RIPML: A Restricted Isometry Property Based Approach to Multilabel Learning

Akshay Soni
Yahoo! Research, Sunnyvale
akshaysoni@yahoo-inc.com

Yashar Mehdad
Airbnb, San Francisco
yashar.mehdad@airbnb.com

Abstract

The multilabel learning problem with large number of labels, features, and data-points has generated a tremendous interest recently. A recurring theme of these problems is that only a few labels are active in any given datapoint as compared to the total number of labels. However, only a small number of existing work take direct advantage of this inherent extreme sparsity in the label space. By the virtue of *Restricted Isometry Property* (RIP), satisfied by many random ensembles, we propose a novel procedure for multilabel learning known as RIPML. During the training phase, in RIPML, labels are projected onto a random low-dimensional subspace followed by solving a least-square problem in this subspace. Inference is done by a k-nearest neighbor (kNN) based approach. We demonstrate the effectiveness of RIPML by conducting extensive simulations and comparing results with the state-of-the-art linear dimensionality reduction based approaches.

1 Introduction

The task of multilabel learning is to predict a small set of labels associated with each datapoint out of all possible labels. Interest in these problems with large number of labels, features, and data-points has risen due to the applications in the area of image/video annotation (Qi et al. 2007), bioinformatics where a gene has to be associated with different functions (Barutcuoglu, Schapire, and Troyanskaya 2006), and entity recommendation for documents and images on a web-scale (Katakis, Tsoumakas, and Vlahavas 2008; Boutell et al. 2004). Modern applications of multilabel learning are motivated by recommendation and ranking problems; for instance, in (Agrawal et al. 2013) each search engine query is treated as a label and the task is to get the most relevant queries to a given webpage. Specific to Natural Language Processing (NLP), developing highly scalable approaches for multilabel text categorization is an important task for variety of applications such as relevance modeling, entity recommendation, topic labeling and relation extraction.

Recently, dimensionality reduction based approaches have gained popularity, for example, by using Compressive Sensing (CS) (Hsu et al. 2009; Kapoor, Viswanathan, and Jain 2012) and the state-of-the-art Low Rank Empirical Risk Minimization (LEML) algorithm (Yu et al. 2014). There has also been advances made in non-linear dimensionality reduction based approaches such as the X1 algorithm (Bhatia et al. 2015). These algorithms, even though being conceptually simple, are still computationally heavy. For instance, Compressive Sensing based approach has a very simple dimensionality reduction procedure based on random projections, but require to solve a sparse reconstruction problem during prediction which is the bottleneck.

To address these issues, we propose a novel approach that leverages the advantages of both Compressive Sensing and the non-linear X1 algorithm. RIPML benefits from a simple random projection based dimensionality reduction technique during training as in Compressive Sensing and then use a kNN based approach during inference as recently proposed in the X1 algorithm (Bhatia et al. 2015). The proposed approach is based on the fact that the number of active labels associated with a datapoint is significantly smaller than the total number of labels, making the label vectors sparse. During training, we exploit this inherent sparsity in the label space by using random projections as a means to reduce the dimensionality of the label space. By the virtue of Restricted Isometry Property (RIP), satisfied by many random ensembles, the distances between the sparse label vectors are approximately preserved in the projected low-dimensional space as well. Given the training feature vectors, we then solve a least-squares problem to predict the low-dimensional label vectors.

During inference, for a new datapoint, we use the output of the least-square problem to estimate the corresponding low-dimensional label vector and then use kNN in the low-dimensional label space to find the k-closest label vectors. In this way, the labels that occur many times in these k-closest label vectors then become the estimated labels for this new datapoint. However, as noted by authors in (Bhatia et al. 2015), kNN is known to be slow if the search for nearest neighbors involve large number of data points which is generally the case. We then leverage the solution provided in (Bhatia et al. 2015) and cluster the training data into multiple clusters and apply RIPML to each cluster separately.

1.1 Related Work

The main advantages of embedding based methods is their simplicity, ease of implementation, strong theoretical foundations, the ability to handle label correlations, the ability

to adapt to online and incremental scenarios, and the ability to work in a language/domain ignorant manner. The idea of Compressive Sensing based approaches (Hsu et al. 2009; Kapoor, Viswanathan, and Jain 2012) is to project the high-dimensional label vector into a smaller random-subspace and then solve a sparse recovery problem in this low-dimensional space. The state-of-the-art (LEML) algorithm (Yu et al. 2014) leverages the low-rank of label matrix to learn the projection matrix and the back-projection matrix in order to estimate the label vectors by solving a single unified optimization problem.

