

EXTREME CLASSIFICATION: LOSS FUNCTIONS, ALGORITHMS AND APPLICATIONS

HIMANSHU JAIN



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
MARCH 2020

©Indian Institute of Technology Delhi (IITD), New Delhi, 2020

EXTREME CLASSIFICATION: LOSS FUNCTIONS, ALGORITHMS AND APPLICATIONS

by

HIMANSHU JAIN

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the



Indian Institute of Technology Delhi

MARCH 2020

Certificate

This is to certify that the thesis titled **EXTREME CLASSIFICATION: LOSS FUNCTIONS, ALGORITHMS AND APPLICATIONS** being submitted by **MR. HIMANSHU JAIN** for the award of **Doctor of Philosophy in Computer Science and Engineering** is a record of bona fide work carried out by him under our guidance and supervision at the Department of Computer Science and Engineering, Indian Institute of Technology Delhi. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma unless otherwise stated explicitly. In particular, the work done in Chapter 2 was done jointly with a PhD student. Work done in Chapters 3, 4 and 5 was done jointly with research fellows at Microsoft Research. In each case, the part done by the collaborators appeared in their respective theses.

Rahul Garg

Professor

Department of Computer Science & Engg.

Indian Institute of Technology Delhi

New Delhi- 110016

Manik Varma

Adjunct Professor

Department of Computer Science & Engg.

Indian Institute of Technology Delhi

New Delhi- 110016

Acknowledgements

I am deeply thankful to my advisor Dr Manik Varma for his guidance and support throughout this thesis. His passion towards the subject, drive to do something impactful and ability to pay attention to the minutest of details always kept me motivated, on toes and pushed me to do my best. I have also vastly benefited from his unique mix of industry and academia experience. I would certainly not have hoped for a better learning experience than this. I would also like to acknowledge the help and support that I received from Prof. Rahul Garg.

I am also thankful to Prateek Jain and Purushottam Kar for their invaluable inputs throughout my research. I am very grateful to Nicolas Mayoraz for hosting me for the summer internship at Google. It was an amazing learning experience and ofcourse I can never forget the half-dome hike. I am also thankful to my collaborators Kush Bhatia, Venkatesh Balasubramanian, Bhanu Chunduri, and Yashoteja Prabhu. I got to learn a lot from each one of you.

I am extremely thankful to Google for their generous financial support throughout my PhD. Because of their support, not only I was able to financially sustain throughout these years, but I could also easily attend conferences and workshops anywhere in the world without worrying about arranging funds. I would also like to acknowledge the administrative staff at the computer science department in IIT Delhi for their continuous help throughout these years.

It is almost impossible to think of successfully completing the PhD without the social support of friends. I was lucky enough to have an amazing group of friends at IIT Delhi. I am particularly grateful to Ankit Anand, Dinesh Khandelwal, Prachi Jain, Happy Mittal

and Yashoteja Prabhu for the endless laughs, heated political discussions, some amazing trips and numerous eat outs (thanks to the lovely mess food). I am also thankful to Kunal, Saurabh, Ashish, Dilpreet, Anup and many other fellow PhD students who made the stay at IIT Delhi a memorable one. I am also thankful to my friends outside IIT Delhi, particularly Varun, Aditya, Anuj, Brajesh and Ayush as they kept visiting and always paid for the restaurant bills. One thing that I eagerly looked forward to every year and that always replenished me was treks. So I am thankful to Vivek, Rahul, Shashank, Aditya and Ankit for the incredible trek experiences.

