eXtreme Multi-label Learning Using Mixture of Factor Analyzers

Undergraduate Project

Vikas Jain
under the guidance of
Prof Piyush Rai

Department of Computer Science
IIT Kanpur

December 19, 2016

Outline

1. Introduction
   - Multi-label Learning
   - Notation and Formulation
2. Related Work
   - Three Popular Methods
   - Embedding Method
3. Methodology
   - Methodology-Prelims
   - Methodology-Mains
4. Code Used and Contributions
5. Future Work

Multi-label Learning

Multi-label Learning

Given an instance, assign all the relevant labels to it.

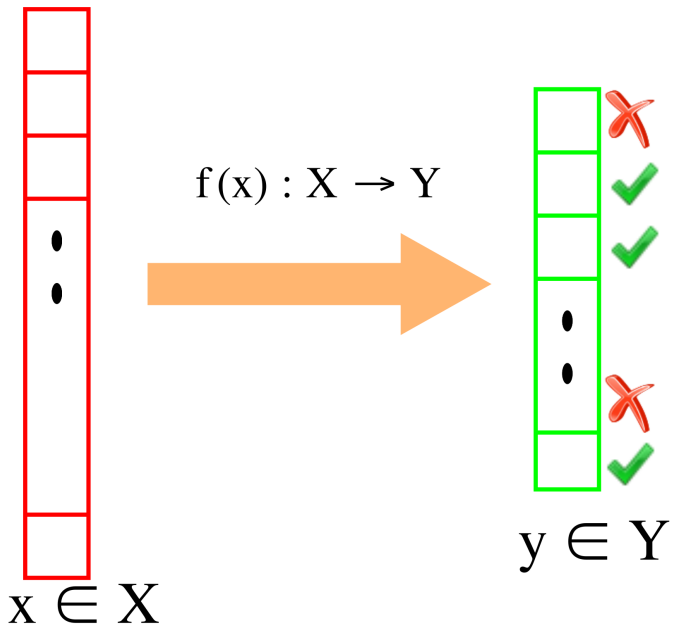


Figure: Mulit-label Learning

Multi-label Learning

Examples

What movies the user is likely to see?

Figure: from[1]

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

Multi-label Learning

Examples

Given an article, mark with all the tags

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

NDA 338

Figure: from [2]

Tags: Politics, Election, India, Narendra Modi ...

[2] Indian Express Newspaper, 2014

Notation and Formulation

Problem: We have objects (with features) that are to be assigned a subset of p labels.

Representation:

- objects: vector $\boldsymbol{x}$ of dim $q \times 1$, $x_i \in \mathbb{R}$.
- labels: vector $\boldsymbol{y}$ of dim $p \times 1$, $y_i \in \{0, 1\}$.

Training data: $\{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^{N}$

Testing: Given $\boldsymbol{x_{new}}$, predict $\boldsymbol{y_{new}}$

Lots of labels?

If $p >> 1$, eXtreme Multi-label Learning (XML)

Outline

1. Introduction
   - Multi-label Learning
   - Notation and Formulation
2. Related Work
   - Three Popular Methods
   - Embedding Method
3. Methodology
   - Methodology-Prelims
   - Methodology-Mains
4. Code Used and Contributions
5. Future Work

Related Work
Three Popular Methods

Three popular methods [3]:

- 1 vs all Method: Learn a classifier for each label type.
- Tree Method: Learn a decision tree with leaf nodes represent the p labels.
- Embedding Methods: Since $p >> 1$, embed labels in a smaller dimensional space and predict in the embedding space.

Comparison[3]:

Name	"Accuracy"	Scalability	Prediction Cost	Model Size	Well Understood?
1-vs-All	Meh!	Yikes!	Are you kidding me!	Did I not make myself clear?	Now we are talking! Excellent
Embedding	Good/ Best	Good/ Best	Good	Good	Good
Tree	Good/ Best	Good/ Best	Best	Large	Meh!

