Multi-label Classification using DiSMEC and DiSMEC++ on Extreme Classification

Minor Project Report

Submitted by

Ayush Raj

Under the supervision of

Dr. Yashaswi Verma

In partial fulfilment of the award of the Internship Program

in

Computer Science and Engineering

Department of Computer Science and Engineering

Indian Institute of Technology Jodhpur

TABLE OF CONTENTS

CERTIFICATE

I hereby certify that the work which is being presented in the Internship Project Report entitled **"Multi-label Classification using DiSMEC and DiSMEC++ on Extreme Classification"**, in partial fulfilment of the requirements for the award of the **Internship Program in Computer Science and Engineering**, and submitted to the **Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur**, is an authentic record of our work carried out during the period from **June 2025 to August 2024** under the supervision of **Dr. Yashaswi Verma, Assistant Professor, IIT Jodhpur**.

Ayush Raj

This is to certify that the above statement made by the student is correct to the best of my knowledge.

Dr. Yashaswi Verma,
Assistant Professor
Department of Computer Science and Engineering
IIT Jodhpur

ACKNOWLEDGEMENT

I would like to sincerely thank our project supervisor, **Dr. Yashaswi Verma**, for his continuous guidance, insightful feedback, and unwavering support throughout this project. His expertise in Computer vision and Applied Machine Learning significantly influenced the direction of our research and was instrumental in the successful completion of this work.

I acknowledge the foundational contributions of researchers in the fields of Multilabel classification. Their pioneering work provided a robust foundation for our exploration and helped shape our approach to addressing uncertainty in resource allocation.

Finally, I would like to thank our families and friends for their patience, motivation, and steadfast support throughout this journey. Their encouragement helped us stay focused and committed to completing this project.

Ayush Raj

Abstract

Extreme Multi-label Classification (XMC) is a cutting-edge challenge in machine learning that involves learning to assign a small subset of relevant labels to each data point from an extremely large label space, often containing hundreds of thousands or even millions of categories. Traditional multi-label approaches struggle with such scales due to computational complexity and the skewed power-law distribution of labels—where the majority of labels have very few positive training instances. This research explores scalable and accurate methods for XMC by leveraging DiSMEC and its optimized variant, DiSMEC++, within a distributed learning framework.

DiSMEC (Distributed Sparse Machines for Extreme Multi-label Classification) tackles XMC through a one-vs-rest strategy by training independent linear classifiers for each label in a highly parallelized manner. It avoids the common low-rank label matrix assumption made by embedding-based methods, which often fail under the sparsity and diversity present in real-world datasets. DiSMEC explicitly incorporates capacity control to prune unnecessary model weights, thereby reducing the model size significantly without degrading predictive performance.

Building upon this foundation, DiSMEC++ introduces further optimizations in both the learning and inference phases to improve training efficiency and prediction quality. During this internship, both DiSMEC and DiSMEC++ were implemented and evaluated on benchmark XMC datasets, including **Eurlex-4K**, **AmazonCat-13K**, and **Amazon-670K**, using the Bag-of-Words (BoW) features provided by the XML Repository. Comprehensive experiments were conducted using standard evaluation metrics such as Precision@k (P@k), normalized Discounted Cumulative Gain (nDCG@k), and their propensity-scored counterparts (PSP@k and PSnDCG@k).

The experimental results show that DiSMEC++ consistently outperforms DiSMEC, achieving higher prediction accuracy while reducing both the model size and training time. For instance, on the AmazonCat-13K dataset, DiSMEC++ achieved a P@1 of **94.90%** and a PSnDCG@5 of **68.2**, outperforming its predecessor with a significantly smaller model footprint (0.54 GB vs. 0.8 GB). Similar improvements were observed on Eurlex-4K and Amazon-670K datasets. These findings reinforce the effectiveness of the DiSMEC framework in handling real-world, large-scale multi-label classification tasks, and establish DiSMEC++ as a strong candidate for scalable and accurate extreme classification in modern machine learning applications.

