Undergraduate Project
Report

Extreme Multi-label Learning with Mixture of Factor Analyzers

Submitted in partial fulfillment of
the requirements for the course CS498A

Submitted by

Vikas Jain
13788

Under the guidance of
Prof Piyush Rai

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur, U.P. , India – 208 016

November 23, 2016

Abstract

Consider the following tasks: Given a product on an e-commerce site, label it with all its relevant attributes; given an image, label it with all the objects/semantics relevant to the image; given a text document, label it with all its relevant tags. Such a problem, commonly known as *Multi-label Learning*, is a generalization of the canonical binary classification problem; essentially, for each input, instead a single binary label (yes/no), we want to predict a binary label vector (denoting the presence/absence of each label from a large vocabulary of labels). Real-world multi-label learning problems have to routinely deal with a massive number of labels (which could easily to tens of thousands to millions), and such problems are known as *eXtreme Multi-label Learning* (XML). A popular strategy for XML exploits the fact that the label vectors, even though high dimensional, can be "compressed" into a lower-dimensional space. This speeds up multi-label learning algorithms, while also leveraging the label correlations. The low-rank assumption however is often violated especially when many labels are rare. To address this issue, we propose a novel generative model based on an (input-dependent) mixture of factor analyzer models, where each factor analyzer is essentially a "local" low-rank model. To facilitate "out-of-sample" predictions on test data, we condition the mixture memberships and label vector embeddings learned by the model, directly on the inputs. We also propose an efficient online Expectation Maximization algorithm to perform parameter estimation in this model, which we further simplify by leveraging data-augmentation methods.

Acknowledgements

The course CS498A is offered every year for undergraduate students at CSE, IITK. The course involves a research project to be completed under the guidance of a faculty professor. The course gives the opportunity to the students to try hands on research experience in their field of interest.

I would like to express my gratitude towards **Prof Piyush Rai** for his kind co-operation and encouragement that helped me in the completion of this project. This project would not have been possible without his kind support.

I would also like to express my special gratitude and thanks to the department UG convener **Prof Nitin Saxena** and course TAs for coordinating the logistics of the course.

The acknowledgement is as important as any other part of the report which is usually taken for granted and skipped. If you are reading this, I would like to thank you as a reader.

Vikas Jain

November 2016
Indian Institute of Technology

Contents

1 Introduction

1.1 Overview

Binary classification is a problem of classifying data into two categories. A simple example of binary classification is spam filter: Given a mail, classify it into either spam or non-spam category (or more generally probability of the mail being spam). These problems are of very important status in machine learning community that provide solution to various real-life problems. A straight-forward extension to the problem of binary classification is Multi-class Classification. In Multi-class classification, instead of binary classes, multiple classes are present and each data point belongs to only one of the class. An popular example of multi-class classification is *Imagenet Challenge* [2]. Imagenet is a huge dataset with over a million image and each image belonging to one of the thousands total classes.

In contrast to Binary and Multi-class Classification, *Multi-label learning* is the problem of assigning multiple possible labels to each data point from a set of possible labels. Figure 1 shows the general problem setting of Multi-label learning. Given a data point, predict a vector of length of number of all possible labels with each entry either 0 or 1 representing the existence of the corresponding label in the data point.

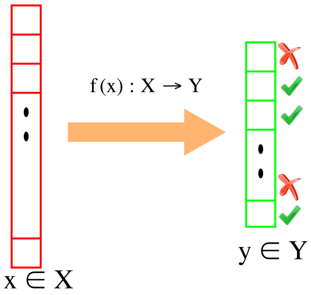

Figure 1: Multi-label Learning

The number of real-life examples of multi-label learning makes it an interesting as well as a challenging problem. The most notable example is *document tagging*: Given a text article, label the article with all relevant tags. Other examples are recommendations on e-commerce, tagging images etc. The figures below show some of the application areas of multi-label learning.