Finally, I can't thank my parents enough for the numerous sacrifices that they made so that I can get a good education. Without their support, encouragement and unflinching faith in me, none of this would have been possible. I am also thankful to my sister and brother-in-law for always supporting me and to all my family members who never asked me the dreaded question – *When are you finishing your PhD?*

Himanshu Jain

Abstract

The objective in extreme multi-label learning is to learn a classifier that can automatically tag a data point with the most relevant subset of labels from an extremely large label set. Extreme multi-label classification is an important research problem as it not only lets us tackle web-scale classification problems but it has also opened a new paradigm for solving ranking and recommendation problems. This thesis focusses on developing scalable algorithms and appropriate loss functions for extreme classification problems which can lead to state-of-the-art performance on large-scale ranking and recommendation applications. In particular, it makes the following contributions –

- 1) It proposes loss functions suitable for extreme multi-label learning that do not erroneously treat missing labels as irrelevant but instead provide unbiased estimates of the true loss function even when ground truth labels go missing under arbitrary probabilistic label noise models. These loss functions also naturally promote the accurate prediction of infrequently occurring, hard to predict, but rewarding tail labels.
- 2) It develops the SLEEC algorithm which is an embedding based extreme multi-label learning algorithm. SLEEC addresses some of the major limitations of embedding based methods such as high training and prediction cost and low prediction accuracy.
- 3) It develops the Slice algorithm for extreme multi-label learning with low-dimensional dense features that scales to 100 million labels and 240 million training points.
- 4) It reformulates the problem of recommending related queries on a search engine as an extreme classification task and demonstrates that Slice could significantly improve recommending related searches on Bing.

सार

चरम मल्टी-लेबल सीखने का उद्देश्य एक क्लासिफायरियर सीखना है जो स्वचालित रूप से एक अत्यंत बड़े लेबल सेट से सबसे प्रासंगिक सबसेट के साथ डेटा बिंदु को टैग कर सकता है। एक्सट्रीम मल्टी-लेबल वर्गीकरण एक महत्वपूर्ण शोध समस्या है क्योंकि यह न केवल हमें वेब-स्केल वर्गीकरण समस्याओं से निपटने में मदद करता है, बल्कि इसने रैंकिंग और अनुशंसा समस्याओं के समाधान के लिए एक नया प्रतिमान भी खोला है। यह थीसिस स्केलेबल एल्गोरिदम को विकसित करने और चरम वर्गीकरण समस्याओं के लिए उचित नुकसान के कार्यों पर ध्यान केंद्रित करती है, जिससे बड़े पैमाने पर रैंकिंग और सिफारिश अनुप्रयोगों पर अत्याधुनिक प्रदर्शन हो सकता है। विशेष रूप से, यह निम्नलिखित योगदान देता है -

- 1) यह चरम मल्टी-लेबल सीखने के लिए उपयुक्त नुकसान कार्यों का प्रस्ताव करता है जो लापता लेबल को अप्रासंगिक नहीं मानते हैं, लेकिन इसके बजाय जमीनी सच्चाई लेबल के मनमाने ढंग से लेबल शोर मॉडल के तहत गायब होने पर भी वास्तविक नुकसान फंक्शन के निष्पक्ष अनुमान प्रदान करते हैं। ये नुकसान कार्य स्वाभाविक रूप से होने वाली सटीक भविष्यवाणी को बढ़ावा देते हैं, भविष्यवाणी करना कठिन है, लेकिन पूंछ को पुरस्कृत करते हैं।
- 2) यह SLEEC एल्गोरिथम को विकसित करता है जो एक एम्बेडिंग आधारित मल्टी-लेबल लर्निंग एल्गोरिथम है। SLEEC उच्चतर प्रशिक्षण और भविष्यवाणी लागत और कम भविष्यवाणी जैसे अंतर्निहित तरीकों की कुछ प्रमुख सीमाओं को संबोधित करता है।
- 3) यह कम आयामी सघन सुविधाओं के साथ चरम मल्टी-लेबल सीखने के लिए स्लाइस एल्गोरिदम विकसित करता है जो 100 मिलियन लेबल और 240 मिलियन प्रशिक्षण बिंदुओं को मापता है।
- 4) यह खोज इंजन पर संबंधित प्रश्नों की सिफारिश करने की समस्या को एक चरम वर्गीकरण कार्य के रूप में सुधारता है और यह दर्शाता है कि बिंग पर संबंधित खोजों की सिफारिश करने में स्लाइस काफी सुधार कर सकता है।

