A Submodular Feature-Aware Framework for Label Subset Selection in Extreme Classification Problems

Elham J. Barezi[1,2,3], Ian D. Wood [1], Pascale Fung [1,2,4], Hamid R. Rabiee [3]
[1]Center for Artificial Intelligence Research (CAiRE)
[2]Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[3]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[4]EMOS Technologies Inc. HK Limited
ejs@cse.ust.hk, pascale@ust.hk, rabiee@sharif.edu

Abstract

Extreme classification is a classification task on an extremely large number of labels (tags). User generated labels for any type of online data can be sparing per individual user but intractably large among all users. It would be useful to automatically select a smaller, standard set of labels to represent the whole label set. We can then solve efficiently the problem of multi-label learning with an intractably large number of interdependent labels, such as automatic tagging of Wikipedia pages. We propose a submodular maximization framework with linear cost to find informative labels which are most relevant to other labels yet least redundant with each other. A simple prediction model can then be trained on this label subset. Our framework includes both label-label and label-feature dependencies, which aims to find the labels with the most representation and prediction ability. In addition, to avoid information loss, we extract and predict outlier labels with weak dependency on other labels. We apply our model to four standard natural language data sets including Bibsonomy entries with users assigned tags, web pages with user assigned tags, legal texts with EUROVOC descriptors(A topic hierarchy with almost 4000 categories regarding different aspects of European law) and Wikipedia pages with tags from social bookmarking as well as news videos for automated label detection from a lexicon of semantic concepts. Experimental results show that our proposed approach improves label prediction quality, in terms of precision and nDCG, by 3% to 5% in three of the 5 tasks and is competitive in the others, even with a simple linear prediction model. An ablation study shows how different data sets benefit from different aspects of our model, with all aspects contributing substantially to at least one data set.

1 Introduction

Multi-label learning has recently attracted attention in the research community due to an increase in applications such as semantic labeling of images and videos, bio-informatics, genetic functions, and music categorization. In addition, multi-label learning can address machine learning problems in web data mining, including recommender systems, multimedia sharing websites, and ranking (Zhang and Zhang, 2010).

An important application of extreme multi-label learning is automatic tagging and social tagging of large information collections such as Wikipedia or the Web. A user can add their own keywords to a text, as if they were the keywords they would use to look for the article in a search engine. Since tags use an open vocabulary, the number of tags is increasing continually in order to adjust to the needs of new information. Moreover, different users can assign different tags to the same resource, resulting in a great diversity of tags for that resource.

The biggest challenge of extreme multi-label learning is the dimension of the output space. As the number of output labels increases, the number of output states increases exponentially. In order to overcome this exponential growth, it is necessary to use label dependencies to simplify the problem (Zhang and Zhang, 2010; Tsoumakas et al., 2010).

We propose a submodular maximization approach with a linear cost to find an informative set of labels. In contrast to the other similar approaches (Balasubramanian and Lebanon, 2012; Bi and Kwok, 2013) which consider only label-label dependencies, we also consider label-feature dependencies and outlier labels that are highly independent of other labels. Solving the problem using the selected (smaller number of) labels leads to minimizing both representation and training error.

Proceedings of NAACL-HLT 2019, pages 1009–1018
Minneapolis, Minnesota, June 2 - June 7, 2019.

Representation ability is equivalent to the power of the selected subset to reconstruct the remaining labels, and prediction ability is equivalent to training accuracy leading to less error propagation from predicted label subset to the remaining labels during reconstruction.

Submodular maximization approaches have proved very effective in many applications, such as finding the most influential nodes in social networks to maximize the spread of information (for applications such as advertising and marketing (Kempe et al., 2003; Ohsaka et al., 2014)) and video and image collection summarization (Gygli et al., 2015; Tschiatschek et al., 2014). There are many effective algorithms such as (Mirzasoleiman et al., 2015) to make submodular optimization approaches much faster or do them in a distributed way (Mirrokni and Zadimoghaddam, 2015) to perform faster parallel processing for very large scale datasets.

2 Related Work

Many of the early proposed multi-label learning approaches struggle with large-scale applications, as they learn each label separately or investigate the label dependencies in a way that leads to a costly and complicated model (Tsoumakas et al., 2010).

The other research trends is to transform the label space to a smaller space and map back the predicted results in the compressed space to the original space. Hsu et al. (2009) presented the first approach targeting label space compression based on compressed sensing, which assumes sparsity of the label space. An expensive optimisation problem has to be solved in the prediction step. Tai and Lin (2012); Chen and Lin (2012); Yu et al. (2014), and (Lin et al., 2014) used orthogonal projections and low-rank assumptions to extract a label matrix decomposition and find a low-dimensional embedding space. In (Bhatia et al., 2015b), the authors perform local embedding of the label vectors. To achieve stronger locality, they cluster the data into smaller regions, which is unstable and costly for high-dimensional spaces and one needs an ensemble of the learners to overcome this instability and achieve a good prediction accuracy.