Recommend multiple movies to a user [1]

NaMo TURNS INDIA RIGHT

Will Carry All Along, Says Modi in Victory Speech

GOOD DAYS WILL BE HERE SOON... I WANT TO ASSURE THE PEOPLE OF INDIA THAT OUR MOTTO IS TO CARRY ALONG EVERYONE HOW MUCH EVER THEY MAY OPPOSE US

Score BOARD

HERE'S WHAT THE NATIONAL AND REGIONAL PARTIES MUSTERED IN THE 543-STRONG LOK SABHA (WON + LEADING)

NDA 338

THE NDA IS IN A MINORITY IN THE RAJYA SABHA AND THE BJP HOPES TO WIN THE CO-OPERATION OF PARTIES FROM THE EAST AND SOUTH

Tag the article [2]

1.2 eXtreme Multi-label Learning

The problem of multi-label learning becomes difficult to learn when of number of labels possible becomes very large. The number of labels can be ten of thousands to millions. The problem of multi-label learning with huge number of labels is termed as *eXtreme Multi-label Learning* (or the popular acronym XML). Traditional Multi-label learning methods challenge the XML problem from various dimenstions:

- Training time
- Testing time
- Scalability
- Model Size
- Accuracy

An efficient solution should address all the above mentioned issued faced in XML. Many existing solutions for XML exploit the fact that the label vectors, even though high-dimensional, can be embedded into a lower-dimensional space. This speeds up multi-label learning algorithms, while also leveraging the label correlations. However, this low-rank assumption is not completely true because many labels are rare. An effective solution for XML should address this misleading low-rank assumption of label vectors.

1.3 This work

In this work, we propose a model for solving the problem of eXtreme multi-label learning. We propose a probabilistic model which embed the label

[1] from http://papersol.blogspot.in/p/vaughdls-movie-recommendations.html

[2] from Indian Express Newspaper, 2014

vectors into a lower-dimensional space using a factor analyzer model. The embedding are modeled using mixture of experts from the input data. In short, the model proposed tries to find the embedding of label vectors after dividing it into groups using the input data. In the next section, some previous work and techniques of XML are presented. The subsequent sections explain the model purposed.

2 Related Work

2.1 Three popular methods

The following three methods are generally used for solving multi-labeling problem:

- **1 vs all method**: In this method, a classifier is learnt for each of the all possible labels. But this method is not scalable, efficient and feasible for XML problems.
- **Tree based methods**: In this method, a decision tree is learnt to find out the possible labels for each data point. The model has very low prediction time which becomes the USP of the method. But to make the method accurate enough, an ensemble of trees is learnt which shoots the model size of this method in case of a XML problem.
- **Embedding based method**: In this method, an embedding of label vectors is learnt to embed label vectors in a lower-dimensional space. These methods are most appropriate for XML methods as they reduce model size as well as prediction time, making the model scalable.

The table below 1 provides a brief comparison of these three methods:

	1 vs all	**Tree based**	**Embedding**
Accuracy	Poor	Good/Best	Good/Best
Scalability	Poor	Good/Best	Good/Best
Prediction Cost	Poor	Best	Good
Model Size	Poor	Large	Good
Theory	Best	Poor	Good

Table 1: Comparison of three popular multi-label learning methods (adapted from [4])

The method proposed in this work in based on the embedding based methods as they have been proved to be successful for extreme multi-label learning paradigm.

2.2 Current SoTA

Recently, an embedding based method is proposed for XML – *SLEEC* - Sparse Local Embeddings for Extreme Multi-label Classification [1]. This method is based on embedding based methods. The main idea of this work is to learn a small ensemble of models which embed label vectors into a lower dimension while preserving the local distance in the embedded space. This method addresses both the issues of the XML problem – circumventing hard low rank assumption as well as scaling up the model to real world problems.

3 Methodology

3.1 Problem Formulation and Notation

We have input data with features, say a data point $x \in \mathbb{R}^q$. We have a set of p labels. The problem is to assign a *subset* of p labels to each data point.

Each data point is represented as $\boldsymbol{x} \in \mathbb{R}^q$ and each label vector is represented as $\boldsymbol{y} \in \{0,1\}^p$. The training data is given as set of input data $\boldsymbol{x}$ along with corresponding label vectors $\boldsymbol{y}$ *i.e.* $\{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^N$. During testing, a new input data $\boldsymbol{x_{new}}$ is given and the model learn will predict $\boldsymbol{y_{new}}$.

A d dimensional guassian is represented as $\mathcal{N}_d(\mu, \Sigma)$ where μ and Σ are the mean and the covariance matrix respectively.

