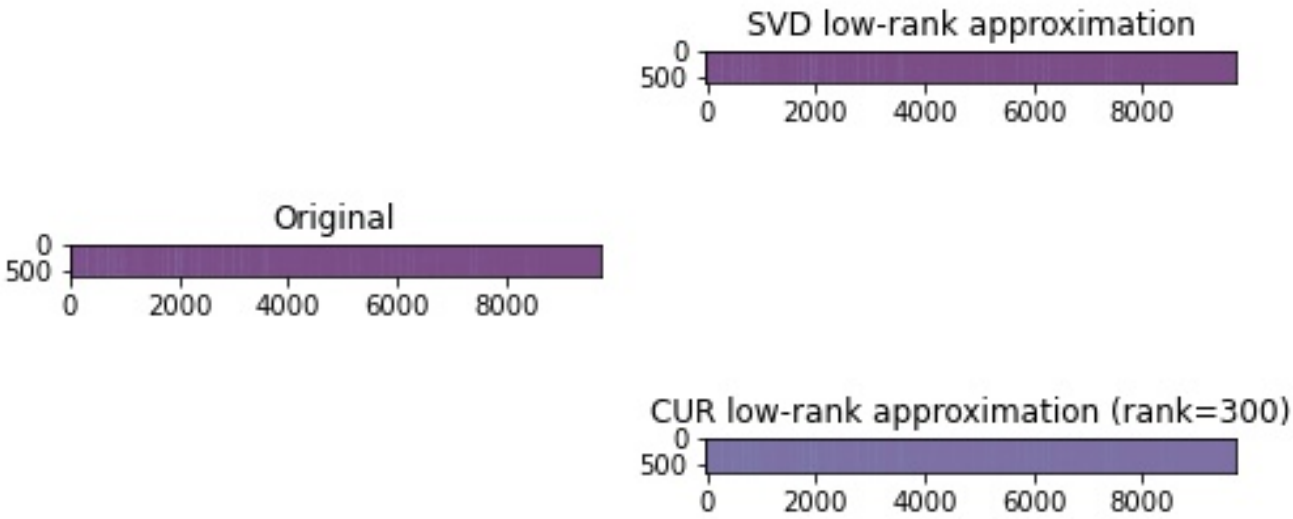

Figure 2: Comparison of Errors of SVD & CUR

CHENNAI MATHEMATICAL INSTITUTE

Comparison of SVD and CUR

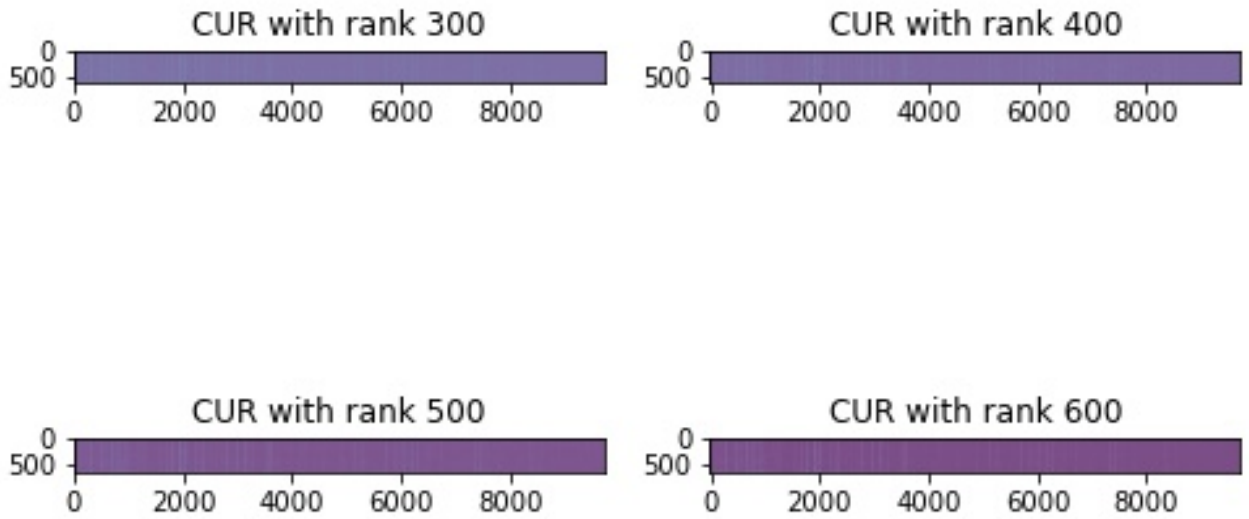

Figure 3: Comparison of Errors of SVD & CUR

CHENNAI MATHEMATICAL INSTITUTE

Uses of CUR

CUR has many uses in data analysis, such as:

- Dimensionality reduction.
- Feature selection, used in machine learning.
- Matrix compression.
- Data clustering, that is cluster columns or rows based on their similarity.

Also it is used over SVD in some cases where we want:

- More interpretability, such as while working with genome data.
- Less computational complexity.
- Better handle sparse or incomplete data.
- Handle semantically low rank matrices.

CHENNAI MATHEMATICAL INSTITUTE

Limitations

Although CUR has many uses in several fields, it has some drawbacks as well.

- Firstly, it is a randomised algorithm, so it has some chance of giving non-satisfactory decomposition.
- Choosing the optimal subset of columns and rows can be difficult sometimes.
- The CUR decomposition may overfit or underfit the data if the subset of columns and rows is not representative of the underlying structure of the data.
- The CUR decomposition can introduce bias or distortion in the decomposition if the columns or rows selected are not independent or orthogonal to one another.

CHENNAI MATHEMATICAL INSTITUTE

Limitations

Strategy for Betterment

We can adopt a few strategies to solve some of the discussed problems as follows:

- Use cross-validation or other techniques to choose the optimal subset of columns and rows.
- Regularize the decomposition to prevent overfitting or underfitting.
- Use techniques such as orthogonal matching pursuit or randomized sampling to reduce bias or distortion in the decomposition.

CHENNAI MATHEMATICAL INSTITUTE

Examples of CUR Usage

Now we will see some cases where CUR is used.

- **Image processing:** CUR decomposition is used to extract features from large image datasets, allowing for more efficient and interpretable analysis of complex visual data.
