Multi Instance Multi Label Learning

A Seminar Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
by

Khushboo Agarwal
(163050066)

Supervisor:
Prof. Ganesh Ramakrishnan

Department of Computer Science
Indian Institute of Technology Bombay
Mumbai 400076 (India)

1 May 2017

Acceptance Certificate

Department of Computer Science
Indian Institute of Technology, Bombay

The seminar report entitled “Multi Instance Multi Label Learning” submitted by Khushboo Agarwal (163050066) may be accepted for being evaluated.

Date: 1 May 2017

Prof. Ganesh Ramakrishnan

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this report. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Khushboo Agarwal

Date: 1 May 2017 (163050066)

Abstract

The technique of multi-instance multi-label learning has the objective of learning a model that can tag a set of data points with more than one label. It is an important research problem which addresses the problem of text classification, scene classification, relation extraction etc. We will discuss here different approaches for performing the multi-instance multi-label learning in a structured manner and figure out the effectiveness and drawbacks of each of those methods.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multi Instance Multi Label (MIML) [Zhou *et al.* (2012)] Learning is a task of associating multiple class labels to an example which is described by multiple instances. It is a very emerging research area due to its vast use in fields like Image classification, Document tagging, Text annotation etc.

1.1 Problem Statement

The problem of Multi Instance Multi Label (MIML) learning can be seen as learning of a mapping function ($f : X- > Y$) from a bag of input X_i to a set of labels Y_i, where the dataset comprises of $(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)$.
Each bag of input X_i consists of multiple instances $X_i = \{x_{i1}, x_{i2}, \ldots, x_{i,n_i}\}$, whereas the output label set consists of $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{i,l_i}\}$. Here n_i is the number of instance in bag X_i and l_i is the number of associated labels in Y_i (1.1). The objective of my seminar work was to read and get a in-depth better understanding of the different approaches which can be used to solve the MIML task.

1.2 Background

In the field of Machine Learning, learning task can be grouped into four different categories:

- **Traditional Supervised Learning:** In this problem, one instance of a task is associated with one label. So the mapping function becomes $f : x_i- > y_i$. This problem is also known as Single Instance Single Label Learning, and is a very commonly used problem.

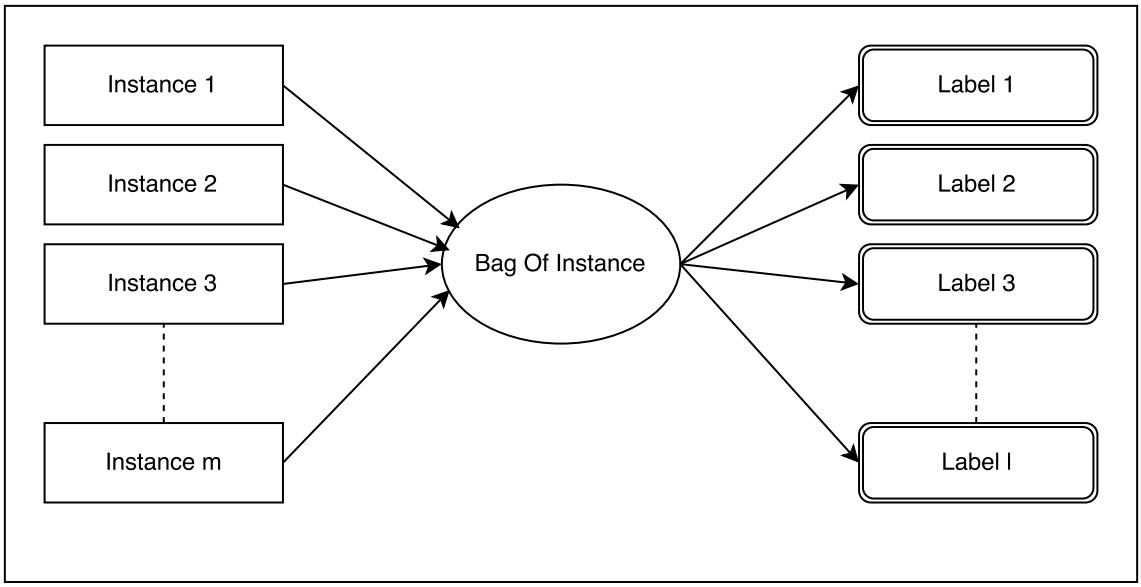

Figure 1.1: Multi Instance Multi Label Learning

- **Multi Instance Learning:** In this problem, multiple instance of a task is associated with one label. So the mapping function becomes $f : \{x_1, x_2, \ldots, x_n\} -> y_i$. This approach has its uses in image classification problem, where different parts of an image coherently point to one object.