The X1 algorithm (Bhatia et al. 2015) builds on the assertion that the critical assumption made by most dimensionality reduction based methods that the training label matrix is low-rank is violated in almost all the real world applications. The authors propose a locally non-linear embedding technique to reduce the dimension of the label vectors while approximately preserving the distances between them. Prediction is done by using kNN in this low-dimensional space over the training data.

2 RIPML

2.1 Background

In order to formulate the problem and present our approach, we first note down the definition of RIP and few matrices which satisfy this property.

Definition: *A matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ satisfy the (k, δ)-RIP for $\delta \in (0, 1)$, if*

$$(1 - \delta)\|\boldsymbol{x}\|_2^2 \leq \|\mathbf{\Phi}\boldsymbol{x}\|_2^2 \leq (1 + \delta)\|\boldsymbol{x}\|_2^2 \tag{1}$$

for all k-sparse vectors $\boldsymbol{x} \in \mathbb{R}^n$.

While it is difficult to construct deterministic matrices which satisfy RIP, the best known guarantees arise from the random matrix theory. For example, following random ensembles satisfy RIP with high probability (Rudelson and Vershynin 2006)

- Gaussian matrix whose entries are i.i.d. $\mathcal{N}(0, 1/m)$ i.e. distributed normally with variance $1/\sqrt{m}$ for $m = \mathcal{O}(k \log(n/k))$
- Bernoulli matrix with i.i.d. entries over $\{\pm 1/\sqrt{m}\}$ with $m = \mathcal{O}(k \log(n/k))$

Note that if n is large and k is very small then we only need $m \ll n$ to satisfy RIP, giving a very low-dimensional distance preserving embedding. If a matrix $\mathbf{\Phi}$ satisfy $(2k, \delta)$-RIP, then for all k-sparse vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we have

$$(1 - \delta)\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \|\mathbf{\Phi}(\boldsymbol{x} - \boldsymbol{y})\|_2^2 \leq (1 + \delta)\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

which essentially means that the distance between the projected vectors $\mathbf{\Phi}\boldsymbol{x}$ and $\mathbf{\Phi}\boldsymbol{y}$ is close to the distance between the original vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. This distance preserving property of random projections is at the core of RIPML.

It is to be noted that the classical Johnson and Lindenstrauss Lemma (Dasgupta and Gupta 2003) shows that any set of points can be embedded in a lower-dimensional space while preserving the distances between them. RIP specializes that result and proves that some random ensembles indeed have this property, and can take advantage of the underlying sparsity to find a space of $\mathcal{O}(k \log(n/k))$ dimension to embed n points that are k sparse.

As noted above, there are many different random ensembles which satisfy RIP, but in this paper we report experimental results using Gaussian ensembles only.

Algorithm 1 RIPML: Inference

Inputs: Test point $\boldsymbol{x}_{\text{new}}$, no. of desired labels p, no. of nearest neighbors k, number of learners F, $\boldsymbol{Z}$, $\widehat{\mathbf{\Psi}}^f$ for $f \in [F]$, $\boldsymbol{Y}$

Step 1: For each $f \in [F]$ do:

a) $\boldsymbol{z}_{\text{new}}^f = \widehat{\mathbf{\Psi}}^f \boldsymbol{x}_{\text{new}}$

b) $\{i_1^f, i_2^f, \ldots, i_k^f\} \leftarrow \text{kNN}(k)$ in $\boldsymbol{Z}$

Step 3: $D = \frac{1}{Fk} \sum_{f=1}^{F} \sum_{i=i_1^f}^{i_k^f} \boldsymbol{y}_i$

Step 4: $\widehat{\boldsymbol{y}}_{\text{new}} \leftarrow \text{Top}_p(D)$

Output: $\widehat{\boldsymbol{y}}_{\text{new}}$

2.2 Algorithm

Training data is of the form $\{(\boldsymbol{x}_i, \boldsymbol{y}_i), i = 1, 2, \ldots, N\}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the feature vector, $\boldsymbol{y}_i \in \{0, 1\}^L$ is the binary label vector and L denotes the total number of labels. For $\ell \in [L]$[1], $\boldsymbol{y}_i[\ell] = 1$ denotes that the ℓ^{th} label is "present" and $\boldsymbol{y}_i[\ell] = 0$ denotes otherwise.