Although the previously proposed approaches make the embedding space smaller and more tractable, they may lead to loss of information as a result of transforming the label space to lower-dimensional spaces. Many of these approaches rely on low-rank assumptions which transform the sparse label space to a new dense embedding space resulting in even lower accuracy, with a higher prediction cost in the new complicated space (Bhatia et al., 2015a).

Balasubramanian and Lebanon (2012) and Bi and Kwok (2013) proposed to select a subset of the labels, and solve the problem in the original label space, based on structure sparsity optimization and SVD decomposition, correspondingly. However, these methods are not tractable for large scale data and not compatible for the real application data. In addition, they have ignored the training error in the label selection step which can lead to selection of the labels that are hard to predict resulting in training error propagation through the next steps.

Another recent thread of research includes the methods that partition the data into smaller groups: In the framework proposed by Barezi et al. (2017), the label space is divided into smaller independent groups, while Agrawal et al. (2013); Prabhu and Varma (2014); Prabhu et al. (2018) propose tree-based methods which partition the data into tree-structured hierarchical groups. These partitioning-based approaches avoid information loss from dimension reduction. However, finding a partitioning tree is a very complicated and time-consuming problem and these approaches require solving a complicated optimization problem to perform partitioning at each node, which is expensive and needs many training samples. In addition, the tree-based approaches suffer from error propagation through the hierarchy and need many training samples to avoid under-fitting in the lower levels of the partitioning tree (Liu et al., 2005).

Instead of making the structural assumption on the relation between the labels, Yen et al. (2016) assume the label space is highly sparse and has a strong correlation with the feature space. They ignore the label space correlation information. Yen et al. (2017) proposed the parallel version of (Yen et al., 2016).

3 Methodology

In this paper, we propose a landmark selection framework for selecting the most informative labels and to solve the multi-label learning problem with these labels. As an example, consider predicting the commercial impact of a new event on

some global organizations (equivalent to the labels in our problem) given a history of the impact of previous events (equivalent to the features and training data in our problem). Instead of predicting the impact on each organization individually, we predict only the impact on a small number of organizations which are both easier to predict and analyze according to available data as well as being more indicative of the economy and the other organizations. Being indicative means that if we know the impact of the new event on these organizations, it can help us to predict the reaction of the other organizations. More formally, we optimize the above set function $f(S)$ in Equation 1.

The proposed method includes both label-label and label-feature dependencies in order to minimize both representation and training error. Previous similar methods ignore label-feature dependencies in the subset selection step, allowing the training error for the selected subset of the labels to be propagated to the reconstructed labels and affecting the final predictions. In addition, to avoid information loss, we also extract and predict outlier labels with weak dependency on other labels and treat them separately.

Our construction results in a monotone submodular function of label sets allowing us to use a maximization framework that benefits from a good theoretical bound by a fast greedy approach with linear cost (Nemhauser et al., 1978). We use a method based on Alternating Direction Method of Multipliers (ADMM) (Boyd, 2011) optimization to learn a linear mapping back to original label space. Therefore, during training, we can select and learn the most informative label subset using a submodular maximization framework of linear cost. During the prediction time, we can use the selected subset to represent the remaining labels using a linear equation with a linear cost in number of the labels.

3.1 Overview of the Submodular Maximization Theorem

Submodular functions have a natural diminishing property which makes them suitable for many applications. A submodular function is a set function with the property that as the size of the selected subset increases, the incremental value of the function by adding a new element to the selected subset does not increase.

The formal definition of a submodular function is as follows:

Definition 1. *For a set function $f(S) : 2^V \rightarrow R$ defined for a finite ground set $V = 1, 2, ...n$, the marginal gain of adding each new member can be computed as $\Delta_f(e|S) = f(e \cup S) - f(S)$. The function $f(.)$ is submodular, if for each $A \subseteq B \subseteq V$, $e \in V \backslash A \cap V \backslash B$, then $\Delta_f(e|A) \geq \Delta_f(e|B)$. Equivalently, the function $f(S) : 2^V \rightarrow R$ is submodular if for any two sets A, $B \in V$, $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$.*

Monotony of sunmodular functions is a useful property which means that the value of the function would not decrease by adding each new member to the input set, and can be defined as following.