- **Multi Label Learning:** In this problem, one instance of a task is mapped with multiple label. So the mapping function becomes $f : x_i -> \{y_1, y_2, \ldots, y_m\}$. This approach has its uses in text annotation problem, where one line of text can point to multiple contexts.

- **Multi Instance Multi Label Learning:** In this problem, one instance of a task is mapped with multiple label. So the mapping function becomes $f : \{x_1, x_2, \ldots, x_n\} -> \{y_1, y_2, \ldots, y_n\}$. This problem is our current area of interest.

1.3 Application

The problem of MIML has been applied in many areas, like Text annotation, Image classification etc. Its application can be observed to be embedded in many real life problems also. For example, if we go to a library, we can tag it with different objects, like books, librarian, study-place etc. These all labels characterize the library, but how do we get these tags? We get these tags by observing different parts(instances) of the library.

In the same way, if we see a newspaper article, one article can be labelled into different related categories. Suppose the article is about "New railway Budget", so by reading different paragraphs of the article, we can classify it into sections like politics, railway, economy etc.

So We can see, MIML problem is so closely interleaved in our daily life scenarios, which make it interesting.

1.4 Approaches

As per my preliminary study, MIML solution space can be labelled into various overlapping groups:

- **Embedding Based:** learns a low dimensional space area to get the correlation between input and output.
- **Label Specific:** learns based on property of output label in mind.
- **Label Dependency:** Based exploits the dependency between labels.
- **End-to-End extraction:** combines Entity extraction task with relation extraction to reduce error.
- **Miscellaneous:** simply explores connection between input and output using ML tools like SVM, CNN etc.

All of these will be discussed in detail in subsequent chapters.

Chapter 2

Techniques of Multi Instance Multi Label Learning

In the previous chapter, a basic overview and idea about Multi Instance Multi Label Learning was provided. In this chapter, a detailed overview of all the techniques in the solution space will be provided. (See Fig. 2.1)

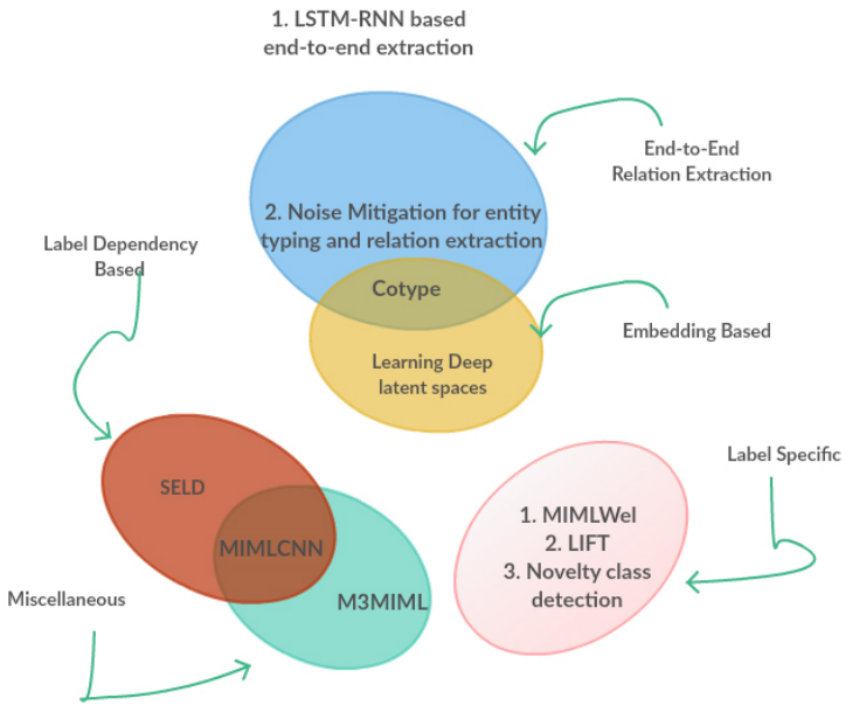

Figure 2.1: Multi Instance Multi Label Learning

To recall, the various techniques are as follows:

- Embedding Based

- Label Specific
- Label Dependency
- Entity Typing
- End-to-End extraction
- Miscellaneous

2.1 Embedding Based Approach

This technique reduces input space and label space into some low-dimensional space. The low dimensional space reveals correlation between input and output space. This type of method reduces learning time as the dimensionality is reduces. It also helps in exploiting relationship between input and output labels.