- **Bioinformatics:** It is also used to analyze gene expression data, identifying key genes and pathways associated with various diseases. **Note that,** we need more interpretability in this field, so CUR is a superior choice than SVD.
- **Recommendation systems:** The CUR decomposition has been used to recommend products or services to users based on their past behavior.
- **Data compression:** It is used to compress large matrices of numerical data, reducing storage requirements and speeding up processing times for machine learning and data analysis tasks.

CHENNAI MATHEMATICAL INSTITUTE

Examples of CUR

We will briefly explain two of the cases here.

- **Bioinformatics:** In this case, CUR has been used to analyze gene expression data.
  - This data consists of large matrices of gene expression levels, where each row corresponds to a different gene and each column corresponds to a different sample or experiment.
  - By performing a CUR decomposition on this data, researchers can identify the most informative genes and samples, allowing them to identify patterns and relationships between genes and diseases.
  - For example, CUR decomposition has been used to identify key genes and pathways associated with cancer, Alzheimer's disease, and other conditions.

CHENNAI MATHEMATICAL INSTITUTE

Examples of CUR

- **Recommendation Systems:** In this case, CUR is used to identify important features and preferences in user data, allowing for more personalized and accurate recommendations.
  - For example, a movie recommendation system might use the CUR decomposition to identify the most important features of movies (such as genre, actors, or director), as well as the most important preferences of users (such as favorite genres or actors).
  - By analyzing these factors using a CUR decomposition, the system can generate recommendations that are tailored to each user's individual preferences.

CHENNAI MATHEMATICAL INSTITUTE

Conclusion

We have seen the CUR decomposition algorithm and discussed some properties of it.