Training Procedure **Step 1 – Label Vector Dimensionality Reduction:** First, we project the training label vectors into a lower-dimensional space while approximately preserving the distances between them. This by the virtue of sparsity of label vectors is achieved by using a RIP satisfying matrix as a dimensionality reduction operator. That is, given a RIP satisfying matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times L}$, we get the low-dimensional label vectors as

$$\boldsymbol{z}_i = \mathbf{\Phi} \frac{\boldsymbol{y}_i}{\|\boldsymbol{y}_i\|_2} = \mathbf{\Phi}\widetilde{\boldsymbol{y}}_i \tag{2}$$

where $\boldsymbol{z}_i \in \mathbb{R}^m$ is the low-dimensional representation of $\boldsymbol{y}_i$. Note that the above matrix-vector product can be efficiently calculated by just adding entries of each row of $\mathbf{\Phi}$ corresponding to the nonzero locations of $\boldsymbol{y}_i$ and then normalizing the result by the square root of number of nonzero entries in $\boldsymbol{y}_i$. If there are s-nonzeros in $\boldsymbol{y}_i$, the above product can be computed in $\mathcal{O}(sm)$ operations rather then $\mathcal{O}(mL)$ operations, required if the label vectors were dense. Since we are operating under the assumption that $s \ll L$, the dimensionality reduction procedure adopted by us is efficient and fast. We normalize the label vectors in (2) in order to work with the cosine similarity as distance metric.

Step 2 – Least-Squares: Given $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ for $i \in [N]$, we want to learn a matrix $\mathbf{\Psi} \in \mathbb{R}^{m \times d}$ such that $\boldsymbol{z}_i \approx \mathbf{\Psi}\boldsymbol{x}_i$ for all $i \in [N]$. We propose to solve following least-square problem to learn $\mathbf{\Psi}$

$$\widehat{\mathbf{\Psi}} = \arg\min_{\mathbf{\Psi}} \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{z}_i - \mathbf{\Psi}\boldsymbol{x}_i)^2 + \lambda \|\mathbf{\Psi}\|_F^2 \tag{3}$$

[1]Here and in rest of the paper, for a non-negative integer M, the notation $[M]$ represents the set $\{1, 2, \ldots, M\}$.

where $\lambda \geq 0$ is the regularization parameter which controls the Frobenius norm[2] of the learned matrix. For reasonable feature dimension d, we can solve (3) in closed form, and if solving in closed form is not an option, we can use optimization approaches like gradient descent to solve it iteratively. The overall output of the training procedure is $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N] \in \mathbb{R}^{m \times N}$ and $\widehat{\boldsymbol{\Psi}}$.

Since our approach is randomized by the choice of $\boldsymbol{\Phi}$, we can learn multiple models for different instances of $\boldsymbol{\Phi}$, and combine their predictions to produce a more accurate model. Let F be the number of learners, then our training procedure gives us $\widehat{\boldsymbol{\Psi}}^f$ where $f \in [F]$. Unless stated otherwise $F = 5$ throughout the paper.

Inference Procedure Given a new feature vector $\boldsymbol{x}_{\text{new}}$, we want to predict the labels associated with it. Given $\widehat{\boldsymbol{\Psi}}^f$ for $f \in [F]$, the following two steps are executed:

Step 1 – Get $\boldsymbol{z}^f_{\text{new}} \in \mathbb{R}^m$: $\boldsymbol{z}^f_{\text{new}} = \widehat{\boldsymbol{\Psi}}^f \boldsymbol{x}_{\text{new}}$.

Step 2 – Find kNN of $\boldsymbol{z}^f_{\text{new}}$: Finds the indices of k vectors from $\boldsymbol{Z}$ which are closest to $\boldsymbol{z}^f_{\text{new}}$ in terms of squared distance. Say those indices are $i^f_1, i^f_2, \ldots, i^f_k$. Then we compute the empirical label distribution as $\frac{1}{Fk} \sum_{f=1}^{F} \sum_{i=i^f_1}^{i^f_k} \boldsymbol{y}_i$ out of which we can pick out the top-p locations corresponding to highest values and give them as an estimate of the labels associated with $\boldsymbol{x}_{\text{new}}$.