[3] courtesy: Prof P Kar talk on XML, IITK (link here)

Related Work
Embedding Method

- Since $p >> 1$, and labels are few, label matrix is redundant.
- How to embed?
- How to train in embedding space?
- How to project back?
- Main challenges: prediction time, training time.
- Current SoTA: an Embedding method, SLEEC [1]

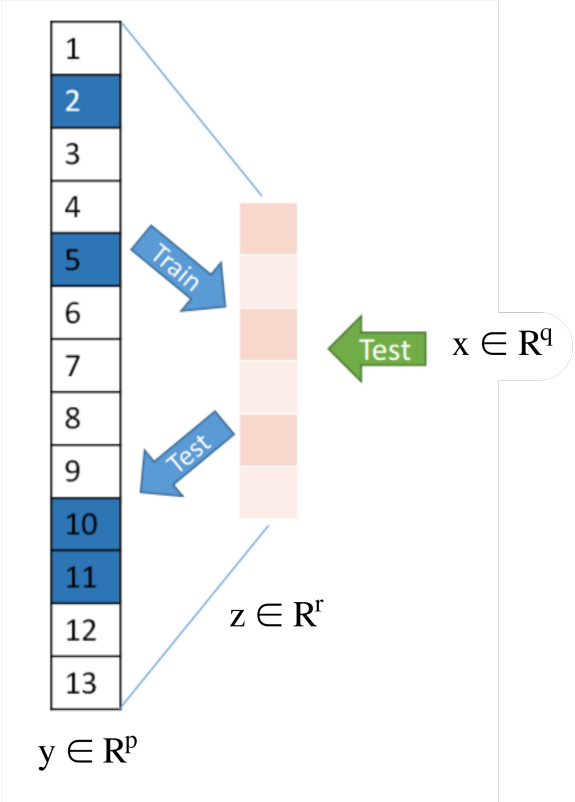


Figure: Label Vector Embedding [4]

[4] courtesy: Prof P Kar talk on XML, IITK (link here)

Outline

1. Introduction
   - Multi-label Learning
   - Notation and Formulation
2. Related Work
   - Three Popular Methods
   - Embedding Method
3. Methodology
   - Methodology-Prelims
   - Methodology-Mains
4. Code Used and Contributions
5. Future Work

Methodology-Prelims

Factor Analysis(FA)[2]

Since $p >> 1$, we can embed label vectors $\boldsymbol{y}$ using factor analyzer model (ignore inputs $\boldsymbol{x}$ as of now).

$$y = \Lambda z + \epsilon$$

where $y \in \mathbb{R}^p, \Lambda \in \mathbb{R}^{p \times r}, z \in \mathbb{R}^z$ and $\epsilon \sim \mathcal{N}_p(0, \Psi), z \sim \mathcal{N}_r(0, I)$ where Ψ is an diagonal matrix as we are assuming label variables in z-embedding are uncorrelated.

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

Methodology-Prelims

Factor Analysis(FA)[2]

Since $p >> 1$, we can embed label vectors $\boldsymbol{y}$ using factor analyzer model (ignore inputs $\boldsymbol{x}$ as of now).

$$y = \Lambda z + \epsilon$$

where $y \in \mathbb{R}^p, \Lambda \in \mathbb{R}^{p \times r}, z \in \mathbb{R}^z$ and $\epsilon \sim \mathcal{N}_p(0, \Psi), z \sim \mathcal{N}_r(0, I)$ where Ψ is an diagonal matrix as we are assuming label variables in z-embedding are uncorrelated.

