Supplementary: Generalized Zero-Shot Extreme Multi-label Learning

ACM Reference Format:
. 2021. Supplementary: Generalized Zero-Shot Extreme Multi-label Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3447548.3467426

1 APPENDIX

THEOREM 1.1. *Given the feature independence assumption, the value of* $\mathbf{w}_1 = -\mathbf{H}_0^{-1}\mathbf{g}_0$ *takes the following form where* ϵ, δ *are small constants:*

$$w_{1k} = \frac{4e_k^+}{e_k^+ + e_k^-} \tag{1}$$

where $e_k^+ = \frac{\sum_j \frac{1+b_j}{2} e_{jk}}{NL}$, $e_k^- = \frac{\sum_j \frac{1-b_j}{2} e_{jk}}{NL}$

PROOF. The objective we wish to minimize is replicated here:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \lambda \sum_{j=1}^{NL} \log(1 + e^{-b_j \mathbf{w}^\top \mathbf{a}_j}) \tag{2}$$

$$\|\mathbf{w}_{k:k+D}\|_0 \leq K \; \forall K \in \{1, \cdots, C\}$$

$$\mathbf{b} \in \{-1, 1\}^{NL}, \mathbf{a}_i \in \mathbb{R}^{CD}, \mathbf{w} \in \mathbb{R}^{CD}$$

where $\mathbf{w}, \mathbf{b}$ are flattened $\mathbf{W}, \mathbf{Y}$; $\mathbf{a}_j$ is flattened form of its corresponding outer product matrix $\mathbf{x}_i \mathbf{z}_l^\top$; $\mathbf{w}_{k:k+D}$ is a sub-vector of $\mathbf{w}$.

At any general $\mathbf{w}$, the gradient and hessian for (2) take the following forms, where $\mathbf{A}$ is the feature matrix with column j being $\mathbf{a}_j$ and $\mathbf{D}$ is a diagonal matrix with $D_{kk} = \frac{1}{1+e^{-b_j \mathbf{w}^\top \mathbf{a}_j}}$

$$\mathbf{g} = \lambda \mathbf{A}\mathbf{D}\mathbf{b} \tag{3}$$

$$\mathbf{H} = \mathbf{I} + \lambda \mathbf{A}\mathbf{D}(\mathbf{I} - \mathbf{D})\mathbf{A}^\top \tag{4}$$

However, at $\mathbf{w}_0 = \mathbf{0}$, the above expressions can be simplified as:

$$\mathbf{g}_0 = \frac{\lambda}{2}\mathbf{A}\mathbf{b} \tag{5}$$

$$\mathbf{H}_0 = \mathbf{I} + \frac{\lambda}{4}\mathbf{A}\mathbf{A}^\top \tag{6}$$

To further simplify the hessian computation, we assume that the features are generated from independent probability distributions with the expectations $\mathbb{E}[a_{jk}] = e_k = \frac{\sum_j a_{jk}}{NL}$. Then, the gradient and average hessian turn out to be

$$\mathbf{g}_0 = \frac{\lambda NL}{2}(\mathbf{e}^- - \mathbf{e}^+) \tag{7}$$

$$\mathbf{H}_0 = I + \frac{\lambda NL}{4}(\mathbf{E}(\mathbf{I} - \mathbf{E}) + \mathbf{e}\mathbf{e}^\top) \tag{8}$$

where, $e_k^+ = \frac{\sum_j \frac{1+b_j}{2} e_{jk}}{NL}$, $e_k^- = \frac{\sum_j \frac{1-b_j}{2} e_{jk}}{NL}$, $\mathbf{e}^+ + \mathbf{e}^- = \mathbf{e}$, and $\mathbf{E}$ is a diagonal matrix with $E_{kk} = e_k$.

Now, with $\mathbf{F}$ a diagonal matrix with $F_{kk} = \frac{4I}{\lambda NL} + E_{kk}(1 - E_{kk})$, the next iterate can be simplified as

$$\mathbf{w}_1 = -\mathbf{H}_0^{-1}\mathbf{g}_0 \tag{9}$$

$$= 2\Big(\mathbf{F} + \mathbf{e}\mathbf{e}^\top\Big)^{-1}(\mathbf{e}^+ - \mathbf{e}^-) \tag{10}$$

$$= 2\mathbf{F}^{-1}(\mathbf{e}^+ - \mathbf{e}^-) - 2\frac{\mathbf{F}^{-1}\mathbf{e}\mathbf{e}^\top\mathbf{F}^{-1}(\mathbf{e}^+ - \mathbf{e}^-)}{1 + \mathbf{e}^\top\mathbf{F}^{-1}\mathbf{e}} \tag{11}$$

where the last step is based on Sherman-Morrison Lemma.

