Extreme Regression for Dynamic Search Advertising

Yashoteja Prabhu [1 2] **Aditya Kusupati** [3] **Nilesh Gupta** [1] **Manik Varma** [1 2]

Abstract

This paper introduces a new learning paradigm called eXtreme Regression (XR) to accurately predict the numerical degrees of relevance of an extremely large number of labels to a data point in contrast to the recently popular extreme classifiers which incorrectly assume strictly binary-valued label relevances. Traditional regression metrics are unsuitable for XR problems since they could give extremely loose bounds for the label ranking quality. Also, the existing regression algorithms won't efficiently scale to millions of labels. This paper addresses these limitations through: (1) *new evaluation metrics* for XR; (2) a *new algorithm* called XReg which decomposes XR into a hierarchy of much smaller regression problems. This paper also introduces a (3) *new labelwise prediction algorithm* in XReg useful for recommendation tasks like Dynamic Search Advertising (DSA). Experiments on benchmark datasets demonstrated that XReg can outperform the state-of-the-art extreme classifiers as well as large-scale regressors and rankers by up to 50% reduction in the new XR error metric, and up to 2% and 2.4% improvements in terms of the propensity-scored precision in extreme classification and the click-through rate in DSA respectively. Deployment of XReg on DSA in Bing resulted in a relative gain of 27% in query coverage. XReg Prabhu et al. (2020) was originally published at WSDM 2020. XReg's source code can be downloaded from http://manikvarma.org/code/XReg/download.html.

1. Introduction

Objective: This paper introduces a new learning paradigm called eXtreme Regression (XR) which can provide elegant solutions to many large-scale ranking and recommendation applications including Dynamic Search Advertising (DSA). To effectively solve XR problems, this paper also develops new evaluation metrics and a new highly scalable and accurate algorithm called XReg.

eXtreme Regression: The objective of eXtreme Regression is to learn to accurately predict the numerical degrees of relevance of an extremely large number of labels with respect to a data point. Many large-scale ranking and recommendation applications can naturally be reformulated as XR problems. For example, the task of DSA can be posed as the problems of predicting the search queries' click probabilities for an ad. This qualifies as XR problems since the total number of queries can potentially be in millions. The predicted relevance estimates could then be used to recommend the most relevant labels to a data point which is the desired end goal of recommendation systems. Alternatively, the recommendations can also be further refined by filtering off less relevant ones or by re-ranking them to improve their relevance, and the relevance estimates provide principled ways of achieving these. To successfully solve an XR problem, new algorithms which could train and predict efficiently over millions of labels as well as millions of data points while also maintaining high prediction accuracy are required. Furthermore, the definition of accuracy, or equivalently regression error, needs to be redefined for XR settings where both the relevant labels and the desired label recommendations are extremely small.

DSA: DSA is a format of search advertising where the ads to be shown against a search query, along with the associated ad-copy, ad-title, bid-phrases *etc.*, are algorithmically obtained by leveraging the content from the ad landing pages. This saves considerable efforts for advertisers, results in faster deployment of new ad campaigns and enables more accurate user targeting. The ads shown by DSA algorithms need to be highly relevant and generate user clicks for the given query in order to earn revenue for the search engine and satisfy the users and advertisers. In addition, these algorithms need to train and predict very efficiently in order to scale to billions of ads and millions of search queries across multiple markets and maintain milliseconds' prediction latencies. This paper solves DSA as an XR task of estimating the click probabilities for the query, ad pairs by using the new XReg algorithm.

eXtreme Regression metrics: This paper proposes new regression metrics for XR which serve as good proxies for

[1] Microsoft Research, India [2] Indian Institute of Technology Delhi, India [3] University of Washington, USA. Correspondence to: Yashoteja Prabhu <t-yaprab@microsoft.com>.