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

BUT there is a problem? Many labels are very rare and the label vector matrix over N inputs is not low-rank.
Solution: Assume label vectors are low-rank over small groups of the data. Use Mixture of Factor Analyzers(MFA)

Methodology-Prelims

Mixture of Factor Analysis(MFA)[2]

Use k factor analyzer models.
Suppose we have a latent indicator $\omega_i \in \{1, \ldots, K\}$ specifying which subspace we should use to generate the data.

$$p(y|z_i, \omega_i = k) = \mathcal{N}_p(\mu_k + \Lambda_k z_i, \Psi)$$

$$p(z_i) = \mathcal{N}_r(0, I)$$

$$p(\omega_i) = multinoulli(\pi)$$

Methodology-Prelims

Mixture of Factor Analysis(MFA)[2]

Use k factor analyzer models.
Suppose we have a latent indicator $\omega_i \in \{1, \ldots, K\}$ specifying which subspace we should use to generate the data.

$$p(y|z_i, \omega_i = k) = \mathcal{N}_p(\mu_k + \Lambda_k z_i, \Psi)$$

$$p(z_i) = \mathcal{N}_r(0, I)$$

$$p(\omega_i) = multinoulli(\pi)$$

BUT we also have $\boldsymbol{x}$ input features?
Solution: Mixture of Experts!

Methodology-Prelims

Mixture of Experts(MoE)[5]

Since we have input features of each data point, we can condition label vectors ***y*** using input features ***x***

$$p(y_i|x_i, \omega_i = k) = \mathcal{N}_p(y_i|w_k^T x_i, \Psi)$$

$$p(\omega_i|x_i) = multinoulli(\sigma(\eta^T x_i))$$

The overall prediction of the model, obtained using

$$p(y_i|x_i) = \Sigma_{k=1}^{K} p(\omega_i = k|x_i) p(y_i|x_i, \omega_i = k)$$

Methodology-Prelims

Mixture of Experts(MoE)[5]

Since we have input features of each data point, we can condition label vectors $\boldsymbol{y}$ using input features $\boldsymbol{x}$

$$p(y_i|x_i, \omega_i = k) = \mathcal{N}_p(y_i|w_k^T x_i, \Psi)$$

$$p(\omega_i|x_i) = multinoulli(\sigma(\eta^T x_i))$$

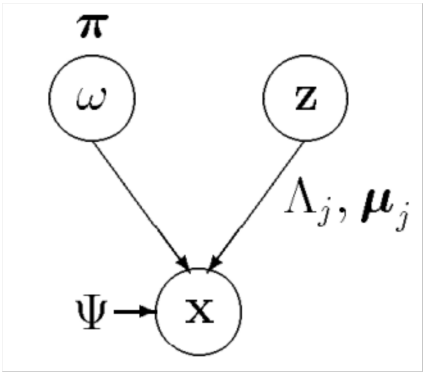
The overall prediction of the model, obtained using

$$p(y_i|x_i) = \Sigma_{k=1}^{K} p(\omega_i = k|x_i) p(y_i|x_i, \omega_i = k)$$

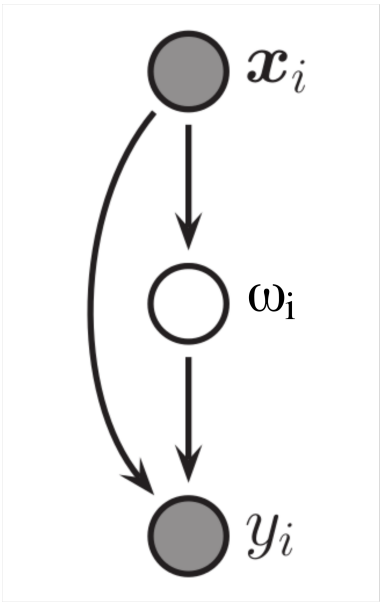
BUT we are not using the fact tha label vectors are low-rank.
And computation is expensive in MoE

Methodology-Mains

Combining MFA and MoE



MFA[2]



MoE[5]