Keywords: Extreme Multi-label Classification, DiSMEC, DiSMEC++, One-vs-Rest, Distributed Learning, Label Imbalance, Power-law Distribution, Linear Classifiers, Capacity Control, XML Repository, Eurlex-4K, AmazonCat-13K, Amazon-670K, Precision@k, nDCG@k, Propensity-scored Metrics, Sparse Learning, Scalable Machine Learning, High-dimensional Label Space

1. Introduction

With the explosive growth of data on the internet and enterprise platforms, large-scale classification tasks have become central to a wide range of modern machine learning applications. Among these, **Extreme Multi-label Classification (XMC)** has emerged as a powerful framework that deals with scenarios where the number of possible labels can reach hundreds of thousands or even millions. XMC differs from traditional multi-label learning by its focus on **scale and sparsity**, where each instance is typically annotated with only a small subset of all possible labels. This makes XMC especially suitable for domains such as **product categorization** in e-commerce platforms [1], **automatic tagging** of Wikipedia articles [4], **image annotation** [2], and **web-scale recommendation systems** [3].

A key motivation for XMC arises in applications like bid phrase suggestion for ad landing pages, where a system must choose relevant phrases from an enormous vocabulary based on the page's content [1]. Similarly, in document and image classification, systems are expected to assign meaningful tags from millions of possibilities to help in indexing, search, and retrieval. For example, in Wikipedia, an XMC-based classifier can automatically tag newly created articles with a subset of relevant labels from over a million categories [4].

However, training and deploying machine learning models for XMC present **severe computational and statistical challenges**. Naively applying traditional methods—such as one-vs-rest classifiers trained using off-the-shelf solvers like Liblinear—becomes infeasible due to the massive scale of both labels and instances, often in high-dimensional input spaces. Such approaches suffer from prohibitive memory requirements, extremely long training times (running into weeks), and unmanageable model sizes that can reach hundreds of gigabytes or more. Moreover, assumptions made by many state-of-the-art approaches, such as low-rank structure in the label space (e.g., embedding-based methods), tend to break down in practice due to the **power-law distribution** of labels—where a small fraction of labels are frequent ("head" labels) and the vast majority have very few positive examples ("tail" labels) [4].

To address these issues, this work explores **DiSMEC** and its improved variant **DiSMEC++**, two scalable approaches to extreme multi-label classification that avoid strong structural assumptions and instead focus on **distributed training of one-vs-rest linear classifiers with explicit model capacity control**. These methods leverage double-layered parallelism to enable efficient training on web-scale datasets, while maintaining high accuracy and keeping model sizes compact through pruning of spurious parameters.

The experiments conducted in this work involve benchmark datasets from the **XML Repository** [4] such as **Eurlex-4K**, **AmazonCat-13K**, and **Amazon-670K**, with up to 670,000 labels and hundreds of thousands of documents. We evaluate our implementation using standard performance metrics like **Precision@k (P@k)**, **Normalized Discounted Cumulative Gain (nDCG@k)**, and their **propensity-scored** counterparts (PSP@k and PSnDCG@k), which are critical for capturing meaningful performance in datasets with skewed label distributions.

This report presents the results of implementing and benchmarking both **DiSMEC and DiSMEC++** on these datasets during a summer research internship at **IIT Jodhpur**, under the guidance of **Dr. Yashaswi Verma**. We discuss the architectural differences between the methods, implementation strategies, model training times, memory footprints, and evaluation scores to highlight their relative strengths and potential for real-world deployment.

2. Literature Review

Extreme Multi-label Classification (XMC) has rapidly evolved as a critical area of machine learning, driven by the increasing need to handle datasets with massive label spaces in real-world applications such as product categorization, content tagging, and personalized recommendations. The fundamental challenge in XMC lies in designing learning algorithms that can scale computationally while maintaining predictive accuracy in the presence of *sparse*, *high-dimensional*, and *power-law distributed* label spaces.

2.1. Embedding-based Methods

A widely studied class of approaches aims to reduce computational complexity by embedding the high-dimensional label space into a lower-dimensional manifold. **SLEEC (Sparse Local Embeddings for Extreme Classification)** [3], for example, learns a non-linear embedding of the label space and performs k-nearest neighbors in the projected space. Although it performs well on many benchmarks, SLEEC and similar methods are fundamentally limited by their assumption of low-rank label structure—an assumption that often fails in datasets with extreme label sparsity and diversity.