We use vanilla kNN for the experiments in this paper, but this step can be made scalable and fast by using techniques such as Locality Sensitive Hashing (Indyk and Motwani 1998). For certain random ensembles like Gaussian, the locality-sensitive functions are already well-known (Datar et al. 2004). In order to keep the exposition simple, we make note of these approaches, but use simple kNN to do the experiments.

2.3 Scaling to Large Datasets

Even though our training procedure is simple and scalable, kNN can be slow for datasets with large number of data points which increases the testing time. In order to tackle large datasets, we first cluster the feature vectors into C clusters using a simple procedure like KMeans. Then for each cluster c, we get the low-dimensional label vectors $\boldsymbol{Z}^c$ (Training – Step 1) and learn $\widehat{\boldsymbol{\Psi}}^c$ (Training – Step 2).

For a new test feature vector, we first find its cluster membership by finding the cluster-center closest to it, and then apply our testing procedure by using $\boldsymbol{Z}^c$ and $\widehat{\boldsymbol{\Psi}}^c$ for that cluster.

3 Experiments and Results

3.1 Experimental Settings

Baselines: Since our approach is based on linear dimensionality-reduction, we compare it with other state-of-the-art linear dimensionality reduction based approaches:

- LEML (Low rank Empirical risk minimization for Multi-Label Learning) with squared loss (Yu et al. 2014). The implementation of this algorithm was provided by the authors.
- CPLST[3] (Conditional Principal Label Space Transformation) (Chen and Lin 2012).
- CSSP (Column Subset Selection Problem) (Bi and Kwok 2013)

Datasets:

We perform experiments on five textual real world datasets. The first three are popular datasets that have been used in the previous works: Bibtex (Katakis, Tsoumakas, and Vlahavas 2008), EURLex (Loza Mencía and Fürnkranz 2010), and Delicious (Tsoumakas, Katakis, and Vlahavas 2008)[4]. These datasets are already partitioned into train and test which we use directly for our experiments. We call these group of datasets as **MLL** datasets.

In order to further prove the generalizability and scalability of our approach, we conduct the same set of experiments on other datasets. These datasets were created and will be released as a by-product of our contribution to this work:

-Relevance Modeling (Chinese Finance News): a set of financial news documents in Chinese with their relevant ticker symbols. In this dataset, each document is labeled with ticker symbols of the companies to which the document is relevant. The annotation was performed by an expert native language editorial team. The evaluation results over these two datasets measure how well our approach deals with documents from other languages and emphasize the generalizability of our approach.

-Entity Recommendation (English Wikipedia): we randomly sampled one million documents from English Wikipedia and labeled the documents with entities. The concept of entity in this work is referred to any segment of text that is linked to another page in Wikipedia. We then filtered out the entities which occurred less than 10 times in the entire dataset. Multilabel learning over such dataset is very challenging due to the number of labels (i.e., entities) in Wikipedia. This dataset measures how well our approach deals with very large number of labels and emphasize the scalability of our approach.

For the document embeddings of the above mentioned two datasets, we use doc2vec (Le and Mikolov 2014) to learn dense low-dimensional vectors. We train the embeddings of the words in documents using skip-bigram model (Mikolov et al. 2013) using hierarchical softmax training. For the embedding of documents we exploit the distributed memory model since it usually performs well for most tasks (Le and Mikolov 2014).

The dimension, number of labels and other statistics for the datasets are shown in Table 1.

Evaluation Criteria: Following the trail of the research in this field (Yu et al. 2014; Chen and Lin 2012; Bhatia et al.

[2]For a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, $\|\boldsymbol{X}\|_F^2 = \sum_{i,j} X_{ij}^2$

[3]The implementation for CPLST and CSSP was taken from: https://github.com/hsuantien/mlc_lsdr

[4]All of these standard datasets are available online at: http://mulan.sourceforge.net/datasets-mlc.html

Dataset	d	avg. nnz($\boldsymbol{x}$)	L	avg. nnz($\boldsymbol{y}$)	Total Datapoints	Train (N)	Test
Bibtex	1836	68.74	159	2.40	7395	4880	2515
EURLex	5000	236.69	3993	5.31	19314	17383	1931
Delicious	500	18.17	983	19.03	16091	12910	3181
Chinese Relevance Modeling	$100-400$	dense	391	1.02	5011	4511	500
Entity Recommendation	400	dense	359524	32.55	510539	500539	10000

Table 1: Statistics of different datasets used in this paper. Here, avg. nnz($\boldsymbol{y}$) denotes the average number of labels per data-point. Similarly, avg. nnz($\boldsymbol{x}$) denotes the average number of non-zero features per data-point.

