

**EFFICIENT AND ACCURATE
ALGORITHMS FOR EXTREME CLASSIFICATION**

YASHOTEJA PRABHU



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INSTITUTE OF TECHNOLOGY DELHI**

APRIL 2023

© Indian Institute of Technology (IITD), New Delhi, 2023

**EFFICIENT AND ACCURATE
ALGORITHMS FOR EXTREME CLASSIFICATION**

By

YASHOTEJA PRABHU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted

**in fulfilment of the requirements of the degree of Doctor of Philosophy
to the**



INDIAN INSTITUTE OF TECHNOLOGY DELHI

APRIL 2023

Certificate

This is to certify that the thesis titled **Efficient and Accurate Algorithms for Extreme Classification** being submitted by **Mr. Yashoteja Prabhu** for the award of **Doctor of Philosophy** in the Department of Computer Science and Engineering is a record of bona fide work carried out by him under my 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, work in Chapters 3 and 4 were done jointly with other Ph.D. students. Work in Chapter 4 was also done jointly with a Master's student. In each case, the part done by the collaborators appeared in their respective theses.

Manik Varma

Adjunct Professor

Department of Computer Science and Engg.

Indian Institute of Technology Delhi

New Delhi - 110016

Amit Kumar

Professor

Department of Computer Science and Engg.

Indian Institute of Technology Delhi

New Delhi - 110016

Acknowledgements

I am immensely grateful to my advisor Manik Varma who has been an advisor, a mentor, and a confidant to me over the years and to whom I owe most of my professional learnings including the importance of strong work ethics and attention-to-detail. His continued support has been of great help during the difficult phases of my life. I am also thankful to his family with whom I have shared delicious lunches and nice conversations.

I am also deeply grateful to my co-advisor Amit Kumar who was always available for guidance and who has supported me over the years through every step of my Ph.D. This degree could not have been completed without his help.

I am thankful to my research collaborators for contributing to my work and for sharing valuable lessons in coding, experimentation, teamwork *etc.* In particular, I thank Himanshu Jain, Shilpa Gopinath, Anil Kag, Kunal Dahiya, Shrutendra Harsola, Rahul Agrawal, Aditya Kusupati and Nilesh Gupta. I am also grateful to Purushottam Kar who kindly helped me to write my first research paper as well as to review this final thesis. I thank the professors and peers at IIT Delhi for useful technical discussions and suggestions throughout my Ph.D. I thank the CSE office at IIT Delhi for their utmost dedication towards attending to student needs. I especially thank Microsoft for greatly supporting my research work through meaningful collaborations with several product groups. I am indebted to Tata Consultancy Services for funding my Ph.D.; and to MSR India, IARCS and other organizations for providing travel grants to attend conferences and workshops.

This journey would not have been so enjoyable without the constant encouragement and support from my friends at work and in the hostel. I thank Ankit Anand, Arindam Bhattacharya, Anup Bhattacharya, Deepak Ravi, Dinesh Khandelwal, Happy Mittal, Himanshu Jain, Kunal Dahiya, Prachi Jain and many other colleagues who have been my

partners for relishing teas and delicious dinners, exchanging friendly banter and heated arguments, enjoying morning cycle rides and evening frisbee. I find myself extremely fortunate to have had a cohort of such talented and lively peers, and their good nature and dedication to their craft will continue to inspire me. I also thank my co-hostelites, Anil Patidar, Amit Mishra, Arup Roy, Jagatpreet Singh, Samarprit Chakraborty, Soumic Sarkar, and others, for all the tasty Aloo Buns we had together, and the happiness and miseries we jointly lived through. Please forgive me if I have missed anybody. These memories will continue to brighten my days.