3.2 Overview of Methodology

We propose a model in which the label vectors $\boldsymbol{y}$ are embedded into a lower dimensional space using a *Factor Analyzer model*(FA). Embedding label vectors into a suitable lower-dimension scale up the model. But a FA model assumes a low rank assumption on the label vectors which is violated if the labels are rare. To overcome this problem, the embedding can be learnt using the input data features $\boldsymbol{x}$. The embeddings can be modelled using a *mixture of experts*(MoE) model which basically model the embeddings using K gaussians where each gaussian mean is learnt via a linear regressor on the input data. In the subsequent section, FA, MoE and the combined model are explained in details.

3.3 Factor Analyzer(FA)

In a factor analyzer model[3], label vectors are modeled as:

$$\boldsymbol{y} = \Lambda \boldsymbol{z} + \epsilon$$

where $\boldsymbol{z} \in \mathbb{R}^r$ s.t. $r << p$, $\Lambda \in \mathbb{R}^{p \times r}$ is known as the factor loading matrix. ϵ is the noise and modelled as

$$\epsilon \sim \mathcal{N}_r(0, \Psi)$$

where $\Psi \in \mathbb{R}^{r \times r}$ is a diagonal matrix.

In this model, the probabilty distribution of $\boldsymbol{y}$ is given as:

$$p(y) \sim \mathcal{N}_p(0, \Lambda\Lambda^T + \Psi)$$

3.4 Mixture of Experts(MoE)

We have label vectors $\boldsymbol{y}$ and the corresponding input data $\boldsymbol{x}$. In a MoE model[7], the label vectors $\boldsymbol{y}$ are modelled as mixture of K gaussians where the parameters of each gaussian (mean and covariance) depend on the input data $\boldsymbol{x}$.

The probability distribution of the label vector $\boldsymbol{y}$ is given as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{k=1}^{K} p(\omega = k|\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}, \omega = k)$$

and probability distributions of ω and $\boldsymbol{y}|\boldsymbol{x}, \omega$ are given as:

$$p(\boldsymbol{y}|\boldsymbol{x}, \omega = k) = \mathcal{N}_p(\boldsymbol{y}|W\boldsymbol{x}, \Psi_k)$$

where $W \in \mathbb{R}^{p \times q}$ is linear regressor and Ψ_k is covariance matrix.

$$p(\omega = k|\boldsymbol{x}) = multinoulli(\sigma(\eta_k^T \boldsymbol{x}))$$

where σ is the sigmoid function and η_k is the weight vector.

3.5 Proposed Combined Model

The proposed model combines the factor analyzer model and the mixture of expert model. The MoE model is applied on the embedding of label vector $\boldsymbol{y}$. Figure 2 shows the graphical representation of the model.

The model proposed is similar to the model proposed by Montanari et al. [6] but they are not trying to solve a XML problem using the model. Inspired from their work the model equations can be written as:

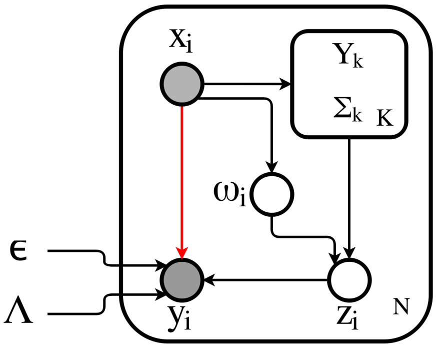

Figure 2: Proposed FA+MoE model

1. $\boldsymbol{y} = \Lambda \boldsymbol{z} + \epsilon$
2. $p(\boldsymbol{z}|\boldsymbol{x}) = \Sigma_{k=1}^{K} \pi_k \mathcal{N}_r(\mu_k, \Sigma_k)$
3. $\mu_k = \Upsilon_k \boldsymbol{x}$
4. $\pi_k(\boldsymbol{x}) = p(\omega_k = 1|x) = \frac{e^{\eta_k^T \boldsymbol{x}}}{1+\Sigma_{k'=1}^{K-1} e^{\eta_k'^T \boldsymbol{x}}}$
5. $p(\boldsymbol{z}|\boldsymbol{x}) = \Sigma_{k=1}^{K} \pi_k(\boldsymbol{x}) \mathcal{N}_r(\Upsilon_k \boldsymbol{x}, \Sigma_k)$
6. $p(\boldsymbol{y}|\boldsymbol{x}) = \Sigma_{k=1}^{K} \pi_k(\boldsymbol{x}) \mathcal{N}_p(\Lambda \Upsilon_k \boldsymbol{x}, \Lambda \Sigma_k \Lambda^T + \Psi)$