The main motivation behind using this technique is to reduce the training dimensionality to explore correlation feature between input and output.

2.2 Label Specific Approach

It uses the idea that only those features will be used for learning, which discriminates between different labels in the label space effectively.

The intuition behind this approach is that each output label carries some characteristics of their own. Suppose we want to classify a college image as classroom and non-classroom image, the materialistic characteristics like white board, biometric attendance system, benches would be crucial features. On the other hand, if we want to discriminate a student from a teacher in an image, we would look for features like age, gestures, qualifications etc features.
Features related to geographical characteristics like longitude, latitude can help in classifying two different climate conditions, cultures, but would be of no use while classifying one flower to another.

2.3 Label Dependency Based Approach

It exploits the relationship between output labels. The technique claims that apart from dependency on input characteristics, output labels depend on each other also.

For example, labels player and cricketer depend on each other. Being a cricketer preceded being a player. So if an instance is tagged with label cricketer, it will be automatically tagged as player.

2.4 End to End Extraction Based Approach

Traditionally, for relation extraction purpose(a special case of MIML), entity extraction and labelling task is considered to be two separate task.
But studies has showed end to end modelling of relation extraction, i.e., entity extraction and relation extraction task, if optimized together gives better result. This approach is useful as error in entity extraction doesn't get pipelined to the next steps, as all the steps are optimized together.

2.5 Miscellaneous Approach

Under this section, all those techniques have been grouped which can't be modelled into the above used techniques. Normally these techniques uses simple machine learning tools, like SVM, CNN etc to get results.

In the subsequent chapters, all these approaches will be dealt with and explained one-by-one.

Chapter 3

Embedding Based Approach

3.1 Motivation

With the increasing amount of data, it has become very difficult to handle those data and apply learning algorithms to infer meaningful information from them.

Embedding based approach transforms input data set and the corresponding output labels into some lower dimensional space, and then performs training step them. Using some decoders, the projected data is translated back to original space then.

So, this technique comes to rescue as it reduces the dimensionality of data by transferring them into a low-dimensional space. This low dimension helps in easy learning and analyzing the correlation between the data, hereby giving better prediction in less time. Also the need to learn individual classifier for each label type is also dissolved, thus giving better accuracy than normal one-to-one classifier.

3.2 Application

Embedding method for MIML problem can be used in cases when:

- The input data set is large.
- There is correlation between input and output data labels.

3.3 Work Done

3.3.1 Learning Deep Latent Spaces for Multi-Label Classification

Yeh *et al.* (2017) proposed a method of learning "Deep Latent Spaces" in order to solve this problem of Multi Label Classification problem. The method uses DNN-based

(Deep Neural Network) label embedding framework for multi label classification. It uses deep canonical correlation analysis (DCCA) [Andrew *et al.* (2013)] and auto-encoder to learn a feature-aware latent subspace for label embedding. At the decoding stage, for projecting input back to original subspace, it uses label-correlation aware loss functions.

Mathematically,

For a training instance $X \in \mathbb{R}^N$ and the corresponding output $Y \in \{0,1\}^l$, where l is the number of labels, the goal is to learn a model to find the relation between X and Y. Foe the purpose, three mapping functions has to be determined:

1. Feature mapping: F_x
2. Encoding function: F_e
3. Decoding function: F_d

The objective function can be formulated as:

$$\theta = \min_{F_x, F_e, F_d} \phi(F_x, F_e) + \alpha\Gamma(F_e, F_d) \tag{3.1}$$

Where,

$\phi(F_x, F_e)$ = Deep correlation-based loss function at the latent space.

$\Gamma(F_e, F_d)$ = label-correlation aware loss function at the output space.