I thank the people at MSR India for hosting me during my last few years of Ph.D. and for providing me the much-needed guidance and help. In particular, I thank Kalika Bali for kindly mentoring me over tasty lunches; all the Research Fellows and Research Engineers for collaborating with me on many projects; and the researchers and HR for all their contributions and assistance. I was greatly benefitted by the regularly organized research talks and discussions. I also immensely enjoyed the evening matches of table tennis and foosball and our trips to Sri Lanka and Mahabalipuram.

I thank my chums, Arjun, Manjunath and Rakesh, who never left my side since our school days and provided moral support during my Ph.D. I thank my dear parents and sister for always believing in me and supporting me through all my endeavors. Finally, I thank my lovely and loving wife, Rohini, whose kindness, gregariousness and conviction have left a permanent imprint on me. Her persistent interest in reading my thesis, specifically this acknowledgements section, has been the primary motivation for me to complete this thesis.

Yashoteja Prabhu

Abstract

Classification problems with an extremely large number of fine-grained classes or labels are frequently encountered in present-day applications. For example, data annotation tasks which involve millions of label choices such as Wikipedia categories or Flickr captions are helpful for effective webpage organization and retrieval. Additionally, large-scale recommendation and ranking tasks can also be naturally posed as classification problems by treating each item to be recommended or ranked as a separate label. Extreme classification is a thriving research area of machine learning which studies such large classification problems.

This thesis aims to provide viable algorithmic solutions to extreme classification by addressing its core technical challenges. Extreme classifiers need to train efficiently on millions of data points and labels, and predict in real-time as expected by web-based applications. At the same time, prediction accuracy should be maximized to enhance user satisfaction and revenue. Towards these goals, this thesis develops (1) approaches based on balanced data point partitioning trees whose training and prediction costs scale only logarithmically in the number of labels and hence scale to large datasets, (2) balanced label partitioning tree-based approaches that leverage powerful 1-vs-All classifiers to significantly boost prediction accuracy without compromising efficiency, and finally, (3) an approach based on novel sparse label indices to minimize model size and RAM consumption while ensuring high accuracy and logarithmic efficiency.

This thesis also explores multiple relaxations to the standard extreme classification paradigm in order to effectively leverage the additionally available meta-data. First, it proposes warm-start extreme classification which offers improved predictions for frequent inputs by leveraging label-side features and past interaction logs. Second, it develops the

extreme regression paradigm which relaxes the conventional binary relevance assumption to learn richer models from partial label relevances. Third, it extends extreme classification to handle zero-shot label prediction, a common setting in recommendation tasks.

The algorithms in this thesis are well-suited for real-world applications involving millions of labels. Empirically they were found to outperform existing state-of-the-art extreme classification algorithms on publicly available benchmark datasets. Moreover, in live flights on the Bing search engine, the proposed techniques were also found to significantly improve the key metrics when applied to Sponsored and Dynamic Search Ad applications

सार

आज के अनुप्रयोगों में बहुत बड़ी संख्या में सुक्ष्म वर्गों या लेबलों के साथ वर्गीकरण की समस्याएं अक्सर सामने आती हैं। उदाहरण के लिए, डेटा टिप्पणी कार्य जिसमें विकिपीडिया श्रेणियों या फ़्लिकर कैप्शन जैसे लाखों लेबल विकल्प शामिल होते हैं जो प्रभावी डेटा संगठन और पहुँच के लिए सहायक होते हैं। इसके अलावा, बड़े पैमाने पर चर संज्ञा और श्रेणी कार्यों को भी स्वाभाविक रूप से वर्गीकरण समस्याओं के रूप में पेश किया जा सकता है, प्रत्येक वस्तु को एक अलग लेबल के रूप में अनुशासित या श्रेणी कर सकते हैं। चरम वर्गीकरण मशीन लर्निंग का एक उभरता हुआ शोध क्षेत्र है जो इस तरह की बड़ी वर्गीकरण समस्याओं का अध्ययन करता है।