Definition 2. *A submodular function $f(.)$ is monotone (non-decreasing) if for every $T \subseteq S$, we have that $f(T) \leq f(S)$.*

A simple example of a submodular function is the setup cost in a factory. Suppose that a factory is capable of making any one of a large finite set V of products. In order to produce product $e \in V$, it is necessary to set up the machines needed to manufacture e, and this costs money. The setup cost is non-linear, and it depends on which other products you choose to produce. For example, if you are already producing iPhones, then the setup cost for also producing iPads is small, but if you are not producing iPhones, the setup cost for producing iPads is large.

We can find a good approximation of the optimum answer for a monotone submodular maximization problems by using the greedy approach and considering the selected subset size constraint. More formally:

Theorem 3. *(Nemhauser et al., 1978) For a non-negative, monotone submodular function f, let S be a set of size k obtained by the greedy strategy similar to Algorithm 1. Then, $f(S) \geq (1 - 1/e) f(S^*)$, where S^* is the optimum solution, and e is Euler's constant approximately equal to* 2.71828

3.2 Submodularity for Label Subset Selection

We propose two submodular functions, aiming to select the most informative subset of the labels. The first function is a penalized version of the graph cut function. It scores label sets with correlation to the other labels and penalizes their similarity to the previously selected labels (f_{pen} in

$$
\begin{aligned}
f(S) &= (How\ members\ of\ set\ S\ are\ individually\ predictable) + \\
&\quad (How\ members\ of\ set\ S\ can\ represent\ the\ members\ not\ included\ in\ S) \\
&= (Prediction\ ability) + (Representation\ ability) \\
&= (Label - Feature\ dependency) + (Label - Label\ dependency) \qquad (1)
\end{aligned}
$$

Algorithm 1 $argmax_S\ f(S)\ s.t.\ \|S\| = k$.

Input: $V = 1, 2, ...n$
Initialization: $S = \emptyset$.
Repeat:
1: $a^\star = argmax_{a \in V \setminus S} f(S \cup \{a\}) - f(S)$
2: $S = S \cup \{a^\star\}$
Until $|S| = k$.
Output: S.

Equation 3). The graph is constructed using the labels as nodes and label correlations as weights for the graph edges. The second function scores the predictability of labels with respect to problem input features (f_{score} in Equation 5). Our final function for identifying the optimal subset of labels is a weighted sum of these (Equation 6).

We consider the label correlations as graph weights w. The graph cut function $f_{cut}(.)$ aims to find a subset of the graph nodes (labels) with the highest weights (strongest dependencies) to the remaining nodes (labels). This captures strong correlation of a label set to the other labels and thus its ability to reconstruct the other labels. The penalised version $f_{pen}(.)$ adds one more term to increase the diversity of the selected labels and avoid choosing similar labels.

$$f_{cut}(S) = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j} \qquad (2)$$

$$f_{pen}(S) = f_{cut}(S) - \lambda \sum_{\substack{i,j \in S \\ i \neq j}} w_{i,j}\ ,\ \lambda \geq 0 \qquad (3)$$

Theorem 4. *$f_{cut}(S)$ is a submodular function and it is monotone for non-large values of $|S|$ (Nemhauser et al., 1978).*

Theorem 5. *$f_{pen}(S)$ is a submodular function and it is monotone for non-large values of λ (Lin et al., 2009).*

The proofs for Theorem 4 and 5 is provided in supplementary Section.

It is important also to consider predictability, which is the training error for the selected subset of the labels, in order to avoid the prediction error of labels with high training error being propagated to the whole label space.

As an estimate of predictability we use either a G^2 or χ^2 independence test for the discrete data, and Fishers Z or t test for the continuous data in order to reject or accept the null hypothesis of independence (Tsamardinos and Borboudakis, 2010). Since, this measures include an implicit normalization, the frequency of the classes in training data does not affect the sampling step.

A higher dependency score for each label and the input feature space means a stronger correlation of the label with the feature space and higher predictability. Given label predictability scores f_{ij} for label i and input feature j and D input features, we calculate dependency scores f_i of the i-th label and the input features:

$$f_i = \sum_{j=1}^{D} f_{ij} \qquad (4)$$

Note that $f_i \geq 0$. We then define the following set function, which also is monotone and submodular (Theorem 6):

$$f_{score}(S) = \sum_{i \in S} f_i \qquad (5)$$

Theorem 6. *$f_{score}(S)$ is a submodular monotone function.*

Proof. For w_i = the sum over the dependency scores of the i-th label and the feature space, $f(S) = \sum_{i \in S} w_i$ is a linear function with $w_i \geq 0$. Any linear function of the form $f(S) = \sum_{i \in S} w_i$ is a submodular function. If $S \subset R$, $\Delta_f(k|S) - \Delta_f(k|R) = 0 \Rightarrow \Delta_f(k|S) \geq \Delta_f(k|R)$. Additionally, if $\forall i\, w_i \geq 0$, then f is monotone, because $f(S \cup k) - f(S) = w_k,\ w_k \geq 0$. $max_{|S|=k} f(S) = max \sum_{i \in S} w_i$. Therefore $f(S)$ is a monotone submodular function. □