The equation 6 is the final equation of probablity distribution of label vector $\boldsymbol{y}$ given input data $\boldsymbol{x}$, derived from equations 1–5.

All the model parameters $(\Lambda, \Psi, \Upsilon_k, \Sigma_k, \eta_k)$ are learnt using *Expectation-Maximization* algorithm as described in [6].

3.6 Model Adaption to XML

The current model proposed is not ideal for solving XML based problems due to two reasons. First, EM algorithm is slow as it compute complete log-likelihood of all given input data. Second, the model proposed assumes that the label vector $\boldsymbol{y}$ belongs to $\mathbb{R}^p$ which is incorrect for XML problem where $\boldsymbol{y} \in \{0,1\}^p$. To solve these two issues, following changes are incorporated:

- **Online EM**[5] Parameters are updated by computing complete log-likelihood over a mini batch rather than over the whole data:

$$\Theta_{t+1} = (1 - \gamma_{t+1})\Theta_t + \gamma_{t+1}\Theta_{mini}$$

- **EM for logistic regression**[8] EM is modified to incorporate binary labels which are modelled using Binomial distribution.

$$y_t \sim Binom(1, w_t)$$

$$\psi_t = log(\frac{w_t}{1 - w_t}) = x^T \beta$$

  Determine β using EM — $\hat{\beta}$

$$\hat{w}_t = \frac{m_t}{2\hat{\psi}_t} tanh(\hat{\psi}_t/2) \ with \ \hat{\psi}_t = x_t^T \hat{\beta}$$

4 Contributions

The code for the proposed model is made available publicly by [6] written in R language. The experiments done on the top of the code are:

- Extended the code provided by adding functionality for running the code with **Online EM** algorithm.
- Run the code for 2 datasets.
- The online EM algorithm shows significant time improvement for datasets with number of labels $\sim$ 100.

5 Future Work

Currently, the implementation support only online EM for the proposed model. To make it a complete model for the XML problem, EM algorithm for logistic regression has to be incorporated. Following are the broad work to be done in the future:

- Implementing EM algorithm for logistic regression to predict binary labels.
- Rigorous empirical analysis of the model – possibly with huge number of labels and calculating the accuracies of the results using suitable metrics.

6 Conclusions

In this work, we propose a novel method for the eXtreme multi-label learning (XML) problem. The model proposed is a probabilistic model from the family of embedding based methods where embeddings are conditioned using the input data features. the label vectors are modelled as Factor Analyzer model where factors can be interpreted as low-dimensional embedding of the label vectors. The embeddings are modelled using Mixture of Experts where each expert is a Gaussian model with its mean as linear regressor of the input data feature. The model is adapted to the XML problem using online EM and EM for logistic regression to predict binary labels instead of real values. This model jointly solves the scalability as well as low-rank issues of XML. The initial results using this method show promising results on the improvement of training time.

References

[1] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738, 2015.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[3] Zoubin Ghahramani and Geoffrey E. Hinton. The em algorithm for mixtures of factor analyzers. Technical report, University of Toronto, 1996.

[4] Purushottam Kar. Lecture on multi-label learning, http://www.cse.iitk.ac.in/users/sigml/lec/Slides/MLL.pdf.

[5] Percy Liang and Dan Klein. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 611–619. Association for Computational Linguistics, 2009.

[6] Angela Montanari and Cinzia Viroli. Dimensionally reduced mixtures of regression models. *Journal of Statistical Planning and Inference*, 141(5):1744–1752, 2011.

[7] Kevin P Murphy. *Machine learning: a probabilistic perspective.* The MIT Press, 2012.

[8] James G Scott and Liang Sun. Expectation-maximization for logistic regression. *arXiv preprint arXiv:1306.0040*, 2013.