इस निबंध का उद्देश्य चरम वर्गीकरण के मूल तकनीकी चुनौतियों के हल के लिए व्यवहार्य एल्गोरिथम समाधान प्रदान करना है। चरम वर्गीकारक को लाखों डेटा बिंदुओं और लेबलों पर कार्य करने के लिए कुशलता से प्रशिक्षित करने की और वेब-आधारित अनुप्रयोगों द्वारा अपेक्षित समय में निष्कर्ष की आवश्यकता होती है। इसके अतिरिक्त, उपयोगकर्ता संतुष्टि और राजस्व को अधिकतम करने के लिए निष्कर्ष यथार्थता अधिक होनी चाहिए। इन लक्ष्यों की ओर, यह निबंध निम्नलिखित विकसित करता है: (1) संतुलित डेटा बिंदु विभाजन वाले पदानुक्रम पर आधारित दृष्टिकोण जिनके प्रशिक्षण और निष्कर्ष की लागत केवल लेबल की संख्या में लॉगरिदमिक रूप से बढ़ती है और इसलिए बड़े डेटासेट के पैमाने पर चलती है, (2) संतुलित लेबल पदानुक्रम-आधारित दृष्टिकोण जो मजबूत 1-विरुद्ध-सभी वर्गीकारक की लाभ उठाते हैं और, यथार्थ और मितव्ययी निष्कर्ष करते हैं, (3) उच्च यथार्थता और लॉगरिदमिक मितव्य सुनिश्चित करते हुए, मॉडल परिमाण और रैम की खपत को कम करने के लिए नई और विरल लेबल सूचकांकों पर आधारित एक दृष्टिकोण।

यह निबंध उपलब्ध मेटा-डेटा का प्रभावी ढंग से लाभ उठाने के लिए चरम वर्गीकरण में उपयोग की जाने वाली समस्या में विभिन्न छूटों की भी पड़ताल करती है। सबसे पहले, यह वार्म-स्टार्ट चरम वर्गीकरण का प्रस्ताव करता है जो बेहतर निष्कर्ष के लिए प्रकट लेबल डेटा का लाभ उठाता है। दूसरा, यह चरम रिग्रेसन विकसित करता है जो आंशिक लेबल प्रासंगिकता से समृद्ध मॉडल सीखने के लिए पारंपरिक बाइनरी प्रासंगिकता धारणा को शिथिल करता है। तीसरा, यह शून्य-शॉट लेबल को संभालने के लिए चरम वर्गीकरण का विस्तार करता है।

इस निबंध में प्रस्तावित चरम वर्गीकरण तकनीक लाखों लेबल वाले वास्तविक दुनिया के अनुप्रयोगों के लिए उपयुक्त हैं। प्रयोगों में, वे सार्वजनिक रूप से उपलब्ध बेंचमार्क डेटासेट पर मौजूदा अत्याधुनिक चरम वर्गीकरण एल्गोरिदम को बेहतर प्रदर्शन किया। इसके अलावा, बिंग सर्च इंजन पर लाइव फ्लाइट्स में, प्रस्तावित तकनीकों को प्रायोजित और

डायनामिक खोज विज्ञापन अनुप्रयोगों पर लागू करने पर प्रमुख मैट्रिक्स में महत्वपूर्ण सुधार करने के लिए भी पाया गया।

Contents

Certificate	i
Acknowledgements	ii
Abstract	iv
Abstract-hindi	vi
Contents	viii
List of Figures	xiii
List of Tables	xv
I Prologue	1
1 Introduction	2
1.1 Extreme Classification	3
1.1.1 Definition and Motivation	3
1.1.2 Research Challenges	5
1.1.3 Applications	7
1.2 Organization of Existing Work	9
1.3 Contributions and Thesis Outline	10