Contents

Certificate	i
Acknowledgements	iii
Abstract	v
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Classification	1
1.2 Ranking and Recommendation as Extreme Classification	3
1.3 Challenges	4
1.3.1 Statistical Challenge: Extreme Multi-label Loss Functions	4
1.3.2 Computational Challenge: Extreme Multi-label Algorithms	6

1.3.3	Applications	6
1.4	Related Work	7
1.5	Contributions and Outline	10
2	Extreme Multi-label Loss Functions	13
2.1	Introduction	13
2.2	Related Work	15
2.3	A Motivating Example	16
2.4	Propensity Scored Losses	18
2.5	Propensity Model	26
2.6	Results	29
2.7	Conclusion	34
3	Embedding Based Approach to Extreme Classification	35
3.1	Introduction	35
3.2	Related Work	38
3.3	Method	40
3.3.1	Learning Embeddings	42
3.3.2	Scaling to Large-scale Data sets	45
3.4	Experiments	46
3.5	Conclusions and Future Work	51

4	1-vs-All Based Approach to Extreme Classification	53
4.1	Introduction	53
4.2	Related Work	55
4.3	Slice	57
4.4	Experiments	64
4.5	Conclusions	73
5	Extreme Classification Application: Related Searches	75
5.1	Introduction	75
5.2	Related Work	77
5.3	Experiments	78
5.4	Conclusion	84
6	Conclusion and Future Directions	85
	Bibliography	89
	List of Publications	99
	Biography	101

List of Figures

1.1	Ad landing page for Geico Car Insurance and some of the advertiser bid phrases	3
2.1	Plot showing the number of times each label occurs in a dataset: 246201 and 452262 labels occur less than 5 times each in Wikipedia and Amazon respectively. Such labels are harder to predict than popular ones but might also be more informative and rewarding in certain applications.	16
2.2	Propensities p_l and their corresponding weights $w_l = 1/p_l$ on Wikipedia and Amazon. The estimated propensities follow a sigmoidal curve on the semi-log plot and provide a principled setting of the weights for recommending rare items as compared to popular heuristics such as $N_l^{-\beta}$ and $\log(N/N_l)$	27
2.3	Plot showing the contribution of each label to the overall propensity scored Precision@1 and Precision@5. PfastXML is significantly more accurate at predicting infrequently occurring (small N_l) tail labels. Figure best viewed under magnification	31

-
- 2.4 (a) propensity curves used for simulating missing labels on the EUR-Lex dataset with each curve labelled with the corresponding percentage of missing labels; (b) propensity scored nDCG@k is unbiased; (c) propensity scoring improves training; and (d) training using incorrect propensities ($A \neq 0.55$) might be better than training without propensities. See text for details. Figure best viewed under magnification. 32
- 3.1 (a) error $\|Y - Y_{\hat{L}}\|_F^2 / \|Y\|_F^2$ in approximating the label matrix Y . Global SVD denotes the error incurred by computing the rank \hat{L} SVD of Y . Local SVD computes rank \hat{L} SVD of Y within each cluster. SLEEC NN objective denotes SLEEC’s objective function. Global SVD incurs 90% error and the error is decreasing at most linearly as well. (b) shows the number of documents in which each label is present for the WikiLSHTC-325K data set. There are about $300K$ labels which are present in < 5 documents lending it a ‘heavy tailed’ distribution. (c) shows Precision@1 accuracy of SLEEC and localLEML on the Wiki-10 data set as we vary the number of clusters. 41
- 3.2 Variation in Precision@1 accuracy with model size and the number of learners on large-scale data sets. Clearly, SLEEC achieves better accuracy than FastXML and LocalLEML-Ensemble at every point of the curve. For WikiLSTHC, SLEEC with a single learner is more accurate than LocalLEML-Ensemble with even 15 learners. Similarly, SLEEC with 2 learners achieves more accuracy than FastXML with 50 learners. 47
- 4.1 Plot showing the contribution of each label frequency ($\sum_i y_{il}$) to the overall Precision@3. Slice is more accurate than DiSMEC on tail labels (smaller ID). 70
- 4.2 The variation in Slice’s (a) accuracy, (b) training time & (c) prediction time on Amazon-670K with the size of the label shortlist $|\mathcal{S}|$. The variation in Slice’s accuracy with the bias parameter b is shown in (d). 72