Since, any sum of submodular functions with positive coefficients is a submodular function, we can combine $f_{pen}(.)$, and $f_{score}(.)$ by positive

weights, which results in a new submodular function that includes both representation ability and prediction ability of the selected labels. We choose a model parameter $\gamma > 0$ giving us our final submodular function:

$$f(S) = f_{pen}(S) + \gamma . f_{score}(S) \quad (6)$$

3.3 Landmark Information Propagation

The main step of our proposed framework is to propagate the predicted value for the selected label subset to the full set of labels in order to recover the original space. Therefore, we aim to find a linear relation including the dependency of the selected labels and all the other labels. In the prediction step, this linear function obtains the full label set by combining the subset (Y_s) and outlier predictions (E) predicted by the regression functions discussed in next section 3.4. Given 1-hot representations Y_s over the reduced set of labels and Y the full set of labels, we seek matrices Z and E that recover the original labels:

$$Y = Y_s Z + E \quad (7)$$

To find optimal Z and E, we solve the optimization problem Equation 8, where Y and Y_s are matrices populated with our training data. Note that $\|E\|_{2,1}$ is the L_1 norm of the L_2 norms of the columns of E.

$$\begin{aligned} \underset{Z,E}{\text{argmin}} \quad & (\|Z\|_1 + \alpha \|E\|_{2,1}) \\ \text{s.t.} \quad & Y = Y_s Z + E \end{aligned} \quad (8)$$

The sparse matrix Z is a $k \times L$ matrix which includes a few representative labels (due to the sparsity constraint $\|Z\|_1$) for each label ($Y = Y_k Z$). The Z matrix includes the dependency information and performs propagation of the predicted label subset to the full label set, while nonzero columns of matrix E show the outlier and tail labels set O, which cannot be computed perfectly through their relation to the other labels. The index set of the nonzero columns of matrix E indicates the outlier labels. α is a model parameter.

The alternating direction method of multipliers (ADMM) method (Boyd, 2011; Nesterov, 2004; Beck and Teboulle, 2009) provides an efficient algorithm for solving this problem, achieving a convergence rate of $O(1/T^2)$ (where T is the number of iterations). ADMM solves the problem with more than one unknown variable, (Z and E in our case), by alternating between optimizing each variable using augmented Lagrangian. Please see the supplementary materials for more detail on the ADMM method and how it is applied in this case.

3.4 Prediction and Mapping Back to the Original Label Space

We now train a linear classifier to predict labels in the reduced label set $S \cup O$ and map back to the full label set. Given features of the training data X, corresponding labels from selected and outlier labels Y_S and Y_O, we learn linear regression parameters $w_s,\ b_s$ for the selected labels and $w_e,\ b_e$ for the outlier labels:

$$\begin{aligned} \underset{w_s,b_s}{\text{argmin}} \quad & \|Y_s - (X * w_s + b_s)\| + \tfrac{\lambda_1}{2}\|w_s\|^2 \\ \underset{w_e,b_e}{\text{argmin}} \quad & \|Y_o - (X * w_e + b_e)\| + \tfrac{\lambda_2}{2}\|w_e\|^2 \end{aligned} \quad (9)$$

Since all these training tasks are independent of each other, this step is highly parallelizable. The final values for the labels are computed by propagating the selected label subset through the linear relation 7:

$$\begin{aligned} \hat{Y}_s &= X * w_s + b_s \\ \hat{E} &= X * w_e + b_e \end{aligned} \quad (10)$$

$$\hat{Y} = \hat{Y}_s Z + \hat{E} \quad (11)$$

An overview of steps for training and prediction are shown in Algorithms 2 and 3.

Algorithm 2 Training Algorithm.

Input: Training Data X and Y.

1: Find the best label subset by submodular optimization over function 6;
2: Find the linear propagation equation through ADMM optimization over problem 8.
3: Find the linear regression models over small subset of labels and outliers by Equation 9

Output: Label subset, outliers, propagation and regression models.

4 Experiments

4.1 Datasets

We used six different datasets in the experiments. The "Bibtex" dataset is a text dataset extracted from the BibSonomy website (Katakis et al., 2008)

Algorithm 3 Prediction Algorithm.

Input: prediction samples X.

1: Predict candidate label subset and outlier labels using regression model 10.
2: Use 11 to produce full set of labels from candidate subset and outlier labels.