II	Extreme Classification using Point Trees	13
2	FastXML: A Fast and Accurate Point-Tree Extreme Classifier	14
2.1	Abstract	14
2.2	Introduction	15
2.3	Related Work	16
2.4	FastXML	18
2.4.1	FastXML overview	19
2.4.2	Learning to partition a node	23
2.4.3	Prediction	26
2.5	Optimizing FastXML	27
2.5.1	Optimizing with respect to \mathbf{r}^\pm	28
2.5.2	Optimizing with respect to δ	29
2.5.3	Optimizing with respect to \mathbf{w}	30
2.5.4	Finite termination	30
2.6	Experiments	32
2.7	Conclusions	42
3	PfastreXML: A Point-Tree based Extreme Classifier with Missing Label De-biasing	43
3.1	Abstract	43
3.2	Introduction	44
3.3	Propensity Scored Losses	45
3.4	Algorithms	47
3.4.1	Propensity scored FastXML	48
3.4.2	PfastreXML	50
3.5	Experiments	54
3.6	Conclusions	57
3.7	Appendix	58
3.7.1	PfastreXML derivations	62

4	SwiftXML: A Warm-Start Extreme Classifier with Label Feature Augmentation	67
4.1	Abstract	67
4.2	Introduction	68
4.3	Related Work	71
4.4	A Motivating Example	72
4.5	SwiftXML	72
4.6	Sponsored Search Advertising	76
4.7	Experiments	78
4.8	Conclusions	88
4.9	Appendix	88
4.9.1	Algorithms	88
4.9.2	Results	92
4.9.3	Derivations of Optimization Algorithms	101
III	Extreme Classification using Label Trees	108
5	Parabel: A Label Partitioning Tree based Extreme Classifier	109
5.1	Abstract	109
5.2	Introduction	110
5.3	Related Work	113
5.4	Parabel	115
5.4.1	Architecture	115
5.4.2	Learning the Label Hierarchy	115
5.4.3	A Hierarchical Probabilistic Model	118
5.4.4	Training	121
5.4.5	Prediction	125
5.5	Experiments	127
5.6	Conclusions	133
5.7	Appendix	133

5.7.1	Algorithms	133
5.7.2	Theorems and Proofs	138
5.7.3	Results	147
6	XReg: A Label-Tree based Extreme Classifier for Partial Relevance	
	Modeling	152
6.1	Abstract	152
6.2	Introduction	153
6.3	Related Work	156
6.4	Extreme Regression Metrics	158
6.5	XReg: eXtreme Regressor	160
6.5.1	Label Tree Construction	161
6.5.2	A Probabilistic Regression Model	161
6.5.3	Pointwise Inference	163
6.5.4	Labelwise Inference	164
6.6	Experiments	165
6.7	Conclusions	171
6.8	Appendix	172
6.8.1	Theorems and Proofs	176
IV	Extreme Classification using Sparse Label indices	184
7	ZestXML: A Generalized Zero-shot Extreme Classifier	185
7.1	Abstract	185
7.2	Introduction	186
7.3	Related Work	190
7.4	Proposed Algorithm: ZestXML	192
7.4.1	Problem Setup	192
7.4.2	Training	194
7.4.3	Prediction	197
7.5	Sponsored Search Advertising	198

7.6	Experiments	200
7.6.1	Experiment Settings	202
7.6.2	Results	204
7.7	Conclusion	208
7.8	Appendix	211
V	Epilogue	216
8	Conclusions and Future Directions	217
8.1	Conclusions	217
8.2	Future Directions	218
	Bibliography	220
	List of Publications	238
	Biography	240