To determine $\phi(F_x, F_e)$, the correlation-based objective function was rewritten as the following deep version:

$$\begin{aligned} \min_{F_x, F_e} \quad & \|F_x - F_e\|_2^F \\ \text{s.t.} \quad & F_x F_x^T = F_e F_e^T = I \end{aligned} \tag{3.2}$$

$\Gamma(F_e, F_d)$ is determined using a label-correlation aware loss function:

$$\begin{aligned} \Gamma(F_e, F_d) &= \sum_{i=1}^{N} E_i \\ E_i &= \frac{1}{|y_i^1||y_i^0|} \sum_{(p,q) \in y_i^1 x y_i^0} exp((F_d(F_e(x_i))^q - F_d(F_e(x_i))^p)) \end{aligned} \tag{3.3}$$

Here,

y_i^1 : set of positive labels in y_i for the instance x_i.

y_i^0 : set of the negative labels.

$F_d(F_e(x_i))^p$: p^{th} entry of the latent space input set.

Thus, minimizing the above loss function is equivalent to maximizing the prediction outputs of all positive-negative label attribute pairs, which implicitly enforces the preservation of label co-occurrence information.

The algorithm for training can be described as below:

Algorithm 1 Learning Of Deep Latent Space

Input: Feature matrix X, label matrix Y, parameter Îś, and dimension d' of the latent space

Output: F_x, F_d, and F_e

Randomly initialize F_x, F_d, and F_e

while *Coverge* **do**

- Randomly select a batch of data X and Y
- Define the loss function by (3.1)
- Perform gradient descent on F d by (3.3)
- Perform gradient descent on F x by (3.2)
- Perform gradient descent on F e by (3.2) and (3.3)

end

For prediction,$\hat{x}$ will be first transformed using encoding function F_x, then followed by the decoding function F_d for predicting the output label $\hat{y}$ (i.e., $\hat{y} = F_d(F_x(\hat{x}))$)

3.3.2 Cotype: Joint extraction of typed entities and relations with knowledge bases

Ren *et al.* (2016) proposed an end-to-end extraction method called CoType for relation extraction. The method transformed entities along with their type labels, and relation mention with their features into two different low-dimensional space. The relation labels were assigned by comparing the distance of labels in some space with the relation mentions in low-dimensional space.
More of it will be discussed in later chapters.

3.4 Technical Ingenuity

Apart from exploiting the advantage of embedding based learning, Deep latent learning method of Chih-kuan can also be extended to handle missing label problems.

3.5 Limitations

The embedding based approach shows some limitation in terms of efficiency. As input dataset is transformed into some low-dimensional space, some information is lost, making

prediction a bit less accurate.

But this limitation is overcomed by the gained in performance.

Chapter 4

Label Specific Approach

4.1 Motivation

The intuition behind this approach is that each output label carries some characteristics of their own. Either they have a common set of features to look on, or they resemble very closely to the positive instances.

Suppose we want to classify a college image as classroom and non-classroom image, the materialistic characteristics like white board, biometric attendance system, benches would be crucial features. On the other hand, if we want to discriminate a student from a teacher in an image, we would look for features like age, gestures, qualifications etc features.
Cities having similar climatic condition will resemble to other city instances for almost all features.

Keeping this intuition in mind, that binary classifier can be made more effective in case of Multi Instance Multi Label Problem, by defining a scope of instances or features to look on for each specific label.

4.2 Application

Label Specific Approach can be used in cases when:

- When the label attributes to be tagged are from different domains, and thus input set contains very non-resembling features. In that case feature selection strategy can be applied on for each labels
- When there are a few untagged but positive labels (missing labels) for an instance. So, a prototype can be defined for each label for comparison.

- When we want to look for some specific new labels which were not present in the training dataset at all.

4.3 Work Done

4.3.1 Multi-Instance Multi-Label Learning with Weak Label

Yang *et al.* (2013) proposed an MIMLWeLYang *et al.* (2013) approach for dealing with weak labels in MIML learning problems. In real life applications, labels are human-annotated in datasets. So,probability of missing any positive label is very high.Thus, all untagged labels does mean negative labels.