Now, we make another simplifying assumption that the features which occur in very few or many points are uninformative and are hence filtered off. Consequently, $\frac{1}{NL} \ll e_k \leq 1$, thus leading to

$$\mathbf{w}_1 \approx 2\Big(\mathbf{E}^{-1}(\mathbf{e}^+ - \mathbf{e}^-) - \frac{\mathbf{1}^\top(\mathbf{e}^+ - \mathbf{e}^-)}{1 + \mathbf{1}^\top\mathbf{e}}\mathbf{1}\Big) \tag{12}$$

$$\approx 2\Big(\mathbf{E}^{-1}(\mathbf{e}^+ - \mathbf{e}^-) + \mathbf{1}\Big) \tag{13}$$

where last step observes that $\sum_k e_k^+ \ll \sum_k e_k^-$ since the count of positive and negative instances are respectively $N \log L$ and NL. Consequently, $w_{1k} = \frac{4e_k^+}{e_k^+ + e_k^-}$. □

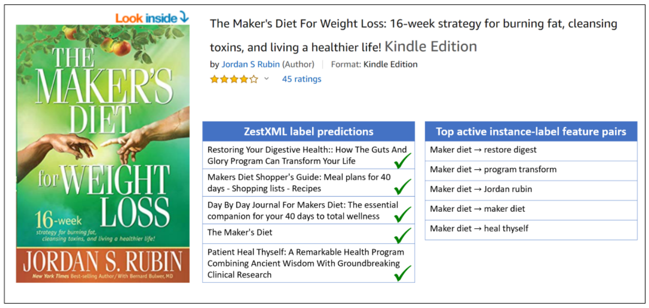
THEOREM 1.2. *Let* $\mathbf{x}$ *be a test point,* σ_x, σ_z *be the bounds over* $L1$ *norms of* $\mathbf{x}, \mathbf{z}$ *respectively and* ϵ *be a small error tolerance parameter. Further, let* $s^* = \max_l \mathbf{x}^\top \mathbf{W}_a \mathbf{z}_l$ *be the score of the top-ranked label by approximate prediction. Then, an efficient algorithm exists which instead uses* $\tilde{\mathbf{W}}_a$ *obtained by truncating parameters smaller than* ϵ *and predicts, in time* $O(\frac{\hat{C}\hat{D}K \log L}{\epsilon})$*, a top-ranked label with score* $\tilde{s}$ *whose regret bounded by* $s^* - \tilde{s} \leq \sigma_x \sigma_z \epsilon$ *.*

PROOF. Let $\mathbf{x} \in \mathbb{R}^C, \mathbf{z} \in \mathbb{R}^D$ be a test point and a label respectively. The objective of this theorem is to efficiently compute $\mathbf{x}^\top \mathbf{W}_a \mathbf{z}$ in an approximate manner. To achieve this, we begin by first projecting $\mathbf{x}$ into the label feature space as $\hat{\mathbf{x}} = W^\top \mathbf{x} \in \mathbb{R}^D$. Let's make the standard assumption that both point feature vectors and label feature vectors are highly sparse with maximum sparsity $\hat{C}, \hat{D}$ respectively. In such a case, the cost of projecting the test point is $\hat{C}K$ where K is the sparsity in W.

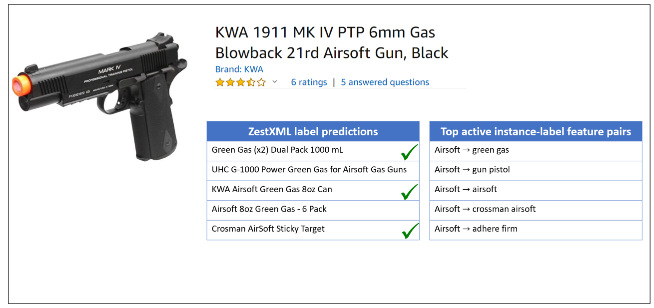
Now, prediction involves outputting the labels $l \in \{1, \cdots, L\}$ with maximum $\mathbf{x}^\top W \mathbf{z}_l = \hat{\mathbf{x}}^\top \mathbf{z}_l$ score. A naive way for this is to iterate through each feature of $\mathbf{z}_l$ for every label l to compute $\hat{\mathbf{x}}^\top \mathbf{z}_l$ with total complexity $\hat{C}K + \hat{D}L$ which is huge since L can