List of Figures

2.1	The variation in FastXML’s precision at 5 with the number of trees selected according to random order; and highest individual prediction accuracy on the training set. The training time can be halved on most data sets with a minimal decrease in prediction accuracy by training only 25 trees in random order.	33
3.1	Plots showing the contribution of each label to the overall propensity scored precision@3. PfastreXML is significantly more accurate at predicting infrequently occurring (small N_l) tail labels. Figure best viewed under magnification.	57
4.1	Item recommendations by PfastreXML and SwiftXML on AmazonCat, Amazon and Bing Ads: PfastreXML predictions are frequently irrelevant due to lack of informative user features (<i>e.g.</i> (a)), emphasis on the wrong features (<i>e.g.</i> (d)) and inability to disambiguate homonyms (<i>e.g.</i> (e)). SwiftXML leverages item correlations (<i>e.g.</i> ”Marx” => ”Karl” in (b)) and helpful information from revealed items and their features (<i>e.g.</i> (a)-(f)) to make much more accurate predictions. See text for more details. Figure best viewed under high magnification.	83
5.1	Parabel can improve the quantity, quality and diversity of predicted queries from ad landing pages for DSA on Bing.	128

6.1	Precision-Recall curves showing that XReg is consistently better than XReg-Zero and Parabel approaches for precision recall tradeoff.	172
7.1	Plot of long query tail in Bing Advertising. Most search queries have few or no previously clicked ads.	190
7.2	Number of parameters vs frequency of label	200
7.3	ZestXML learns most compact model on both Wikipedia-1M and Amazon-1M datasets and is comparable to most scalable extreme classification methods w.r.t training/prediction time (* marked methods were run on GPU)	201
7.4	Hyperparameter ablation	206
7.5	ZestXML consistently outperforms the baselines across both zero shot and few shot labels on Amazon-1M	207
7.6	Item recommendations by ZestXML on Amazon-2M: In each figure, first table highlights the top predictions of ZestXML and the second table provides the top active point-label feature pairs. See text for more details. Figure best viewed under high magnification.	209

List of Tables

2.1	Data set statistics	32
2.2	Results on small and medium data sets. FastXML was run with default hyper-parameter settings on all data sets. FastXML-T presents results when the parameters were tuned.	37
2.3	Results on large data sets comparing the performance of FastXML to LPSR trained with Naïve Bayes as the base classifier.	38
2.4	FastXML’s wall clock training time (in hours) vs the number of cores used on a single machine.	39
2.5	The variation in FastXML’s performance with the number of training iterations. \mathcal{W}_i denotes the iteration at which \mathbf{w} is updated for the i^{th} time at the root node on the Ads-430K data set. Precision values and training times are reported for the full ensemble.	39
2.6	FastXML learns more stable and balanced trees than MLRF and LPSR leading to both faster training as well as faster prediction. Tree balance is measured as $H/\log(N/\text{MaxLeaf})$, where H is the average length of the path traversed by a point in that tree and $\log(N/\text{MaxLeaf})$ is the average length of a path traversed in a perfectly balanced tree with at most MaxLeaf points at each leaf node. Smaller values of tree balance are better with a balance of 1 indicating a perfectly balanced tree.	40

2.7	Results obtained by replacing the $nDCG@L$ loss function in FastXML with others such as $nDCG@5$ (FastXML- $nDCG5$) or precision at 5 (FastXML-P5) and by replacing the Gini index in MLRF with the proposed $nDCG@L$ loss function.	41
3.1	(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 $\hat{\mathbf{y}}$	47
3.2	Dataset statistics	48
3.3	The proposed PfastreXML and PfastXML algorithms make significantly more accurate predictions as compared to state-of-the-art SLEEC, FastXML and other baseline algorithms. PfastreXML’s predictions are more accurate than PfastXML’s with negligible training and prediction overheads. Performance is evaluated according to the unbiased propensity scored $Precision@k$ (P_k) and $nDCG@k$ (N_k) for $k = 1, 3$ and 5	52
3.4	PfastreXML 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 SLEEC or FastXML indicating that it has better coverage of tail labels.	56
3.5	The proposed PfastreXML and PfastXML algorithms make significantly more accurate predictions as compared to state-of-the-art SLEEC, FastXML and other baseline algorithms. PfastreXML’s predictions are more accurate than PfastXML’s with negligible training and prediction overheads. Performance is evaluated according to $Precision@k$ (P_k) and $nDCG@k$ (N_k) for $k = 1, 3$ and 5	60
4.1	Dataset statistics	79