Output: Full label set for input X.

contains metadata for the bibtex items like the title of the paper, the authors, etc and extracts the features according to the term frequency. The "Mediamill" dataset is extracted from the Mediamill contest datasets, which include low-level multimedia features (visual and textual features) extracted from 85 hours of international news videos from the TRECVID 2005/2006 benchmark datasets (Snoek et al., 2006) labeled using a lexicon of 101 semantic concepts, like commercials, nature, and baseball.

The "Eurlex" dataset includes 19,348 legal documents from European nations, containing several different types of documents, including treaties, legislation, case-law and legislative proposals, classified according to the EUROVOC descriptor using 3993 different classes, and 5000 features extracted using common TF-IDF term weighting (Mencia and Fürnkranz, 2008). The "Delicious" dataset is a text dataset extracted from the `del.icio.us` social bookmarking site on the 1st of April 2007 and contains textual data of web pages along with their user defined tags (Tsoumakas et al., 2008). The content of web pages was represented using the Boolean bag-of-words model. "Wiki10-31K" is a collection of social tags for given Wikipedia pages with TF-IDF features (Zubiaga, 2012). The statistics of these datasets are provided in Table 1.

4.2 Experimental Setup

For the small datasets, "Bibtex", "Mediamill", "Delicious", and "Eurlex", the reported results are the average of 10 different experiments for random partitions of each dataset. For the larger dataset, "Wiki10-31K", we did one experiment with the training and testing partition reported in Table 1.

For all experiments we chose a label subset size of 100, except for Mediamill where we chose 30 since 100 would represent all labels. Model tuning is done in two phases: first we tune α for group sparsity (Equation 8), and γ for weighting of the submodular functions (Equation 6), then we tune for λ_1 and λ_2, the regression parameters for mapping back to the original label set (Equation 9) with α and γ fixed. All parameters were chosen by measuring the precision of 10-fold cross validation and using a grid search over the values $\{0, 10^{-3,\ldots,+3}\}$ for each dataset.

The proposed method was compared with several state-of-the-art methods with diverse approaches. LEML (Yu et al., 2014), CPLST (Chen and Lin, 2012), CS (Hsu et al., 2009) and SLEEC (Bhatia et al., 2015b) which are embedding based approaches with a low-rank or sparse assumption in the label space. ML-CSSP (Bi and Kwok, 2013) which solves the problem in the original label space which ignores the training error in the subset selection step. FastXML (Prabhu and Varma, 2014), and PD-sparse (Yen et al., 2016) which do not use an embedding transformation and aim to solve the problem without using compression or sampling. We have used the reported results, if available, and otherwise tuned the parameters for the baseline algorithms by means of 10-fold cross validation.

5 Results and Discussion

Table 2 shows the average and standard deviation of Precision@k for the four small-scale datasets, "Bibtex", "Mediamill", "Delicious", and "Eurlex", and the large-scale dataset "Wiki10-31k". For "Wiki10-31k", results are reported only for those baselines that were tractable. The results for nDCG@k are included in supplementary Material, Table 5. Since the SLEEC and FastXML methods are ensemble-based, using multiple non-linear models, it is not fair to compare them with the single model methods such as our own. These methods partition the sample space into smaller tractable clusters and obtain separate classifiers for each partition. We compare our method with these in Table 3.

The proposed approach in most cases has significantly better results than other methods on both measures. The embedding based approaches suffer from accumulation of the embedding and training error (Balasubramanian and Lebanon, 2012), however in the proposed approach, we have removed the embedding step and considered the training error minimization at the label subset selection step. On the other hand, the non-embedding approaches such as PD-sparse (Yen et al., 2016) ignore the label space inter-

Dataset	Domain	Number of Features	Number of Labels	Training Points	Testing Points
Bibtex	Text	1836	159	4880	2515
Delicious	Text(Web)	500	983	12920	3185
Mediamill	Video	120	101	30993	12914
Eurlex	Text	5000	3993	17413	1935
Wiki10-31K	Text	101938	30938	14146	6616