In this approach, bag label $Y_i = [y_{i,1}, \ldots, y_{i,L}]$, where each label $y_{i,l} = 1$ if the l^{th} label is tagged for bag X_i, and 0 otherwise.
Each bag X_i is a dimensional-vector $\phi C(X_i)$, where C are prototypes for each bag. using this mapping, each bag represented by a single feature vector, and thus classical PU learning approach for single-instance learning algorithms can be applied.
Now, learning for each label l can be done using $f_l(X) = w_l^T \phi C(X)$ where $w_l \in \mathbb{R}$. Let $W_{l,\tilde{l}}$ denote $[w_l, w_{\tilde{l}}]$ for the pair of related labels $(l, \tilde{l})$. To find $W = [w_1, \ldots, w_L]$ and output matrix $\bar{Y}$, the following optimization function is used:

$$\begin{aligned} &\min_{W,\bar{Y},C} -\eta \sum_{l=1}^{L} V(\{\bar{y}_{i,l}, X_i\}_{i=1}^m, w_l) + \sum_{1\le l,\tilde{l}\le L} R_{l,\tilde{l}} \|W_{l,\tilde{l}}\|_{2,1}^2 + \beta\theta(\{X_i\}_{i=1}^m, C) \\ &\text{s.t. } |\bar{Y}_l - Y_l|_1 / |Y_l|_1 < \epsilon \\ &\bar{y}_{i,l} = y_{i,l} \text{ if } y_{i,l} = 1, \forall l = 1, , L. \end{aligned} \tag{4.1}$$

Here, V is sum of losses on each bag, which is very computation intensive, for a large number of bags. So, class mean of bags can be used as a effective approximation for the loss function with the intuition that class means of bags for each label are separated with a large margin. Thus,

$$V(\{\bar{y}_{i,l}, X_i\}_{i=1}^m, w_l) = \frac{\sum_{i=1}^m w_l^T \phi^C(X_i) \bar{Y}_i, l}{\sum_{i=1}^m \bar{Y}_i, l} - \frac{\sum_{i=1}^m w_l^T \phi^C(X_i)(1 - \bar{Y}_i, l)}{\sum_{i=1}^m (1 - \bar{Y}_i, l)} \tag{4.2}$$

Then, $\phi^C(X_i)$ can be defined as:

$$\phi^C(X_i) = [s(X, c_1^1), \ldots, s(X, c_{r1}^1), s(X, c_1^2), \ldots, s(X, c_{rL}^L)] \tag{4.3}$$

Here, $s(X, c)$ is a similarity function. Specifically, $[c_1^l, \ldots, c_{rl}^l]$ are prototypes for the l^{th} label.

Equation 4.1 is optimized using some block-coordinate descend algortihm, which is described below:

Algorithm 2 MIMLWel

Input: $\{X_i, \hat{Y}_{i=1}^m\}, R, \eta, \beta, \epsilon$

Output: W, Y, and C

Perform clustering for the positive bags on each label to initialize prototypes C **while** *not converged to* **do**

    while *not converged to* **do**
        Fix C and $\bar{Y}$, update W
        Fix C and W, update $\bar{Y}$
    end
    Fix W and $\bar{Y}$, update C

end

4.3.2 LIFT: Multi-Label Learning with Label-Specific Features

LIFT approach, proposed by Zhang and Wu (2015), exploits label specific features to discriminate various various label specific property. It works in 2 elementary steps:

- label-specific features construction : To construct features specific to each label
- classification models induction : one for each label.

First it defines the set of positive training instances P_k as well as the set of negative training instances N_k correspond to the each labels, i.e.,

$$\begin{aligned} P_k &= \{x_i | (x_i; Y_i) \in \mathbb{D}, l_k \in Y_i\} \\ N_k &= \{x_i | (x_i; Y_i) \in \mathbb{D}, l_k \notin Y_i\} \end{aligned} \tag{4.4}$$

Then, P_k and N_k is clustered into equal number of m_k clusters, using some k-means or other clustering algorithm. Then, the input space X is mapped to a $2.m_k$ dimensional label specific feature-space using the below equation:

$$\phi_k(x) = [d(x, p_1^k); \ldots; d(x; p_{m_k}^k); d(x; n_1^k); \ldots; d(x; n_{m_k}^k)] \tag{4.5}$$

Here, d(. ; .) returns the euclidean distance between two instances.

In the second step, a family of m classification models $(g_1; g_2; \ldots; g_m)$ are learned, one for each label-specific feature.