KDD '21, August 14–18, 2021, Virtual Event, Singapore

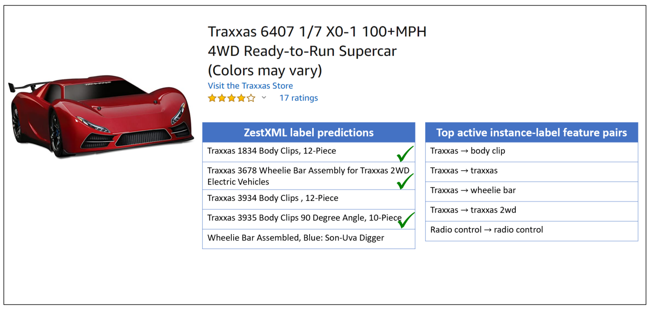
ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467426



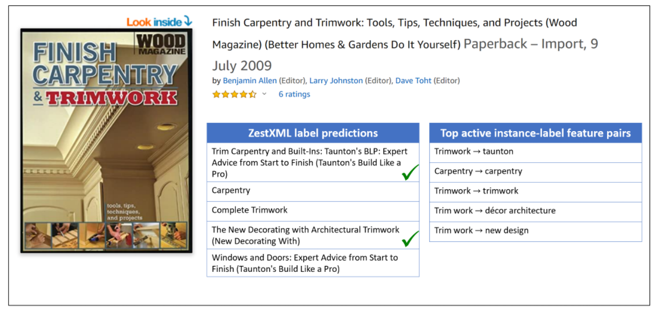
(a) Amazon: The Maker's Diet For Weight Loss: 16-week strategy for burning fat, cleansing toxins, and living a healthier life



(b) Amazon: KWA M1911 MKIV PTP Blowback Airsoft Pistol airsoft gun



(c) Amazon: Traxxas 6407 1/7 X0-1 100+MPH 4WD Ready-to-Run Supercar



(d) Amazon: Finish Carpentry and Trimwork: Tools, Tips, Techniques, and Projects (Wood Magazine)

Figure 1: 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.

Table 1: Comparison of ZestXML with other ZSL and XML algorithms

Algorithm	G.ZSL P@1	G.ZSL P@3	G.ZSL P@5	Algorithm	G.ZSL P@1	G.ZSL P@3	G.ZSL P@5	Algorithm	G.ZSL P@1	G.ZSL P@3	G.ZSL P@5
EURLex-4.3K				Amazon-1M				Wikipedia-1M			
ZestXML-tuned	91.42	82.36	69.5	ZestXML-tuned	**24.13**	**15.02**	**10.78**	ZestXML-tuned	30.63	22.20	17.22
ZestXML-OMP	71.50	57.61	47.85	ZestXML-OMP	20.27	12.88	9.47	ZestXML-OMP	13.25	8.76	7.20
AttentionXML	**93.61**	**83.42**	**69.99**	AttentionXML	19.07	10.98	7.45	AttentionXML	**34.11**	**24.72**	**18.98**
Astec	91.12	80.12	66.26	Astec	18.43	10.81	7.46	Astec	20.70	13.60	10.31
Decaf	81.35	68.61	56.6	Decaf	20.01	11.72	8.08	Decaf	28.27	19.75	15.24
Parabel	90.81	81.35	68.64	Parabel	17.78	10.21	6.89	Parabel	28.07	19.4	14.56
DiSMEC	91.46	82.32	69.31	DiSMEC	19.34	11.23	7.7	DiSMEC	24.10	17.59	13.75
Bonsai	91.53	82.08	69.13	Bonsai	18.99	10.98	7.47	Bonsai	29.15	20.49	15.65
XReg	86.83	78.08	66.97	XReg	17.36	10.5	7.24	XReg	24.69	17.69	13.92
PfastreXML	83.05	72.21	61.14	PfastreXML	14.46	8.7	6	PfastreXML	23.55	15.34	11.47
FastText ANNS	31.82	21.29	17.27	FastText ANNS	13.92	7.83	5.45	FastText ANNS	8.59	4.95	3.68
Bert ANNS	10.50	5.89	4.40	Bert ANNS	19.27	11.42	8.16	Bert ANNS	11.03	5.91	4.25
Topic Model	14.43	9.32	7.19	Topic Model	2.04	1.85	1.70	Topic Model	2.30	1.60	1.30

be in millions. This calls for a faster but approximate approach to prediction.