Workshop on Extreme Classification Theory and Applications at 37^{th} International Conference on Machine Learning, Vienna, Austria, 2020. Copyright 2020 by the author(s).

the ranking accuracy and for the qualities of the subsequent label filtering and re-ranking steps. These metrics average of the largest few regression errors which are usually caused by highly underestimating or highly overestimating the relevances of the most or the least relevant labels which in turn degrade the ranking quality. The new XMAD@k metric can give up to 69x tighter bounds over ranking regret than MAD. These new metrics can guide the crucial steps in XR.

eXtreme Regressor algorithm: This paper also develops a new eXtreme Regressor (XReg) algorithm which can efficiently regress on to millions of label relevance weights in only logarithmic time. XReg hierarchically clusters the labels into a balanced tree and learns approximate regressors in each tree node which are common to all the labels in the node. Due to high label sparsity, each data point only participates in a logarithmic number of tree nodes which can lead to a significant speed up during both training and prediction by using appropriate algorithms. XReg essentially extends the state-of-the-art Parabel extreme classifier to the regression setting. XReg consistently outperforms extreme classifiers, large-scale regressors and rankers in terms of ranking accuracy. On a DSA dataset with 5M ads & 1M queries, XReg can train within just 20 hours using 1 core, predict in just 3 ms per query and give up to 58% & 27% lifts in revenue and query coverage when deployed online.

Labelwise inference: The standard prediction scenario involves recommending the most relevant labels for a test point, referred here as pointwise prediction, but applications such as DSA and movie recommendation can more naturally be posed in the reverse manner of predicting the most relevant ads or movies (*i.e.* test points) for each query or user (*i.e.* each label), referred here as labelwise prediction. On these tasks, pointwise prediction might recommend a small set of highly popular labels that are relevant to all test points resulting in low label coverage.

Contributions: This paper: (a) introduces a new learning paradigm called eXtreme Regression (XR) and reformulates tagging, movie recommendation and DSA applications as XR problems; (b) develops new evaluation metrics and a highly scalable and accurate algorithm called XReg to effectively tackle XR problems; and (c) demonstrates that XReg can significantly improve revenue and query coverage on Bing DSA when deployed in production.

Please refer to Prabhu et al. (2020) for a more comprehensive motivation and related work.

2. Extreme Regression Metrics

Notation: Let an XR dataset comprise N data points $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a D dimensional feature vector and $\mathbf{y}_i \in [0, \infty)^L$ is a ground truth relevance weight vector for point i. The weight y_{il} measures the true degree of relevance of label l to point i, with higher values indicating higher relevance. Similarly, let $\hat{\mathbf{y}}_i \in [0, \infty)^L$ denote the predicted relevance weight vector for point i. The function $S(\mathbf{v}, k)$ indicates the *ordered* index set of the k highest scoring labels in a score vector $\mathbf{v} \in [0, \infty)^L$.

Extreme regression metrics: Let, $\mathbf{e}_i$ be the vector of regression errors where $e_{il} = |\hat{y}_{il} - y_{il}|$. The new XR metrics, eXtreme Mean Absolute Deviation at k (XMAD@k) and eXtreme Root Mean Square Error at k (XRMSE@k) are defined as follows:

$$\text{XMAD@}k(\hat{\mathbf{y}}_i, \mathbf{y}_i) = \frac{1}{k} \sum_{l \in S(\mathbf{e}_i, k)} e_{il} \tag{1}$$

$$\text{XRMSE@}k(\hat{\mathbf{y}}_i, \mathbf{y}_i) = \sqrt{\frac{1}{k} \sum_{l \in S(\mathbf{e}_i, k)} e_{il}^2} \tag{2}$$

For ease of discussion, this paper mainly focusses on the XMAD metric, although most of the observations and results also apply to XRMSE. XMAD@k averages the k maximum regression errors but is minimized when all the L label relevances are predicted exactly right. The following lemma shows that XMAD serves as a good proxy for the ranking error. This is based on an intuition that the ranking errors at the top occur mainly due to either highly underestimating or highly overestimating the relevances of the most or the least relevant labels respectively leading to high regression errors on such labels. Prabhu et al. (2020) further shows that the bounds on ranking regret are much tighter when the proposed extreme regression metrics are used along with extension to create labelwise metrics.