4.3.3 Multi-instance multi-label learning in the presence of novel class instances

Another approach for detecting Novel class Instance in MIML problem was conveyed by Pham *et al.* (2015), using a discriminative probabilistic model.It determines the bag level prediction for known and novelty classes by using instance level annotation. A class $c = 0$ is assigned to the novelty class.

In this approach, two versions of output is used. Y_b denotes bag label(not including novely class), Y_b^n denotes union of all n outputs for each instance, including novelty classes. The relation between $Y_b^n \subset \{0, 1, 2, \ldots, C\}$ and $Y_b \subset \{1, 2, \ldots, C\}$ satisfies

$$p(Y_b|Y_b^n) = I(Y_b = Y_b^n) + I(Y_b \cup 0 = Y_b^n) \tag{4.6}$$

Instance level prediction can be done for a bag b as follows:

$$p(y_b i|x_{b_i}, w) = \frac{\prod_{c=0}^{C} e^{I(y_{b_i}=c)} w_c^T x_{b_i}}{\sum_{c=0}^{C} w_c^T x_{b_i}} \tag{4.7}$$

Maximum likelihood inference is used for learning the model parameters.

$$p(Y_D, X_D|w) = p(X_D) \prod_{b=1}^{N} p(Y_b|X_b, w) \tag{4.8}$$

Using law of total probability:

$$p(Y_b|X_b, w) = \sum_{y_{b_1}}^{C} \ldots \sum_{y_{b_n}}^{C} [\{I(Y_b = \cup_{j=1}^{n} y_{b_j}) + I(Y_b \cup \{0\} = \cup_{j=1}^{n} y_{b_j}) \mathrm{x} \prod_{i=1}^{n} p(y_{b_i}|x_{b_i}, w)] \tag{4.9}$$

As the log-likelihood of equation (4.8) is diificult to solve, Expectation-Maximization(EM) method is used to get the model parameters. The concept of partial bag labels were also observed to get the bag label probability with approximation in linear time only.

4.4 Technical Ingenuity

The weak label problem can be resembled with PU learning approach, but the existing PU-learning algorithms are designed mostly for single-instance data, whereas the training data in our problem are bags rather than single-instances; therefore, existing PU-learning algorithms could not be applied directly.

LIFT method gave a very simplistic approach to increase the efficiency of binary classifier, just by introducing the concept of feature selection in the problem.

Method introduced for MIML learning in the presence of novel classes can be also used to get label for each instances in the bag, thus can be proved very useful in case of image classification. The partial bag label method to get probability of a bag label prediction was very effective in the sense that complexity came down from exponential to linear in time.

4.5 Limitations

Label specific approaches described capture label specific properties well, but they completely overlook label dependency properties. Also, sometimes its difficult to identify all features relevant to each label, introducing error in learning.

Chapter 5

Label Dependency Based pproach

5.1 Motivation

In MIML problem, when simple binary classifiers are used, they overlook the dependency between labels.
In real life applications, there exists a dependency between the labels in the sense that for labels "sportsperson" and "cricketer", if a person is tagged as "cricketer", he/she can be automatically tagged as "sportsperson" also. In simple sense, its not always the case the output labels will depend only on input, but can depend on other output labels also (see Fig. 5.1).

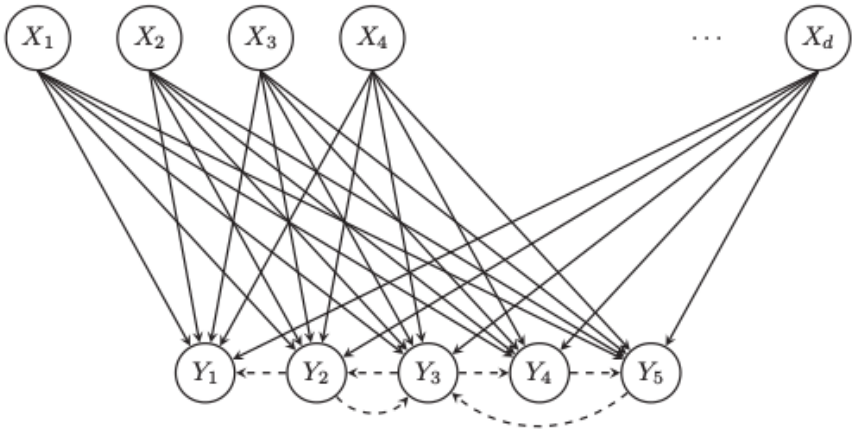

Figure 5.1: Label dependency example

So, Label dependency based approach explores correlation between different output labels to get a better feel of Multi Instance Multi Label problem.