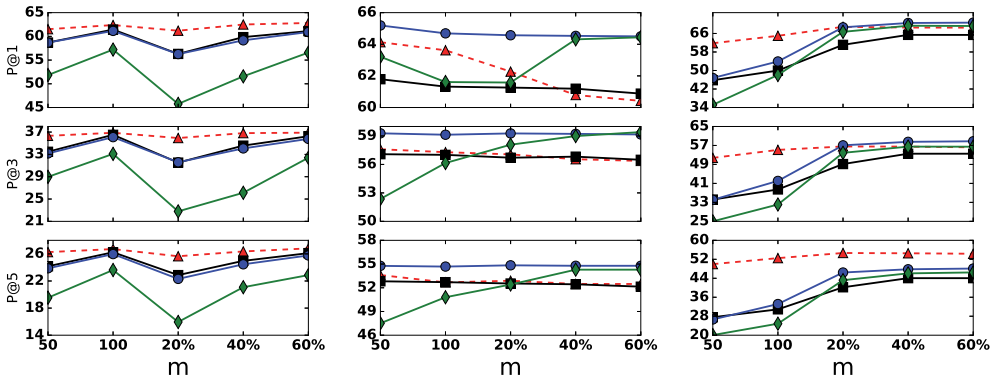

Figure 1: Column 1, 2 and 3 represent Bibtex, Delicious, and EURLex datasets. Results for standard datasets: Bibtex, Delicious and EURLex. Legend: RIPML (- -▲- -), LEML (–■–), CPLST (–●–), CSSP (–◆–). Y-axis: m = [50, 100, 20% of L, 40% of L, 80% of L]. Here $k = 5$.

2015), we take precision at K (P@K) as our evaluation criteria. Precision at K is the fraction of correct labels in the top-K label predictions. For the ease of comparison with other research papers we use $K = 1,\ 3$ and 5 in this paper.

3.2 Results on MLL datasets

The P@1, P@3 and P@5 results are tabulated in Figure 1. We can observe that our results outperform the strong baselines in Bibtex and EURLex datasets. However, for Delicious dataset, our approach performs worse than CPLST. We suspect this might be due to the fact that the average number of nonzero labels per data-point is quite large for this dataset and also the features are very sparse, see Table 1.

An interesting point to note from these results is that RIPML remains stable with varying m while other approaches start with a lower precision for small m and then gradually improve with increasing m. This stability of RIPML can be attributed to RIP which allows to obtain a stable distance-preserving embedding with $m = \mathcal{O}(s\log(s))$ where s is the maximum number of nonzeros in the label vectors. Thus, increasing m above a certain threshold only results in a marginal improvement. This property also allows RIPML to perform better than other algorithms for small m – see results for Bibtex, and EURLes for $m = 50$ or 100.

Figure 2 shows the variation of precision with number of nearest neighbors used for kNN. Bibtex and Delicious are quite stable with the choice of number of nearest neighbors but EURLex performs better with smaller number of nearest neighbors.

3.3 Results on Relevance Modelling Datasets

In this work, we obtain the low-dimensional embedding of documents using the method described in Section 3 with window sizes of 10. In order to see the effect of dimensions, we also experiment with four low-dimensional models (200, 300 and 400 dims). During training the document embeddings we limit the number of iterations to 10 to increase the efficiency.

Figure 3 shows the results for Chinese relevance modeling datasets with different ambient dimensions. All these results are averaged over 5 random train-test splits. It is interesting to note that RIPML performs better than CPLST and CSSP always, but performs worse than LEML in certain cases. Another interesting observation is that even though the P@1 is good for these datasets, P@5 drops significantly for all the approaches. This is due to the very low number of relevant tickers for each document (average number of relevant tickers per document is about 1). In conclusion, RIPML performs well for detecting the relevant tickers, considering that our approach doesn't require any linguistic preprocessing which is one of the main challenges in multilingual NLP community.

Figure 4 shows the variation of precision with respect to number of nearest neighbors used during prediction for these two datasets. For these datasets, small number of nearest neighbors results in a better precision, due to very less average number of relevant ticker symbols per document.