3. XReg: eXtreme Regressor

This section describes XReg's key components including the label tree construction, the probabilistic regression model and the pointwise and labelwise prediction algorithms using the same model. XReg learns a small ensemble of up to 3 label trees quite similarly to Parabel (Prabhu et al., 2018).

3.1. A Probabilistic Regression Model

XReg is a regression method which takes a probabilistic approach to estimating the label relevance weights. Firstly, all the relevance weights are normalized to lie between 0 and 1 by dividing by its maximum value, thus allowing them to be treated as probability values. Note that while click-through rates in DSA are already valid probabilities, the inverse propensities and the user rating could exceed 1. Also, note that the predicted estimates can be easily scaled back since no information is lost due to this normalization.

XReg treats the normalized relevance weights for each label as the marginal probability of its relevance to a data point, which is, in fact, the case in DSA. This allows XReg to minimize the KL-divergence between the true and the predicted marginal probability for each label with respect to each data point. KL-divergence (Kullback & Leibler, 1951) measures how close 2 distributions are and is minimized when the 2 are identical, thus justifying its use while regressing on to probability values.

To reduce the complexity of naive 1-vs-All approach, XReg leverages the previously trained label tree. XReg expresses the marginal probability of a label as the probability that a data point traverses the tree path starting from the root to the label. Let the path from root to label l consist of nodes $n_{l1}, \cdots, n_{lH}$ where H is tree height, n_{l1} is the root and n_{lH} is the leaf node containing solely label l. Let z_{lh} denote the probability that a data point $\mathbf{x}$ visits the node n_{lh} after it has already visited the parent $n_{l(h-1)}$. Then the true marginal probability y_l that the label l is relevant to $\mathbf{x}$ is equivalent to $y_l = \prod_{h=1}^{H} z_{lh}$. Similar equality holds for predicted marginal probability: $\hat{y}_l = \prod_{h=1}^{H} \hat{z}_{lh}$. XReg then learns to minimize an upper bound on the KL-divergence between the two.

The unvisited node assumption formalizes the observation that the children of an unvisited internal node will never be traversed and that the labels in an unvisited leaf node will never be visited by a data point (Prabhu et al., 2018). Due to the above theorem, XReg can separately minimize the KL-divergence over the true and predicted probabilities that a data point takes a particular edge in the tree, and still end up minimizing the KL-divergences over each of the individual marginal label probabilities. The true probability value of edge traversal z_{lh} is essentially the probability that the data point visits any of the labels in the subtree rooted at the node indexed lh. We instantiate it to be equal to the largest marginal probability of any label in the subtree, by assuming the worst-case scenario that labels in each subtree are fully correlated, which promotes model robustness.

The KL-divergence minimization is mathematically equivalent to training a logistic regressor for estimating z_{lh} values for each tree edge where every data point is duplicated with weights z_{lh} and $1 - z_{lh}$:

$$\min_{\mathbf{w}_n} \|\mathbf{w}_n\|^2 + \frac{C}{|\mathcal{I}_n|} \sum_{i \in \mathcal{I}_n} \{ s_{in} z_{in} \log(1 + \exp(-\mathbf{w}_n^\top x_i)) + \tag{3}$$

$$s_{in}(1 - z_{in}) \log(1 + \exp(+\mathbf{w}_n^\top x_i)) \} \tag{4}$$

where n is used to index the node instead of lh, $\mathcal{I}_n$ only include those points which reach the node n. The problem in (Eq. 3) is strongly convex and was optimized using the modified CDDual algorithm available from Liblinear package (Fan et al., 2008). To summarize, each internal node in XReg contains 2 1-vs-All regressors which give the probability that a data point traverses to each of its children, each leaf node contains M 1-vs-All regressors which gives the conditional probability of each label being relevant given the data point reaches its leaf.

We make a mild assumption that each data point has at most $O(\log L)$ positive labels is made which is often valid on extreme learning datasets. As a result, each data point traverses at most $O(\log^2 L)$ tree edges, which directly leads to a huge reduction in training complexity thus resulting in $O(N\hat{D}\log^2 L)$ where $\hat{D}$ is the average number of non-zero features per data point.