5.2 Application

The dependency based approach has its use basically in cases when the output domain contains many similar or related labels, and basically when one label is a subset of another.

5.3 Work Done

5.3.1 Leveraging Supervised Label Dependency Propagation for Multi-label Learning

Fu *et al.* (2013) proposed an iterative model for "Leveraging Supervised Label Dependency Propagation for Multi-label Learning". In his model, the prediction of each label whenever updated is propagated to other labels also by a random restart method.

The loss function for each input instance is defined as:

$$E_i = \frac{\sum_{j=1}^{m}(y_j(x_i) - p(y_j£x_i))^2}{2m} \tag{5.1}$$

Here $p(yk£xi)$ is y_kâĂŹs probability of being x_iâĂŹs true label predicted by $fp(y_j£x_i)$ is y_kâĂŹs probability of being x_iâĂŹs true label, and $(y_j(x_i)$ is 1 when label y_k is x_i's true label, otherwise 0.

They proposed a SELD framework to iteratively update and propagate the label prediction to other labels, by a formula:

$$p_i(x) = \alpha.\frac{\sum_{j=1}^{m} p_j(x).c_{ij}}{m} + (1-\alpha).f_i(x) \tag{5.2}$$

Here, $p_i(x)$ and $f_i(x)$ are final and initial probability of y_i being the true label of x_i respectively. c_{ij} gives the degree of y_iâĂŹs dependency on y_j.
So final prediction of y_i depends on a linear combination of initial prediction obtained using some classifier and the final prediction of various labels dependent on it.
The dependency matrix c_{ij} is calculated using the below equation:

$$c_{ij} = g(y_j \rightarrow y_i) = g(\theta^T.A^{ij}) \tag{5.3}$$

θ controls the features' weight and A^{ij} is a d-length feature vector, depicting the dependency between y_j and y_i.

So the overall objective is to get a optimum value of θ to minimize the loss function by the algorithm:

Algorithm 3 Leveraging Label dependency for multi label problem

Input: Training dataset: $D = (x_i, Y_i)1in$ and learning rate lr

Output: Optimal parameter $\theta^T =< {}_1, \ldots, {}_l >$

Initialize $\theta(0)$

r =0

while *E(θ) has not converged* **do**

  shuffle the instances in D randomly

  for *i = 1 to n* **do**

    for *t = 1 to d* **do**

      $(\theta_t(r+1)) = \theta_k(r) - lr.\frac{\delta E_i(\theta(r))}{\delta \theta_t}$

      update matrix C according to Eqn.(5.3)

    end

  end

end

5.3.2 Relation Extraction with Multi-instance Multi-label Convolutional Neural Networks

Another approach called MIMLCNN also considers label dependency in relation extraction task by using shared entity-pair level representation for each label. More of it will be discussed in subsequent chapters.

5.4 Technical Ingenuity

The method proposed by Bin Fu is a very simplistic one, and uses a supervised learning optimization function, thus more accurate label dependency can be learnt from the method.

As it does not add dependency into into feature set, it has an advantage of assigning flexible weights to dependencies.
Also, it exploits label dependency in iterative manner, and updates labelsâĂŹ predictions simultaneously thus predictions converge to the global optimum more quickly.

5.5 Limitation

While considering label dependency, skewness can be major concern. As number of labels are generally much less than the number of features, so learning functions will be more biased on features than labels, as label dependencies are very less.

Chapter 6

End-to-End-Extraction

6.1 Motivation

End-to-end relation extraction is generally used for relation extraction purpose, in which the whole work from entity identification to relation labelling is done jointly, and under one model.
Since in real life applications, entity identification is done from a KB or are handcrafted . It increases the chance of missing out few labels.

Moreover, if entity, relation extraction tasks are interrelated. If done, separately, error in one step may flow to another.

So joint extraction is useful in the sense that it increases the performance of relation extraction task by modelling entity extraction and relation extraction together at training time only.

6.2 Application

End-to-end extraction can be used when there are error in the entity extraction or feature identification process, which, if propagated to subsequent stages can lead to degraded performance.