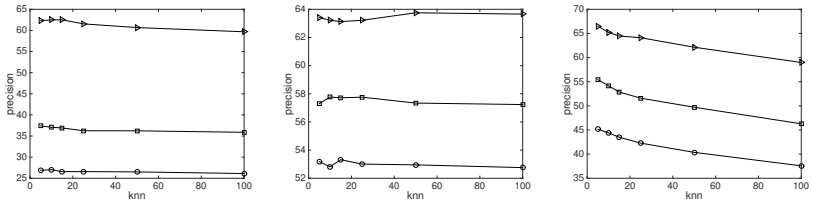

Figure 2: Column 1, 2 and 3 represent Bibtex, Delicious, and EURLex datasets. Row-1: precision@{1,3,5} vs. k for kNN. Here –▷–, –□– and –○– corresponds to precision@1, 3 and 5 respectively. Here $m = 100$.

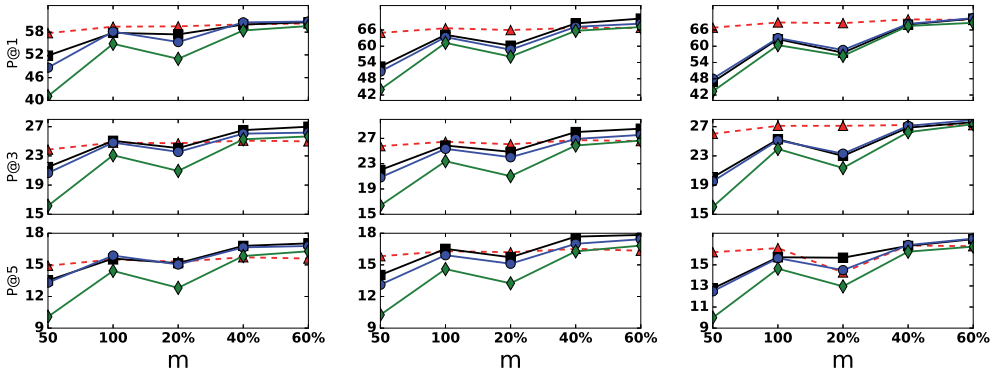

Figure 3: Column-1, 2 and 3 correspond to $d = 200, 300$, and 400 respectively. Results on Chinese Relevance Modeling for different ambient dimensions. Legend: RIPML (- -▲- -), LEML (–■–), CPLST (–●–), CSSP (–◆–). X-axis: m = [50, 100, 20% of L, 40% of L, 80% of L]. Here $k = 5$.

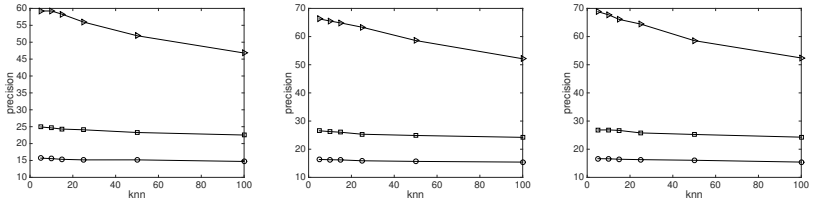

Figure 4: Column-1, 2 and 3 correspond to $d = 200, 300$, and 400 respectively. Row-1: precision@{1,3,5} vs. k for kNN for Chinese Ticker dataset Here –▷–, –□– and –○– corresponds to precision@1, 3 and 5 respectively. Here $m = 100$.

3.4 Results on Entity Recommendation Dataset

We created this dataset from Wikipedia to show how clustering can be used to scale to big datasets with very large number of labels, and the effect of clustering on performance. In order to get a baseline to compare the effect of clustering, we first conduct experiments without using the clustering step. Given that our training and testing procedure is very efficient, we were able to train our model on this data in less than 3 minutes on a laptop[5]. We used 10000 data points for testing and rest for training. Our results are averaged over 5 random train-test split. It took approximately 72 milliseconds to predict labels for each test data point. We are not providing any comparison for this dataset with other approaches because we were not able to train a model within reasonable amount of time.