XReg supports both pointwise and labelwise inference depending on the task at hand and more details can be found in Prabhu et al. (2020).

4. Experiments

Please refer to Prabhu et al. (2020) for a detailed disussion on datasets and baselines along with hyperparameters, metrics and hardware.

Results on benchmark datasets: Table 1 compares XReg's performance to diverse baselines on datasets belonging to tagging, recommendation and DSA applications. In terms of prediction accuracy, XReg consistently achieves close to best performance in terms of WP@5, Tau@5 as well as XMAD@5 metrics. In particular, XReg can be up to 2.4%, 3.89% and 2x better than all baselines in WP@5, Tau@5 and XMAD@5 respectively.

On most tagging datasets, XReg scores within 2% of the state-of-the-art ProXML (Babbar & Schölkopf, 2018) in terms of the popular PSP@5 metric but can be up to 1000x faster during both training and prediction.

XReg consistently outperforms extreme classifiers like Parabel and DiSMEC (Babbar & Schölkopf, 2017) which train only on binary labels. In particular, XReg can be up to 9% and 45% better than Parabel over pointwise and labelwise datasets in terms of WP@5. The larger gains on labelwise datasets are due to pointwise prediction in Parabel which can lead to low label coverage, especially on datasets like MovieLens with only 8K test points but around 140K labels. Owing to similar classifier architectures, XReg can be highly efficient just like Parabel. XReg is at most 3.75x and 4.8x slower during training and prediction and has at most 2.15x the model size as Parabel.

Owing to their high scalability, both Parabel and XReg scale to the largest DSA-1M dataset where none of the other approaches scale. On this dataset, XReg has 50% smaller XMAD@5 than Parabel.

XReg-t denotes the re-ranked XReg where the predicted relevance estimates are combined with tail classifier scores to improve ranking performance over more informative tail labels. XReg-t consistently improves performance over XReg since most XR datasets are dominated by tail labels. XReg-t can be up to 5.66% and 5.58% better than XReg in terms of PSP@5 and Tau@5. However, XReg-t often increases XMAD@5 over XReg since tail classifiers are

Table 1. 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. More results are in the full paper.

Method	PSP-p@5 (%)	Tau-p@5 (%)	XMAD-p@5	Training time (hrs)	Test time /point (ms)	Model size (GB)
			BibTex			
PfastreXML	59.75	53.68	**0.3151**	0.0050	0.2348	0.0246
Parabel	57.36	51.48	0.3372	0.0015	0.1945	0.0035
LEML	56.42	51.58	0.3520	0.0229	0.1737	0.0032
1-vs-all-LS	**60.14**	**54.21**	0.3337	**0.0007**	0.1137	0.0023
RankSVM	59.12	52.58	0.7089	0.0015	**0.0719**	0.0023
DiSMEC	57.23	51.47	0.3371	0.0004	0.0951	**0.0012**
ProXML	58.30	-	-	-	-	-
XReg	58.61	52.35	0.3158	0.0035	0.1642	0.0030
XReg-t	58.77	52.46	0.3386	0.0025	0.1256	0.0043
			EURLex-4K			
PfastreXML	45.17	48.85	0.1900	0.0887	1.3891	0.2265
Parabel	48.29	50.75	0.4227	**0.0245**	**1.1815**	0.0258
LEML	32.30	37.24	0.2115	0.3592	4.4483	0.0281
1-vs-all-LS	**52.27**	**53.96**	**0.1744**	0.1530	4.5378	0.1515
RankSVM	46.70	51.43	1.1967	0.1834	4.7635	0.1470
DiSMEC	50.62	52.33	0.4308	0.0999	1.9489	**0.0072**
ProXML	51.00	-	-	-	-	-
XReg	49.72	52.86	0.1849	0.0642	1.2899	0.0378
XReg-t	50.40	53.45	0.2132	0.0544	1.2074	0.0692
			Wiki10-31K			
PfastreXML	15.91	20.29	0.5705	0.3491	11.6855	0.5466
Parabel	13.68	19.83	0.7085	**0.3204**	**3.7275**	0.1799
LEML	13.05	20.06	0.5716	0.9546	54.9470	0.5275
1-vs-all-LS	21.89	26.71	**0.5459**	2.4341	129.8342	16.9871
RankSVM	18.46	25.84	1.2236	4.9631	92.2684	10.8536
DiSMEC	15.61	22.43	0.7140	2.1945	13.8993	**0.0290**
XReg	16.94	24.97	0.5716	0.6184	3.7649	0.3218
XReg-t	**22.60**	**30.55**	0.5506	0.6431	5.4910	0.9026
			WikiLSHTC-325K			
PfastreXML	28.04	36.38	0.1437	7.1974	6.9045	13.3096
Parabel	37.22	41.71	0.2459	**1.2195**	**2.2486**	**3.0885**
DiSMEC	39.50	-	-	-	-	-
ProXML	**41.00**	-	-	-	-	-
XReg	36.92	41.62	**0.1411**	4.5119	3.0312	3.5105
XReg-t	40.33	**43.39**	0.3140	3.8552	3.0896	4.1955
			Amazon-670K			
PfastreXML	28.53	30.97	0.4019	3.3143	11.4931	9.8113
Parabel	32.88	31.32	0.4292	**0.5815**	2.3419	**1.9297**
DiSMEC	34.45	31.94	0.4275	373	1414	3.7500
ProXML	**35.10**	-	-	≈ 1200	≈ 1000	-
XReg	33.24	34.72	**0.3869**	1.4925	2.4633	3.4186
XReg-t	34.29	**35.83**	0.4473	1.1864	**2.2242**	4.5952