- CUR decomposition is a randomised algorithm which is used in low-rank matrix approximation.
- It has more interpretability and lesser worst case time complexity than SVD.
- Hence it can be used in many fields where SVD doesn't provide much actual insight about the data.

To end, we quote the authors of our selected paper:
"Although the SVD per se cannot be blamed for its misapplication, the desire for interpretability in data analysis is sufficiently strong so as to argue for interpretable low-rank matrix decompositions." - M. W. Mahoney & P. Drineas

CHENNAI MATHEMATICAL INSTITUTE

References

Here are the list of references that we used in making this project:

- Original paper by Mahoney and Drineas. Link.
- Wikipedia. Link.
- YouTube: Stanford University 'Data Mining' course. Link.
- 'Numerical Linear Alebra' by Golub & Loan
- MathStackExchange. Link.
- ChatGPT

CHENNAI MATHEMATICAL INSTITUTE

A randomized algorithm is an algorithm that uses a random number at least once during its operation to make decisions. These algorithms have the advantage of often being simpler and faster than deterministic algorithms, as well as providing probabilistic guarantees on their performance. Randomized algorithms are commonly used in areas such as machine learning, optimization, cryptography, and computer graphics. One example of a randomized algorithm is the Monte Carlo method, which uses random sampling to estimate the value of complex integrals and solve other mathematical problems. Another example is the Karger-Stein algorithm for computing the minimum cut of a graph, which randomly selects edges to contract until only two nodes remain, providing an approximation of the true minimum cut with high probability.

Notes

Generalized Inverse

A generalized inverse is a mathematical tool that provides a solution to a matrix equation even when the matrix in question is not invertible. For a given matrix A, its generalized inverse is denoted as A^+ (also called the pseudoinverse of A) and has the property that $A \times A^+ \times A = A$ and $A^+ \times A \times A^+ = A^+$. Unlike the standard inverse, which only exists for square matrices that are full rank (i.e., have linearly independent columns), the generalized inverse can be computed for any rectangular matrix, regardless of its rank. The generalized inverse has many applications in linear algebra, optimization, statistics, and machine learning. For example, it is used in ridge regression and principal component analysis to compute solutions even when the data matrix is not full rank, and in control theory to solve systems of linear equations that may be under- or over-determined.

CHENNAI MATHEMATICAL INSTITUTE

Notes

Semantically Low Rank Matrices

A semantically low-rank matrix is a type of matrix that may not have a low numerical rank, but has a low rank with respect to the underlying semantics or structure of the data it represents. In other words, the matrix may have many components, but only a small subset of them are meaningful or informative for the problem at hand. For example, in natural language processing, a large corpus of text data may have many dimensions (words) but only a small subset of them are semantically important for understanding the meaning of the text. Similarly, in image processing, a high-resolution image may have many pixels, but only a small subset of them are semantically important for representing the object or scene in the image. By exploiting the semantically low-rank structure of such matrices, it is possible to reduce their dimensionality and complexity without losing important information. This can lead to more efficient and effective algorithms for tasks such as clustering, classification, and data compression.

CHENNAI MATHEMATICAL INSTITUTE

Importance sampling is a statistical technique used to estimate properties of a target distribution by generating samples from an alternate distribution that has non-zero probability where the target distribution is non-zero. In other words, it allows us to compute an integral with respect to a target distribution by drawing samples from a different distribution that is easier to sample from but is related to the target distribution. This can be particularly useful in situations where the target distribution is complex or difficult to sample from directly. Importance sampling can be used to estimate expected values, moments, and other properties of the target distribution. The accuracy of the estimates depends on the choice of the importance distribution, with better results achieved when the importance distribution closely matches the target distribution. Applications of importance sampling can be found in fields such as physics, finance, and machine learning.

Acknowledgement

We are thankful to our professor **Priyavrat Deshpande** for providing such a great topic to make our presentation on.
We would like to thank our **classmates** for their respective presentations, which helped us better understand how to do ours.

Figure 4: Source

CHENNAI MATHEMATICAL INSTITUTE