4.2	SwiftXML can be up to 14% and 37% more accurate as compared to state-of-the-art extreme classifiers and warm-start recommendation algorithms respectively according to unbiased propensity-scored Precision@5 (PSP5). Results for PSP1, PSP3 and biased Precision@ k are presented in the Section 4.9.	85
4.3	SwiftXML can be up to 14% and 4% more accurate according to unbiased propensity scored Precision@5 as compared to baseline extensions of PfastreXML incorporating label features via early and late fusion respectively. Results for other metrics, including biased Precision@ k , are reported in the Section 4.9.	86
4.4	SwiftXML could increase the relative click-through-rate (CTR) and relative quality of ad recommendation (QOA) by 10% while simultaneously reducing the bounce rate (BR) by 30% on sponsored search on Bing. . .	87
4.5	The proposed SwiftXML performs consistently better, across different revealed label percentages, as compared to baseline PfastreXML extensions: PfastreXML-early and PfastreXML-late. Performance is evaluated according to the unbiased propensity scored Precisions (PSP1,PSP3,PSP5). . .	93
4.6	The proposed SwiftXML makes significantly more accurate predictions as compared to both state-of-the-art extreme classifiers and classical recommendation algorithms. SwiftXML consistently improves as more and more test labels are revealed, and achieves accuracy gains of upto 14% as compared to the baselines. Performance is evaluated using unbiased propensity-scored Precision (PSP1,PSP3,PSP5).	94
4.7	The proposed SwiftXML makes significantly more accurate predictions as compared to both state-of-the-art extreme classifiers and classical recommendation algorithms. SwiftXML consistently improves as more and more test labels are revealed, and achieves accuracy gains of upto 14% as compared to the baselines. Performance is evaluated using unbiased propensity-scored nDCG (PSN1,PSN3,PSN5).	95

4.8	The proposed SwiftXML performs consistently better, across different revealed label percentages, as compared to baseline PfastreXML extensions: PfastreXML-early and PfastreXML-late. Performance is evaluated according to the unbiased propensity scored nDCGs (PSN1,PSN3,PSN5). . . .	96
4.9	The proposed SwiftXML makes significantly more accurate predictions as compared to both state-of-the-art extreme classifiers as well as classical recommendation algorithms. SwiftXML consistently improves as more and more test labels are revealed, and achieves accuracy gains of upto 3% as compared to the baselines. Performance is evaluated using standard precisions (P1,P3,P5).	97
4.10	The proposed SwiftXML makes significantly more accurate predictions as compared to both state-of-the-art extreme classifiers as well as classical recommendation algorithms. SwiftXML consistently improves as more and more test labels are revealed, and achieves accuracy gains of upto 3% as compared to the baselines. Performance is evaluated using standard nDCG metrics (N1,N3,N5).	98
4.11	The proposed SwiftXML performs consistently better, across different revealed label percentages, as compared to baseline PfastreXML extensions which make use of label features. Performance is evaluated according to the standard Precisions (P1,P3,P5).	99
4.12	The proposed SwiftXML performs consistently better, across different revealed label percentages, as compared to baseline PfastreXML extensions which make use of label features. Performance is evaluated according to the standard nDCG metrics (N1,N3,N5).	100
5.1	Dataset statistics	127