Let $\tilde{\mathbf{W}}_a$ be a sparsified $\mathbf{W}_a$ after settings its parameters which are smaller than ϵ to 0. Now it is easy to see that, $\mathbf{x}^\top \mathbf{W}_a \mathbf{z} - \mathbf{x}^\top \tilde{\mathbf{W}}_a \mathbf{z} \leq \sigma_x \sigma_z \epsilon$. The projection $\mathbf{x}^\top \tilde{\mathbf{W}}_a$ costs at most $O(\hat{C}K)$ non-zeros. Further, due to the form of $\mathbf{w}_1$ in Theorem 1.1, each non-zero maps onto at most $\frac{\log L}{\epsilon}$ labels. Therefore, the total time complexity is

Table 2: Comparison of ZestXML with other ZSL and XML algorithms on proprietary Bing Ads-31M dataset

Algorithm	G.ZSL		
	P@1	**P@3**	**P@5**
Ads-31M			
ZestXML-tuned	**15.45**	**9.70**	**7.12**
ZestXML-XOMP	10.23	7.71	6.14
Parabel	3.66	2.40	1.83
Xreg	3.13	2.04	1.55
FastText ANNS	4.75	3.28	2.61
Bert ANNS	**6.75**	**4.58**	**3.62**
Topic Model	1.33	1.00	0.87

Algorithm 1 Extreme Hard Thresholding Pursuit

input:
Training point feature matrix $\mathbf{X} \in \mathbb{R}^C \times \mathbb{R}^N$
Label feature matrix $\mathbf{Z} \in \mathbb{R}^D \times \mathbb{R}^L$
Ground truth relevance matrix $\mathbf{Y} \in \mathbb{R}^L \times \mathbb{R}^N$
Model sparsity $K \in \mathbb{N}$
output:
Sparsified parameter matrix $\mathbf{W}_a \in \mathbb{R}^C \times \mathbb{R}^D$
procedure Extreme Hard Thresholding Pursuit
$\mathbf{sumX} \leftarrow \text{row\_sum}(\mathbf{X})$ # $O(N\hat{C})$ sparse sum of each column
$\mathbf{sumZ} \leftarrow \text{row\_sum}(\mathbf{Z})$ # $O(L\hat{D})$ sparse sum of each column
sumX, **sumZ** are column vectors
$\mathbf{Xt} \leftarrow \mathbf{X}^\top$
$\mathbf{ZtY} \leftarrow \mathbf{Z}^\top * \mathbf{Y}$ # $O(N\hat{D}\log L)$ matrix product
for $c \in \{1, \cdots, C\}$ **do**
$\mathbf{n} \leftarrow \mathbf{ZtY} * \mathbf{Xt}[:, c]$
$\mathbf{d} \leftarrow sumX[c] * \mathbf{sumZ}$
$\mathbf{p} \leftarrow \mathbf{n}./\mathbf{d}$ # elementwise division
$\mathbf{p} \leftarrow \text{truncate}(\mathbf{p}, K)$ # retain only highest K values in $\mathbf{p}$
$\mathbf{W}_a[:, c] = \mathbf{p}$

Algorithm 2 ZestXML Label Shortlister

input:
Test point $\mathbf{x} \in \mathbb{R}^C$
Label feature matrix $\mathbf{Z} \in \mathbb{R}^D \times \mathbb{R}^L$
Approximate parameter matrix $\mathbf{W}_a \in \mathbb{R}^C \times \mathbb{R}^D$
Error tolerance $\epsilon \in \mathbb{R}$
output:
Relevance scores $\tilde{\mathbf{s}} \in \mathbb{R}^L$
procedure ZestXML Label Shortlister
$\tilde{\mathbf{W}}_a \leftarrow threshold(\mathbf{W}_a, \epsilon)$ # retain only those values $\geq \epsilon$
$\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{W}}_a^\top \mathbf{x}$
$\tilde{\mathbf{s}} \leftarrow \mathbf{Z}\hat{\mathbf{x}}$ # labels with +ve scores are shortlisted

bounded by $O(\frac{\hat{C}\hat{D}K\log L}{\epsilon})$ which includes the cost of projecting the test point and then iterating over labels indexed against each non-zero projection feature. □