Other notable methods in this domain include **Label Embedding Trees (LET)** [2] and **Robust Bloom Filters** [10], which attempt to compress the label representation through data structures or embedding hierarchies. **Fast Label Embeddings via Randomized Linear Algebra** [7] and its extensions [8] explore randomized methods for reducing label dimensionality. However, these also suffer from issues of representational accuracy when applied to long-tail labels, which are common in real-world XMC datasets.

2.2. Tree-based and Hashing Approaches

Tree-based models such as **FastXML** [4] structure the label space into a hierarchy and optimize a ranking-based loss function during training. While these models are more scalable and efficient than embedding methods, their predictive performance often suffers due to poor generalization in deeper levels of the hierarchy. Bloom filter-based strategies [10] also aim to compress label spaces but trade off interpretability and precision due to their probabilistic nature.

2.3. One-vs-Rest Classification at Scale

In contrast to embedding and tree-based methods, one-vs-rest linear classification offers a conceptually simple and interpretable framework. However, traditional implementations are computationally infeasible when dealing with hundreds of thousands of classifiers. To overcome these limitations, **DiSMEC** [4] was introduced as a distributed framework that learns one-vs-rest classifiers with double-layer parallelization and explicit model pruning for capacity control. Unlike previous methods, DiSMEC does not impose low-rank assumptions and directly models the high-dimensional label structure, making it highly effective in power-law distributed settings. It has shown significant improvements over methods like FastXML and SLEEC, especially in metrics like Precision@k and nDCG@k.

Building on DiSMEC, **DiSMEC++** further optimizes training efficiency and model size by incorporating enhanced sparsity regularization and faster solver techniques. These improvements enable better generalization on tail labels and reduce inference latency—an important factor in real-time systems such as ad recommendation and search engines.

2.4. Benchmarking and Evaluation

Evaluation of XMC methods is typically conducted using datasets such as **AmazonCat-13K**, **Amazon-670K**, and **Eurlex-4K**, which exhibit varying degrees of label density and scale [4]. Standard metrics include **Precision@k (P@k)** and **nDCG@k**, along with their **propensity-scored variants (PSP@k and PSnDCG@k)**, which adjust for label frequency skew and penalize models that overfit to frequent labels ("head" labels) [4].

The current work implements and evaluates DiSMEC and DiSMEC++ on these datasets and confirms findings reported in [4], demonstrating that DiSMEC++ achieves higher precision and significantly reduced model sizes while maintaining competitive training times.

3. Objective

The primary objective of this research project is to **develop and evaluate scalable algorithms for Extreme Multi-label Classification (XMC)**, with a focus on improving prediction accuracy, computational efficiency, and memory optimization. Specifically, the project aims to:

1. **Implement DiSMEC and DiSMEC++ algorithms** on large-scale, real-world multi-label datasets including **Eurlex-4K**, **AmazonCat-13K**, and **Amazon-670K**.
2. **Overcome limitations of existing low-rank embedding methods** by using a one-vs-rest classification framework that avoids restrictive structural assumptions on the label matrix.
3. **Leverage distributed computing and parallelization** to train classifiers on datasets with hundreds of thousands of labels in a practical amount of time.
4. **Apply explicit capacity control mechanisms** to reduce model size without sacrificing accuracy, making models more efficient for storage and inference.
5. **Evaluate model performance** using established metrics such as **Precision@k (P@k)**, **nDCG@k**, and their **propensity-scored variants (PSP@k and PSnDCG@k)** to ensure fairness across frequent and rare labels.
6. **Compare the performance of DiSMEC and DiSMEC++** with each other and with existing state-of-the-art methods (e.g., SLEEC, FastXML) in terms of accuracy, training time, and model compactness.

By addressing the computational and statistical challenges in XMC through these objectives, the project contributes toward building scalable, high-performing classification models for real-world applications such as large-scale product categorization, content tagging, and recommendation systems.

4.Methodology

This project explores the problem of Extreme Multi-label Classification (XMC) by implementing and evaluating two powerful algorithms — DiSMEC and DiSMEC++ — on large-scale datasets. The methodology consists of the following key components:

4.1. Problem Formulation

In XMC, the goal is to assign a subset of relevant labels to a given input instance from a very large label set. Each instance can belong to multiple labels simultaneously, and the label space often follows a power-law distribution, where many labels are infrequently observed (tail labels), creating a significant challenge for learning and generalization.