In order to improve the performance for this challenging dataset, we then applied kMeans clustering algorithm to cluster the training data (features) into different number of clusters and trained RIPML on each of the clusters separately by borrowing ideas from (Bhatia et al. 2015). For every test data-point, we first figure out which cluster it belongs to by finding the nearest cluster center and then apply the kNN based prediction procedure on that cluster. Clustering helps us in two ways – it makes prediction faster by allowing us to do kNN on a small amount of data and it also increases the prediction accuracy. This is evident from the results shown in Table 2 where we present precision for varying number of clusters. Going from no clustering to around 43 clusters increases the precision by about 10%. With 43 clusters it took $\sim$ 7 minutes to train and approximately 6.5 milliseconds to predict labels for each test data point.

[5]Apple MacBook Pro with 2.5GHz Intel Core i7 and 16 GB RAM.

Entity Recommendation (d = 400)												
	Clusters = 1			Clusters = 25			Clusters = 43			Clusters = 56		
m	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5
50	31.02	23.60	20.20	39.42	31.72	27.61	41.72	33.67	29.32	41.23	33.69	29.54
100	33.55	25.58	21.70	42.10	34.24	29.89	44.11	36.08	31.39	43.91	36.16	31.65
250	35.76	27.20	23.46	44.44	35.81	31.38	46.28	37.95	33.15	46.30	37.92	33.45

Table 2: Results on Entity Recommendation dataset. Clusters = 1 means that clustering step was not done.

4 Conclusions

In this paper, we presented a novel, scalable, and general multilabel learning algorithm based on random-projections and kNN called RIPML. We demonstrated its performance on six different real world datasets which includes three popular and three new datasets. The new datasets would also be released for the use of research community as a part of this work.

We would like to extend our algorithm to explore the problem of missing label cases. We also plan to study the impact of using different RIP, satisfying random/deterministic ensembles. Moreover we will investigate the performance of other loss functions for the regression phase.

References

Agrawal, R.; Gupta, A.; Prabhu, Y.; and Varma, M. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 13–24. New York, NY, USA: ACM.

Barutcuoglu, Z.; Schapire, R. E.; and Troyanskaya, O. G. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics* 22(7):830–836.

Bhatia, K.; Jain, H.; Kar, P.; Jain, P.; and Varma, M. 2015. Locally non-linear embeddings for extreme multi-label learning. *CoRR* abs/1507.02743.

Bi, W., and Kwok, J. 2013. Efficient multi-label classification with many labels. In Dasgupta, S., and Mcallester, D., eds., *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, 405–413. JMLR Workshop and Conference Proceedings.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition* 37(9):1757 – 1771.

Chen, Y.-n., and Lin, H.-t. 2012. Feature-aware label space dimension reduction for multi-label classification. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1529–1537.

Dasgupta, S., and Gupta, A. 2003. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms* 22(1):60–65.

Datar, M.; Immorlica, N.; Indyk, P.; and Mirrokni, V. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, 253–262. ACM.

Hsu, D.; Kakade, S. M.; Langford, J.; and Zhang, T. 2009. Multi-label prediction via compressed sensing. *CoRR* abs/0902.1284.

Indyk, P., and Motwani, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 604–613. ACM.

Kapoor, A.; Viswanathan, R.; and Jain, P. 2012. Multilabel classification using bayesian compressed sensing. 2645–2653.

Katakis, I.; Tsoumakas, G.; and Vlahavas, I. 2008. Multilabel text classification for automated tag suggestion. In *In: Proceedings of the ECML/PKDD-08 Workshop on Discovery Challenge*.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In Jebara, T., and Xing, E. P., eds., *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 1188–1196. JMLR Workshop and Conference Proceedings.

Loza Mencía, E., and Fürnkranz, J. 2010. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In Francesconi, E.; Montemagni, S.; Peters, W.; and Tiscornia, D., eds., *Semantic Processing of Legal Texts – Where the Language of Law Meets the Law of Language*, volume 6036 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1 edition. 192–215. accompanying EUR-Lex dataset available at http://www.ke.tu-darmstadt.de/resources/eurlex.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Qi, G.-J.; Hua, X.-S.; Rui, Y.; Tang, J.; Mei, T.; and Zhang, H.-J. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, 17–26. New York, NY, USA: ACM.

Rudelson, M., and Vershynin, R. 2006. Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *Information Sciences and Systems, 2006 40th Annual Conference on*, 207–212. IEEE.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2008. Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, XX.

Yu, H.-F.; Jain, P.; Kar, P.; and Dhillon, I. S. 2014. Large-scale Multi-label Learning with Missing Labels. In *ICML*.