Table 1: Dataset statistics

	Proposed	PD-sparse	LEML	CPLST	CS	ML-CSSP
Bibtex						
P@1	**64.56**±0.79	61.29±0.65	62.54±0.52	62.38±0.63	58.87±0.61	44.98±1.15
P@3	**39.51**±0.34	35.82±0.46	38.41±0.42	37.84±0.48	33.53±0.49	30.43±0.59
P@5	**28.80**±0.26	25.74±0.30	28.21±0.24	27.62±0.27	23.72±0.29	23.53±0.37
Delicious						
p@1	*65.13*±0.39	51.82±1.40	**65.67**±0.73	*65.31*±0.88	61.36±0.38	63.04±1.28
P@3	59.07±0.41	44.18±1.04	**60.55**±0.48	59.95±0.43	56.46±0.33	56.26±1.13
P@5	54.52±0.34	38.95±0.94	**56.08**±0.43	55.31±0.50	52.07±0.30	50.16±0.83
Mediamill						
P@1	**84.25**±0.27	81.86±4.08	*84.01*±0.31	83.35±0.33	*83.82*±5.92	78.95±0.23
P@3	*67.29*±0.24	62.52±2.31	*67.20*±0.23	66.18±0.22	**67.32**±4.42	60.93±0.24
P@5	**52.90**±0.15	45.11±1.14	52.80±0.18	51.46±0.20	52.80±2.61	44.27±0.20
Eurlex						
P@1	**81.04**±0.81	76.43±1.04	63.40±1.58	72.28±0.99	58.52±1.06	62.09±2.12
P@3	**67.91**±0.97	60.37±0.74	50.35±1.44	58.16±1.11	45.51±0.71	48.39±1.31
P@5	**56.81**±0.97	49.72±0.74	41.28±1.07	47.73±0.97	32.47±0.58	40.11±1.10
Wiki10-31k						
p@1	**86.05**	82.14	73.47	-	-	-
P@3	**76.85**	69.68	62.43	-	-	-
P@5	**67.77**	58.76	54.35	-	-	-

Table 2: Non-ensemble models with k=100 or 30 (Mediamill). Best in **bold** and not significantly different to best at p=0.05 in *italics*.

	Proposed	SLEEC	FastXML
Bibtex			
P@1	*64.56*±0.79	**65.08**±0.65	63.42±0.67
P@3	*39.51*±0.34	**39.64**±0.39	39.23±0.57
P@5	*28.80*±0.26	**28.87**±0.32	28.86±0.38
Delicious			
P@1	65.13±0.39	67.59±0.53	**69.61**±0.58
P@3	59.07±0.41	61.38±0.59	**64.12**±0.75
P@5	54.52±0.34	56.56±0.54	**59.27**±0.65
Mediamill			
P@1	84.25±0.27	**87.82**±0.33	84.22±0.27
P@3	67.29±0.24	**73.45**±0.30	67.33±0.20
P@5	52.90±0.15	**59.17**±0.34	53.04±0.18
Eurlex			
P@1	**81.04**±0.81	79.26±0.86	71.36±1.63
P@3	**67.91**±0.97	64.30±0.88	59.90±1.58
P@5	**56.81**±0.97	52.33±0.80	50.39±1.40
Wiki10-31k			
p@1	**86.05**	85.88	83.03
P@3	**76.85**	72.98	67.47
P@5	**67.77**	62.70	57.76

Table 3: Ensemble-based nonlinear models. Best in **bold** and not significantly different to best in *italics*.

dependency information which can be useful to improve the prediction accuracy for the labels which are not easy to predict only from input features.

ML-CSSP (Bi and Kwok, 2013) and the work of Balasubramanian and Lebanon (2012) attempt, like us, to find the most informative labels in order to perform label subset selection. However, our approach improves on their results, supporting the idea that considering only the label space information (ignoring label-feature dependency information) in the label selection step can lead to label sets that are not easy to predict whose training error will be propagated through to final model predictions.

The SLEEC and FastXML methods are ensemble-based methods using multiple nonlinear models and can be expected to outperform single model methods such as ours. SLEEC aims to partition the sample space into smaller tractable clusters to obtain a nonlinear embedding and trained model for each partition. FastXML finds a partitioning tree by using nonlinear binary classifiers to partition the samples at each node, which is a very complicated and unstable problem for high-dimensional spaces. Therefore, for both SLEEC and FastXML methods, they need an ensemble of the learners in order to overcome this instability and achieve a good prediction accuracy. Table 3 shows that SLEEC performs best on the Mediamill and FastXML performs best on the Delicious dataset. This shows that finding a representative subset using a linear method is not a consistent assumption for these datasets than the low-rank and tree-based assumptions. However, for Bibtex datasset, our proposed method is competitive with the best results, and for Eurlex and Wiki10-31k, our method is substantially better than both SLEEC and FastXML, a notable achievement for

a single model approach.

5.1 Ablation study

The ablation study results in Table 4 shows how different data sets benefit from different parts of our proposed framework, with all parts contributing substantially to at least one data set. We have reported the results by considering only label-label dependency information (f_{pen}), label-feature dependency information (f_{score}) and combining all 3 parts (f_{pen}, f_{score} and outlier information). The results support the assertions that considering only the label space information (ignoring label-feature dependency information) in the label selection step causes prediction error of labels with high training error to be propagated to the whole label space and that it is important to also select outlier labels that are hard to predict from other selected labels.