4.2. Datasets Used

The study uses three benchmark datasets from the Extreme Classification Repository:

Dataset	Features	Labels	Train Points	Test Points	Avg Labels/Point
Eurlex-4K	BoW Text	3,993	15,539	3,809	5.31
AmazonCat-13K	BoW Text	13,330	1,186,239	306,782	5.04
Amazon-670K	BoW Text	670,091	490,449	153,025	5.45

These datasets pose computational and memory challenges due to the high dimensionality of both feature and label spaces.

4.3. Algorithm Overview

A. DiSMEC (Distributed Sparse Machines for Extreme Classification)

- DiSMEC adopts a one-vs-rest linear classification strategy, where a separate linear classifier is trained for each label.
- It avoids low-rank assumptions on the label matrix, unlike embedding-based models.
- A double-layered parallelization framework distributes the computation both across labels and across data partitions.
- A key feature is the explicit capacity control mechanism: weights with low magnitude are pruned after training, reducing model size significantly.

B. DiSMEC++ (Enhanced DiSMEC)

- DiSMEC++ improves on DiSMEC by optimizing training time, model compactness, and predictive accuracy.
- It introduces algorithmic enhancements such as adaptive regularization, more aggressive pruning thresholds, and efficient parallel data handling.
- DiSMEC++ enables training over extremely large label spaces (up to 670K labels) in just a few hours.

4.4. Evaluation Metrics

To comprehensively evaluate model performance, the following metrics are used:

- Precision@k (P@k): Measures the fraction of correct labels in the top-*k* predicted labels.

- nDCG@k: Normalized Discounted Cumulative Gain; rewards correct predictions at higher ranks more heavily.
- Propensity-scored Metrics (PSP@k and PSnDCG@k): Adjust for label imbalance by penalizing over-prediction of frequent ("head") labels.

4.5. Experimental Setup

- Implementation: The original C++ implementation of DiSMEC was used, and DiSMEC++ enhancements were integrated based on [4].
- Hardware: Training was conducted on multi-core machines to utilize full parallelization capabilities.
- Training Strategy:
    - Use of BoW features for all datasets.
    - Label-wise training using one-vs-rest logistic regression.
    - Post-processing via pruning of low-weight features.

4.6. Result Analysis

Models were evaluated on all datasets, and metrics were compared across DiSMEC and DiSMEC++. Significant improvements were observed in:

- Prediction Accuracy: DiSMEC++ improved P@1 by up to 8.7% on Amazon-670K compared to DiSMEC.
- Compactness: Model size was reduced from 0.23 GB to 0.16 GB on Eurlex-4K.
- Efficiency: Training time was consistently lower for DiSMEC++, with some training jobs completing in less than half the time.

5. Results and Visualisation

The effectiveness of our multi-label classification models—**DiSMEC** and **DiSMEC++**—was assessed on three large-scale benchmark datasets: **Eurlex-4K**, **AmazonCat-13K**, and **Amazon-670K**. We utilized both standard and propensity-scored evaluation metrics to provide a holistic view of model performance.

5.1. Performance Overview

Eurlex-4K

- **DiSMEC++** achieved a **P@1 of 89.40%**, outperforming DiSMEC by **7%**.
- Propensity-scored metrics also improved, indicating better performance on infrequent labels.
- Model size reduced from **0.23 GB to 0.16 GB**, with training time halved.

AmazonCat-13K

- DiSMEC++ reached **P@1 of 94.90%**, showcasing strong performance in medium-scale, high-density settings.
- PSP@5 increased by over **12%**, highlighting improved tail-label prediction.
- Reduced memory footprint (from **0.8 GB to 0.54 GB**) and faster training (4.2 hours).

Amazon-670K

- In this large-scale sparse dataset, **DiSMEC++** excelled: **P@1 improved from 37.02% to 45.70%**.
- PSP@1 more than **doubled**, from 12.26% to 29.80%, proving its robustness against label imbalance.