Method	CTR-p@5 (%)	Tau-p@5 (%)	XMAD-p@5	Training time (hrs)	Test time /point (ms)	Model size (GB)
			SSA-130K			
PfastreXML	27.79	23.77	0.0655	1.3765	5.2419	1.6258
Parabel	**32.97**	**30.25**	0.1430	**0.2283**	1.9098	**0.3625**
LEML	6.54	8.10	**0.0654**	8.3253	161.6891	1.1308
RankSVM	13.06	14.03	2.7871	9.6026	130.0945	7.4834
DiSMEC	32.75	29.16	0.1562	31.4358	61.0967	0.0802
XReg	32.39	28.27	0.0684	0.4570	7.4715	0.7871
XReg-t	32.81	28.73	0.1131	0.5049	**1.7746**	1.4156

Method	Rating-l@5 (%)	Tau-l@5 (%)	XMAD-l@5	Training time (hrs)	Test time /point (ms)	Model size (GB)
			YahooMovie-8K			
PfastreXML	10.18	19.72	0.6286	**0.0241**	8.5074	0.0753
Parabel	9.73	28.22	0.6284	0.0299	**0.9639**	0.1307
LEML	21.79	28.85	0.6408	0.0593	5.3650	0.0586
1-vs-all-LS	21.63	31.24	0.6269	0.0740	6.8841	1.6977
RankSVM	24.88	33.28	1.0579	0.1282	5.1620	0.7172
DiSMEC	24.53	32.75	0.6207	0.0337	3.4258	0.0376
XLR	4.66	10.72	0.6716	-	4.7724	**0.0293**
XReg	25.86	35.00	0.6248	0.0685	4.1965	0.2829
XReg-t	**26.05**	**35.33**	**0.6185**	0.0615	3.6353	0.4500
			MovieLens-138K			
PfastreXML	7.25	22.84	0.9199	0.4514	19.8270	0.1837
Parabel	3.51	37.80	0.9200	1.7790	**1.6132**	3.4322
LEML	43.19	64.78	0.8722	**0.4186**	91.4262	0.2535
1-vs-all-LS	42.16	63.92	0.8832	2.5756	121.6169	16.1334
DiSMEC	45.35	61.55	0.8857	1.5437	74.9537	1.0514
XLR	9.67	21.42	0.9134	4.579	68.347	**0.0634**
XReg	48.94	66.99	0.8741	2.6287	7.7996	3.6223
XReg-t	**49.29**	**67.36**	**0.8285**	2.7437	9.8279	4.8958