	f_{pen}	f_{score}	$f_{pen} + \alpha f_{score}$	**+Outliers**
Bibtex				
P@1	60.98	**63.27**	63.29	**64.55**
P@3	34.86	**37.10**	37.55	**39.51**
P@5	25.94	**26.73**	27.05	**28.78**
Mediamill				
P@1	81.12	81.83	**84.25**	84.25
P@3	64.15	65.92	**67.79**	67.99
P@5	51.26	51.66	**52.70**	52.90
Delicious				
P@1	62.71	62.71	**64.33**	65.14
P@3	56.95	56.95	**58.30**	59.10
P@5	52.63	52.63	**53.58**	**54.55**
Eurlex				
P@1	**56.60**	3.84	56.60	**81.04**
P@3	**37.88**	3.11	37.88	**67.91**
P@5	**29.71**	3.01	29.71	**56.81**
Wiki10-31k				
P@1	**81.86**	54.34	81.86	**86.05**
P@3	**68.51**	40.41	68.51	**76.85**
P@5	**56.77**	33.00	56.77	**67.77**

Table 4: Ablation Study. Bold indicates a difference of $\geq 0.8\%$

We also investigated the effect of changing the subset size S on the final prediction quality (we have ignored the outlier effect in these experiments). Figure 1 shows an initial marked increase in performance with subset size, however the results gets more stable when the subset size gets larger. This observation, which is consistent with the submodular property, provides a clue that using a more complicated training model, like a non-linear model, for a smaller selected set of labels may lead to higher performance than increasing the subset size while using a linear model.

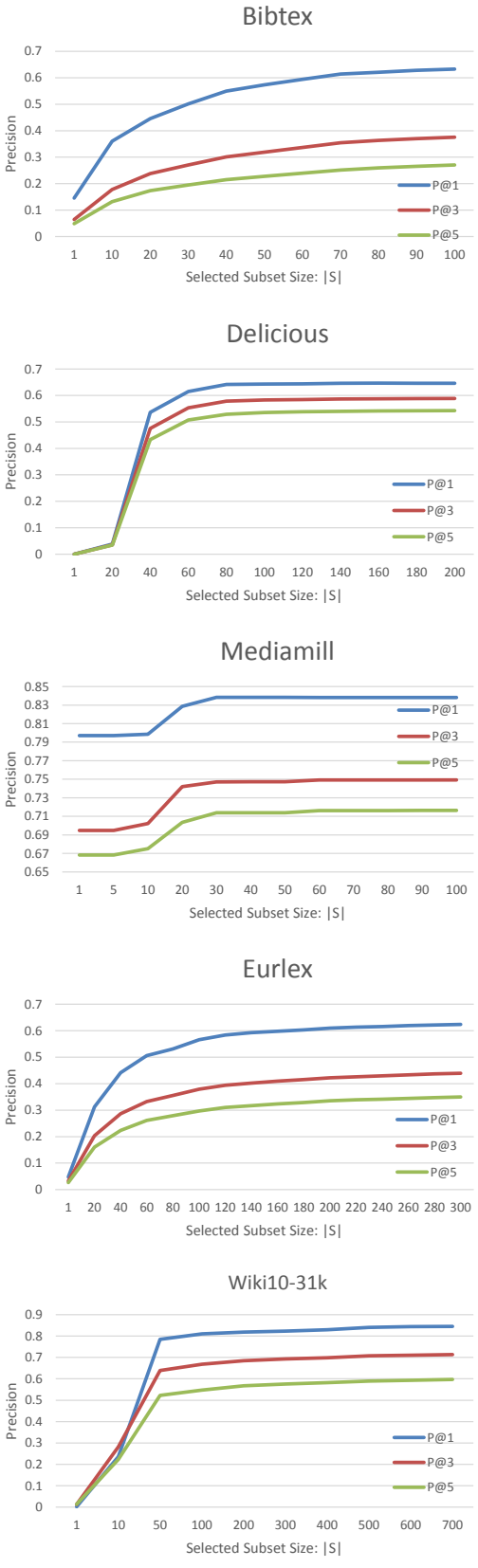

Figure 1: Precision score changes by subset size without outliers.

6 Conclusion and Future Work

We propose a novel approach for extreme multi-label classification that simplifies the problem by selecting an informative and easily modelled subset of labels and subsequently mapping back to the full set of labels. While the method is very well applicable to text datasets, it is applicable as a general ML method for different domains. Our novel label selection mechanism follows three principles: A new submodular maximisation framework that combines label-label dependencies and label training error together with a mechanism to identify outlier labels that are hard to reconstruct. Modelling only the most informative labels helps to avoid transforming the label space to a new embedding space leading to accumulation of training and embedding errors. We use a greedy approach for our monotone submodular framework with linear cost and good theoretical convergence. Extensive experiments using a linear prediction model on selected labels conducted on five standard real-world datasets demonstrate that our method achieves better performance than single model approaches, and better or comparable performance to ensemble based methods. In future, we can improve our model by using nonlinear training model instead of a simple linear regression model for the selected subset of the labels. Moreover, ablation study results suggest that a nonlinear propagation model to reconstruct the full label set may be of benefit.