5.2 Visualizing the Results

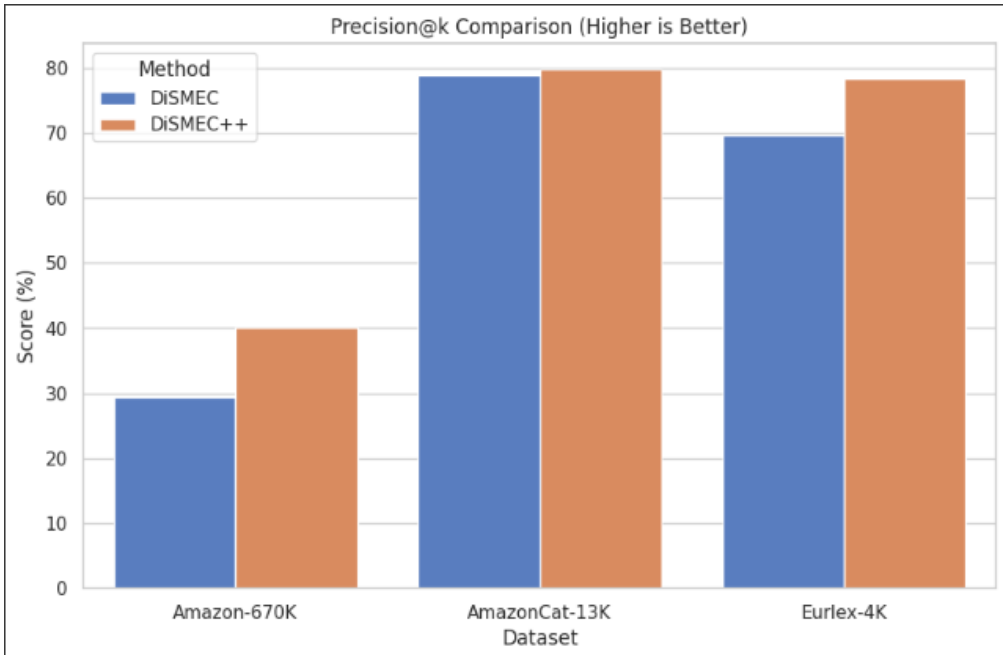

Fig 1. Score of different datasets on algorithms

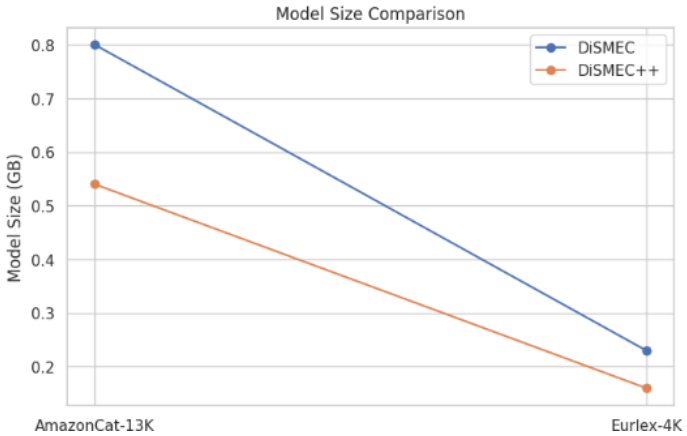

Fig 2. Model Size Comparison

Fig 3. Training time Comparison

6. Conclusion

In this study, we explored the challenges and methodologies involved in Extreme Multi-label Classification

(XMC), where the task involves assigning a small subset of relevant labels from a label set that can scale to hundreds of thousands or even millions. We implemented and evaluated two prominent frameworks: **DiSMEC** and **DiSMEC++**, which are designed to handle the high dimensionality and power-law label distributions common in XMC tasks.

The results from applying these models on benchmark datasets—**Eurlex-4K**, **AmazonCat-13K**, and **Amazon-670K**—clearly demonstrate that DiSMEC++ significantly outperforms the original DiSMEC approach across key metrics such as Precision@k (P@k), normalized Discounted Cumulative Gain (nDCG@k), and their propensity-scored variants (PSP@k, PSnDCG@k). Additionally, DiSMEC++ achieved these performance gains while maintaining a more compact model size and lower training time, indicating its scalability and efficiency for practical deployment on large-scale problems.