5.2	Parabel is significantly faster at training and prediction than state-of-the-art extreme classifiers while having almost the same precision@ $r = 1, 3, 5$ values. Results are reported for Parabel with $T = 1, 3$ trees trained using the log loss (l) and the squared hinge loss (s). XML-CNN times are not directly comparable as it was trained on a GPU. Please see the text for details.	129
5.3	Alternative choices of Parabel’s components leads to worse performance. Results have been reported in terms of precision@5. Please see the text for details.	130
5.4	The relative improvement of Parabel over BM25 on Dynamic Search Advertising on Bing.	130
5.5	Results of Parabel and baseline algorithms on benchmark datasets where data points were represented by dense deep XML-CNN [95] embeddings. Parabel is significantly more accurate than tree and embedding based baselines. Parabel is also $2x - 500x$ faster at training and $150x$ faster at prediction as compared to 1-vs-All classifiers while being up to 4% worse in terms of precisions.	148
5.6	Results comparing Parabel’s performance to tree, embedding and 1-vs-All based baseline algorithms where accuracy is measured in terms of precision@ r (Pr), nDCG@ r (Nr), propensity-scored precision@ r ($PSPr$) and propensity-scored nDCG@ r ($PSNr$). All numbers are in percentages.	149
5.7	Parabel variants on EURLex-4K	150
5.8	Variation in Parabel’s performance with the number of label trees, <i>i.e.</i> hyperparameter T , on WikiLSHTC dataset. Parabel’s accuracy increases by 2% with an ensemble of 3 trees and witnesses diminishing returns with more trees.	150
5.9	Variation in Parabel’s performance with the number of maximum labels in the leaf nodes, <i>i.e.</i> hyperparameter M , on WikiLSHTC dataset. Both accuracy and test time increase with larger M , with Parabel achieving around $1ms$ test time per point and minimal loss in accuracies at $M = 100$. Results are reported for Parabel-s-T=3 with 3 trees.	151

5.10	Variation in Parabel’s performance with the beam search width, <i>i.e.</i> hyperparameter P , on WikiLSHTC dataset. Higher P values indicate more thorough tree search. Parabel accuracy initially increases with P and quickly saturates at around $P = 10$. Results are reported for Parabel-s-T=3 with 3 trees.	151
6.1	Dataset statistics	165
6.2	XReg achieves the best or close to the best ranking and regression performance in both pointwise (“p”) and labelwise (“l”) prediction settings. Re-ranking with tail classifiers (XReg-t) further improves the performance in many cases. More results are in the Section 6.8.	166
6.3	XMAD@ k is a better indicator of the filtering and re-ranking qualities than purely ranking metrics like WP@ k or traditional regression metrics like MAD.	168
6.4	Ranking regret at k is up to 69x more closely bounded by $2 \cdot \text{XMAD}@2k$ compared to the traditional MAD as proposed in Section 6.4. $k = 5$, “p”: pointwise, “l”: labelwise and “t”: use of tail classifiers. Please refer to the text for details.	168
6.5	The ablation study of Parabel leading to per-label XReg-t which clearly outperforms its predecessors on ranking metrics.	169
6.6	XReg has the best or close to the best ranking and regression performance across all the datasets compared to state-of-the-art extreme classifiers and large-scale regressors and rankers. Re-ranking with tail classifiers (XReg-t) further improves the accuracies. PSP@ k , CTR@ k and Rating@ k are variants of WP@ k as discussed in Section 6.4. “p”: pointwise, “l”: labelwise.	174
6.7	Hyperparameter tuning for # trees (T), Max leaf labels (M), Beam width (P) and points reaching leaf node per label in labelwise prediction of XReg. Note: The hyperparameters in bold face are finally chosen for the default setting.	175

7.1	Dataset Statistics.	200
7.2	ZestXML achieves the highest or close to the highest PSP for Generalized Zero-Shot(G.ZSL) task among other XML and dense ANNS baselines on all datasets	202
7.3	Comparing the shortlisting performance of ZestXML XHTTP with popular IR samplers TF-IDF, BM25	202
7.4	Comparison of ZestXML with other dense ANNS and XML algorithms on proprietary Bing Ads-31M dataset	208
7.5	Comparison of ZestXML with other ZSL and XML algorithms	210
7.6	Comparison of ZestXML with other ZSL and XML algorithms on proprietary Bing Ads-31M dataset	210
7.7	Comparison of ZestXML with zero-shot multi label algorithms on EURLex-4.3K	211