5.1 Screenshots of Bing related searches results for various queries. 83

List of Tables

2.1	(a) presents unbiased propensity scored loss functions $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ corresponding to precision@k and nDCG@k for an unrestricted probabilistic label noise model which is the focus of this chapter. The unbiased losses in (b), including the Mean Reciprocal Rank (MRR) and the Average Discounted Gain (ADG), require either knowledge of $\mathbf{1}^\top \mathbf{y}^*$ or that labels go missing with probability $1 - g_l / \mathbf{1}^\top \mathbf{y}^*$ with known g_l (except for the F-score). Note that $\hat{\mathbf{y}}$ has only k non-zero entries for precision@k, nDCG@k and recall@k and that r_l represents the rank of label l in $\hat{\mathbf{y}}$	19
2.2	Dataset statistics	30
2.3	PfastXML has more unique labels C_k in the top $k = 1, 3$ and 5 predictions across all test points in a dataset as compared to FastXML indicating that it has better coverage of tail labels.	32
2.4	PfastXML makes significantly more accurate predictions as compared to FastXML and the Popularity baseline. Performance is evaluated according to the unbiased propensity scored Precision@k (P_k) and nDCG@k (N_k) for $k = 1, 3$ and 5.	33

3.1	Dataset Statistics: N and M are the number of training and test points respectively, D and L are the number of features and labels, respectively, and \bar{D} and \bar{L} are the average number of nonzero features and positive labels in an instance, respectively.	48
3.2	Precision Accuracies (a) Large-scale data sets : SLEEC is 35% more accurate in terms of P@1 and 22% in terms of P@5 than LEML, a leading embedding method. Other embedding based methods do not scale to the large-scale data sets; (b) Small-scale data sets : SLEEC consistently outperforms state of the art embedding based approaches. WSABIE, which also uses kNN classifier on its embeddings is significantly less accurate than SLEEC on all the data sets, showing the superiority of proposed embedding based algorithm.	50
3.3	Stability of SLEEC learners. Mean precision values over 10 runs of SLEEC on WikiLSHTC-325K with varying number of learners are shown. Each individual learner as well as ensemble of SLEEC learners was found to be extremely stable with with standard deviation ranging from 0.16% on P1 to 0.11% on P5.	51
4.1	Dataset statistics	64
4.2	Results on Amazon-670K dataset with high dimensional sparse bag-of-words features.	65
4.3	Results on extreme classification datasets.	67
4.4	Results on publically available datasets with 100 dimensional GloVE embeddings. Results are similar to what were obtained using XML-CNN embeddings.	68
4.5	Slice's precision@ k , nDCG@ k , training time and prediction time on the large-scale related searches datasets.	68
4.6	Results on small-scale related searches datasets.	69

4.7	Results on extreme classification datasets in terms of propensity scored precision@ k	71
4.8	Slice's coverage@ k on the RS-101M dataset.	71
4.9	Slice's accuracy does not vary much on Amazon-670K if HNSW was replaced by LSH or exact NN search.	73
5.1	Slice makes significantly more accurate predictions as compared to leading related searches algorithms in Bing.	79
5.2	Prediction accuracy of related searches algorithms becomes worse when their suggestion set is not restricted to set on which Slice was trained. . . .	80
5.3	Relative improvements in online metrics when Slice was added to the ensemble serving related searches on Bing.	80
5.4	Related searches recommendations by Bing and Slice. Bing recommended less than three suggestions for tail queries in (a) and (b) and recommended poor suggestions for input query in (c) while Slice provided eight relevant and diverse suggestions for all the input queries.	82