Method	CTR-l@5 (%)	Tau-l@5 (%)	XMAD-l@5	Training time (hrs)	Test time /point (ms)	Model size (GB)
			DSA-130K			
PfastreXML	28.18	**34.75**	0.0422	1.3765	5.2419	1.6258
Parabel	33.97	28.37	0.0891	**0.2283**	**1.9098**	0.3625
LEML	10.36	7.70	**0.0415**	8.3253	212.1707	1.1308
DiSMEC	34.06	27.96	0.1039	31.4358	55.4037	0.0802
XLR	0.09	0.10	0.4816	5.5430	64.1134	**0.0678**
XReg	35.66	28.51	0.0439	0.4570	7.4715	0.7871
XReg-t	**36.32**	28.45	0.0587	0.3669	8.4376	1.3512
			DSA-1M			
Parabel	37.95	30.93	0.1004	**9.2800**	**2.5031**	**5.6774**
XReg	37.57	31.09	**0.0563**	20.7463	3.1792	11.0178
XReg-t	**38.81**	**31.41**	0.0714	15.4201	3.4036	18.7434

Table 2. XReg significantly improves query coverage over the existing ensemble for DSA on Bing. *Note*: Cov: Query Coverage, CY: Click Yield, IY: Impression Yield, BR: Bounce Rate

Method	Relative Cov (%)	Relative CY (%)	Relative IY (%)	Relative CTR (%)	Relative BR (%)
Pointwise XReg	-	105	105	100	100
Labelwise XReg	**127**	**148**	**150**	98	100

not regressors but are good generative classifiers which and therefore increase regression errors. Since the tail classifiers are extremely efficient to train and the re-ranking step is only applied to a small number (100s) of labels with high relevance estimates from XReg, XReg-t can be very efficient with 1.1, 1.96 and 2.8 times the training time, prediction time and model size as XReg in worst case.

Prabhu et al. (2020) further contains 1) Additional results for WP@k, Tau@k where k=1,3, nDCG@5 and XRMSE@5 2) Extensive experimentation on filtering and re-ranking on the prediction relevance weight estimates from XReg, 3) Analysis of ranking errors and regression metrics can be found in Prabhu et al. (2020) and 4) Further ablation studies showing the effectiveness of XReg and the novel labelwise inference algorithms.

DSA Results: Table 1 shows the offline evaluation on DSA-130K and DSA-1M while Table 2 showcases the results of the live deployment of labelwise XReg in Bing DSA pipeline. Even though few of the extreme classification techniques could scale to DSA-130K, the live deployment requires the techniques to scale to tens of millions of labels (queries) and data points (ads). In the actual deployment only PfastreXML, Parabel, and XReg were able to scale.

Table 2 compares XReg's performance to the existing DSA ensemble, consisting of BM25 information retrieval based algorithm (Jones et al., 2000) and PfastreXML when deployed on Bing. Both pointwise and labelwise XReg were deployed and evaluated. Pointwise XReg increased RPM, CY, and IY by 5% while maintaining the CTR and BR. Finally, the labelwise XReg boosts the revenue by 58%, improves query coverage by 27% along with a 48% and 50% increase in click yield and impression yields at a cost of only 2% reduction in CTR.

References

Babbar, R. and Schölkopf, B. Dismec: Distributed sparse machines for extreme multi-label classification. In *WSDM*, 2017.

Babbar, R. and Schölkopf, B. Adversarial extreme multi-label classification. *arXiv preprint arXiv:1803.01570*, 2018.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. LIBLINEAR: A library for large linear classification. *JMLR*, 2008.

Jones, K. S., Walker, S., and Robertson, S. E. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 2000.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86, 1951.

Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., and Varma, M. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *WWW*, April 2018.

Prabhu, Y., Kusupati, A., Gupta, N., and Varma, M. Extreme regression for dynamic search advertising. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, February 2020.