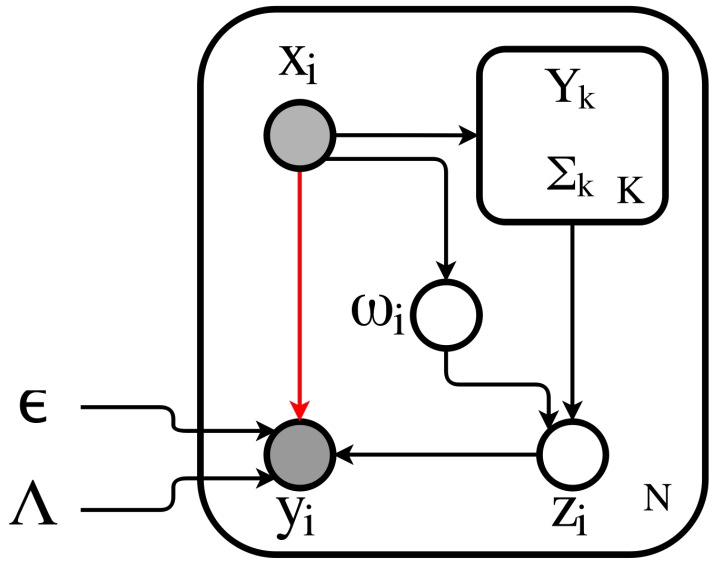
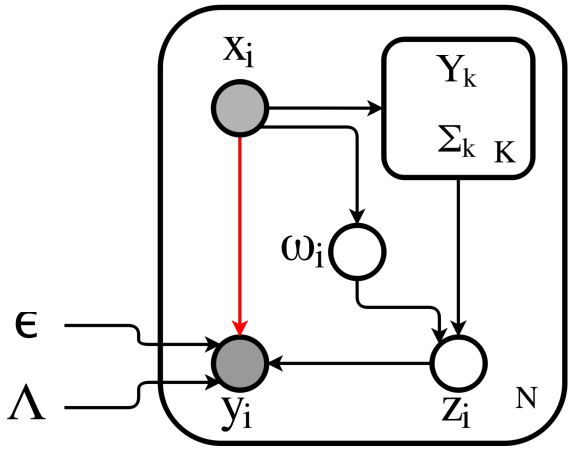


Figure: Combining MoE and MFA

Methodology-Mains

Combined Model[4]



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

Methodology-Mains
Combined Model

Inference:

- log-likelihood:

$$\ell(\theta) = \Sigma_{i=1}^{N} log(f(y_i, x_i; \theta))$$

- complete log-likelihood:

$$\ell(\theta) = \Sigma_{i=1}^{N} logf \Sigma_{\omega} \int (y_i, x_i, z_i, \omega_i; \theta) dz_i$$

- Decompose:
  $logf(y, x, z, \omega; \theta) = logf(y|z; \theta) + logf(z|x, \omega; \theta) + logf(\omega|x; \theta)$
- Parameters update through Expectation-Maximization

Methodology
Model Adaption to XML

There are two problems with the model proposed:

- Traditional EM is not scalable for learning — Online EM[3]

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

- Prediction values are in $\mathcal{R}$ while we want binary labels (0,1) — EM for logistic regression[6]

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

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

Determine β using EM — $\hat{\beta}$

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

Outline

1. Introduction
   - Multi-label Learning
   - Notation and Formulation
2. Related Work
   - Three Popular Methods
   - Embedding Method
3. Methodology
   - Methodology-Prelims
   - Methodology-Mains
4. Code Used and Contributions
5. Future Work

Code Used and Contributions

- The code by [4] is made publicly available written in R.
- 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 ~ 100.

Outline

1. Introduction
   - Multi-label Learning
   - Notation and Formulation
2. Related Work
   - Three Popular Methods
   - Embedding Method
3. Methodology
   - Methodology-Prelims
   - Methodology-Mains
4. Code Used and Contributions
5. Future Work

Future Work

- Writing and documenting the code in Python/Matlab.
- Implementing EM algorithm for logistic regression to predict binary labels.
- Rigorous empirical analysis of the model.


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.


Zoubin Ghahramani, Geoffrey E Hinton, et al.
The em algorithm for mixtures of factor analyzers.
Technical report.


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.


Angela Montanari and Cinzia Viroli.
Dimensionally reduced mixtures of regression models.

Journal of Statistical Planning and Inference, 141(5):1744–1752, 2011.

Kevin P Murphy.
Machine learning: a probabilistic perspective.
2012.

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

Thank you!

And a big thanks to Prof Piyush Rai!