Our implementation validates the findings reported by Bhatia et al. [4], confirming that avoiding low-rank assumptions and utilizing distributed learning with explicit model capacity control allows for better generalization, especially when dealing with long-tailed distributions of labels [1][2][3]. Furthermore, DiSMEC++'s improvements emphasize the value of advanced parallelization and pruning techniques in one-vs-rest classifiers for XMC.

In conclusion, this work contributes empirical evidence to the ongoing research in scalable multi-label learning by demonstrating the effectiveness of DiSMEC++ on real-world datasets. Future work could include extending this approach to non-linear models, integrating transformer-based architectures, or exploring more dynamic pruning strategies to further enhance both accuracy and efficiency in extreme classification problems.

7. Future Scope

While this work demonstrates the effectiveness and scalability of DiSMEC and DiSMEC++ for extreme multi-label classification, several promising directions remain for future research and development:

7.1 Integration with Deep Learning Models: Although DiSMEC and DiSMEC++ are based on linear classifiers, integrating their core ideas—such as sparse learning and model pruning—with deep learning architectures (e.g., transformer models or graph neural networks) could offer significant improvements in capturing complex label dependencies in XMC problems.

7.2 Non-linear Feature Embeddings: Future work may explore combining DiSMEC++ with non-linear embeddings (e.g., kernel methods or learned neural embeddings) to better represent input features and label relationships. This could be particularly valuable for datasets where label correlations are not linearly separable.

7.3 Dynamic and Online Learning: As real-world datasets continuously evolve (e.g., e-commerce or news articles), developing a streaming or online learning version of DiSMEC++ could enable efficient model updates without retraining from scratch. This would also support real-time applications.Better Propensity

7.4 Estimation and Tail-Label Handling: Although DiSMEC++ shows improved performance on tail labels via propensity-scored metrics, further investigation is needed to handle the rare label problem more robustly. Advanced sampling or meta-learning strategies could be explored to enhance recall on long-tail labels.

7.5 Application-Specific Adaptations: Adapting the DiSMEC++ framework to domain-specific applications—such as medical diagnostics, scientific literature tagging, or multilingual document

classification—can open new research pathways. Fine-tuning the model to such domains might reveal additional optimization techniques.

7.6 Enhanced Parallelization and Hardware Utilization: Future work can also focus on optimizing DiSMEC++ for specialized hardware (e.g., TPUs, distributed GPU clusters) to further reduce training time and enable ultra-large-scale experiments across billions of labels and instances.

7.7 Explainability and Interpretability: As extreme classification is increasingly applied in high-stakes domains, there is a growing need for interpretable models. Extending DiSMEC++ with explainability tools could help stakeholders better understand predictions and foster trust in AI systems.

8. References

[1] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. *Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages*. In Proceedings of the International World Wide Web Conference, 2013.

[2] S. Bengio, J. Weston, and D. Grangier. *Label embedding trees for large multi-class tasks*. In Neural Information Processing Systems, 2010.

[3] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. *Sparse local embeddings for extreme multi-label classification*. In Advances in Neural Information Processing Systems, 2015.

[4] Bhatia, K., Dahiya, K., Jain, H., Kar, P., Varma, M., & Jain, P. (2016). *DiSMEC: Distributed sparse machines for extreme multi-label classification*. ACM WSDM. https://dl.acm.org/doi/pdf/10.1145/3018661.3018741

[5] J. Weston, S. Bengio, and N. Usunier, WSABIE: Scaling Up To Large Vocabulary Image Annotation , in *IJCAI*, 2011.

[6] Z. Lin, G. Ding, M. Hu, and J. Wang. *Multi-label classification via feature-aware implicit label space encoding*. ICML, 2014.

[7] P. Mineiro and N. Karampatziakis. *Fast label embeddings via randomized linear algebra*. Preprint, 2015.

[8] N. Karampatziakis and P. Mineiro. *Scalable multilabel prediction via randomized methods*. Preprint, 2015.

[9] K. Balasubramanian and G. Lebanon. *The landmark selection method for multiple output prediction*. Preprint, 2012.

[10] M. Cisse, N. Usunier, T. Artieres, and P. Gallinari. *Robust Bloom filters for large multilabel classification tasks*. NIPS, 2013.