Acknowledgments

This work was partially funded by grants #16214415 and #16248016 of the Hong Kong Research Grants Council, ITS/319/16FP of Innovation Technology Commission, and RDC 1718050-0 of EMOS.AI.

References

Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of WWW-13*, pages 13–24, Seoul, Korea.

Krishnakumar Balasubramanian and Guy Lebanon. 2012. The landmark selection method for multiple output prediction. In *Proceedings of ICML-12*, pages 983–990, Edinburgh, Scotland.

Elham J Barezi, James T Kwok, and Hamid R Rabiee. 2017. Multi-label learning in the independent label sub-spaces. *Pattern Recognition Letters*, 97:8–12.

Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on Imaging Sciences*, 2(1):183–202.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Prateek Jain, and Manik Varma. 2015a. Locally nonlinear embeddings for extreme multi-label learning. In *NIPS*, Montreal, Canada.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015b. Sparse local embeddings for extreme multi-label classification. In *NIPS*, pages 730–738.

Wei Bi and James Kwok. 2013. Efficient multi-label classification with many labels. In *Proceedings of ICML-13*, pages 405–413, Atlanta.

Stephen Boyd. 2011. Alternating direction method of multipliers. In *Talk at NIPS workshop on optimization and machine learning*.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Yao-Nan Chen and Hsuan-Tien Lin. 2012. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, pages 1529–1537, Harrahs and Harveys.

Michael Gygli, Helmut Grabner, and Luc Van Gool. 2015. Video summarization by learning submodular mixtures of objectives. In *Proceedings CVPR 2015*, pages 3090–3098.

Bingsheng He and Xiaoming Yuan. 2012. On the o(1/n) convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709.

Daniel Hsu, Sham Kakade, John Langford, and Tong Zhang. 2009. Multi-label prediction via compressed sensing. In *NIPS*, volume 22, pages 772–780, Vancouver CANADA.

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, page 75.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM.

Hui Lin, Jeff Bilmes, and Shasha Xie. 2009. Graph-based submodular selection for extractive summarization. In *IEEE Workshop on ASRU*, pages 381–386.

Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. 2014. Multi-label classification via feature-aware implicit label space encoding. In *Proceedings of ICML-14*, pages 325–333, Beijing, China.

Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. 2005. Support vector machines classification with a very large-scale taxonomy. *Acm Sigkdd Explorations Newsletter*, 7(1):36–43.

Eneldo Loza Mencia and Johannes Fürnkranz. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer.

Vahab Mirrokni and Morteza Zadimoghaddam. 2015. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 153–162. ACM.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. 2015. Lazier than lazy greedy. In *AAAI*, pages 1812–1818.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294.

Y Nesterov. 2004. Introductory lectures on convex programming: a basic course, volume i.

Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In *AAAI*, pages 138–144.

Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 993–1002. International World Wide Web Conferences Steering Committee.

Yashoteja Prabhu and Manik Varma. 2014. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 263–272, New York City. ACM.

Cees GM Snoek, Marcel Worring, Jan C Van Gemert, Jan-Mark Geusebroek, and Arnold WM Smeulders. 2006. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *14th Annual ACM International Conference on Multimedia*, pages 421–430. ACM.

Farbound Tai and Hsuan-Tien Lin. 2012. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542.

Ioannis Tsamardinos and Giorgos Borboudakis. 2010. Permutation testing improves bayesian network learning. *Machine Learning and Knowledge Discovery in Databases*, pages 322–337.

Sebastian Tschiatschek, Rishabh K Iyer, Haochen Wei, and Jeff A Bilmes. 2014. Learning mixtures of submodular functions for image collection summarization. In *Advances in neural information processing systems*, pages 1413–1421.

G Tsoumakas, I Katakis, and I Vlahavas. 2008. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, pages 30–44.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer.

Ian EH Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, and Eric Xing. 2017. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 545–553. ACM.

Ian EH Yen, Xiangru Huang, Kai Zhong, Pradeep Ravikumar, and Inderjit S Dhillon. 2016. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *Proceedings of ICML-16*.

Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S Dhillon. 2014. Large-scale multi-label learning with missing labels. In *Proceedings of ICML-14*, volume 32, Beijing, China.

Min-Ling Zhang and Kun Zhang. 2010. Multi-label learning by exploiting label dependency. In *16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 999–1008. ACM.

Arkaitz Zubiaga. 2012. Enhancing navigation on wikipedia with social tags. *arXiv preprint arXiv:1202.5469*.