6.3 Work Done

6.3.1 Noise Mitigation for Neural Entity Typing and Relation Extraction

Yaghoobzadeh *et al.* (2016) proposed a model for noise mitigation in case of entity typing and relation extraction. It believed on the concept that [Yaghoobzadeh and

Schütze (2017)] entity typing in relation extraction process can reduce the prediction error compared to traditional pipe-lined process. The intuition of the model was that entity typing can also be modelled as a multi instance multi label problem, as each entity as several context mentions, thus various instances. Also, they can belong to various types, thus a multi label problem also.

Thus, an entity e is represented by a set of q contexts $\{c_1, c_2, \ldots, c_q\}$. If $P(t|c_i)$ represents probability that context has type e, then $P(t|e)$ is the probability that entity e belongs to type e, and can be calculated by either by taking averages(MIML-AVG), max(MIML-MAX) or a combination of both(MIML-MAX-AVG) over all $P(t|c_i)$.

Then a type-aware relation extraction is performed, by embedding entities along with their type with the input feature set, thus forming a extended context representation.

6.3.2 End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures

Miwa and Bansal (2016) proposed a method for end-to-end relation extraction using LSTMs on Sequences and Tree Structures. It detects entities and detects relation between them, and then jointly update the model parameters for both entity and relation labels. It creates two layer based on LSTM-RNN structure, one to realize word-sequence, other for dependency tree.

The sequence layer represents sentential context information. It uses bi-directional LSTM [Graves *et al.* (2013)] for sequence representation. The input vector is a concatenation of word embedding and POS embedding vector, i.e., $x_t = [v_t^{(w)}; v_t^{(p)}]$. Output vector consists of concatenation of hidden state vector of each direction for each word, i.e., $s_t = [\underset{h_t}{\rightarrow}; \underset{h_t}{\leftarrow}]$.

Entity detection is done in a left-to-right manner using a two-layered RNN. Previous entity label is used concatenated with the LSTM output of that word, to account for label dependencies.This layer assigns a tag(BILOU scheme) to each entity along with the usual type of it.

The dependency layer tries to find relation between between two entities by finding a shortest path between them in the dependency tree. It is assumed that shortest path contains important information for relation extraction. Dependency layer is stacked above sequence layer to capture both sequence information and dependency information in the output. The input to the dependency layer is a vector containing concatenation of its corresponding hidden state vectors s_t in the sequence layer, dependency type embedding $v_t^{(d)}$ (denotes the type of dependency to the parent), and label embedding $v_t^{(e)}$, i.e., $x_t = [s_t; v_t^{(d)}; v_t^{(e)}]$

For relational extraction, candidate entity pair is formed by taking a combination of all L or U tagged entities in BILOU scheme. for each candidate pair, a candidate relation vector is formed using bidirectional LSTM. The output of the LSTM is the concatenation of lowest common ancestor of the target word pair p in the top LSTM, and the hidden state vectors of the two LSTM units representing the first and second target words in the top-down LSTM-RNN. This $d_p = [\uparrow h_{pA}; \downarrow h_{p1}; \downarrow h_{p2}]$, is given an input to two layered RNN for detecting the relation labels.

6.3.3 CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases

Cotype method proposed by Ren *et al.* (2016) performed joint extraction of entities and relations with KB. In this method, entity mention describes the token span which represents an entity, and relation span represents is a ordered pair of entity mentions. Clearly, all entity mentions and relation mentions can't be linked to some entities or relations in KB. Training data is formed by taking only linkable entities and relation mentions, with their typesets.

For generating the entity mentions, the text is segmented into various phrase, and candidate entity set is chosen by taking segment quality S_d for document d into account, which can be calculated as follows:

$$\sum_{d}^{D} \log p(S_d, d) = \sum_{d}^{D} \sum_{t=1}^{|} d| \log p(b_{t+1}^{(d)}, c^{(d)} | b_t^{(d)}) \\ p(b_{t+1}, c|b_t) = p(b_{t+1} b_t).p(c|b_{t+1} b_t).Q(c) \tag{6.1}$$

Here $p(b_{t+1}, c|b_t)$ is the probability that segment c, ending on index b_{t+1} is a good entity mention.
Q(c) is equally weighted combination of the phrase quality score and POS pattern quality score, calculated using Random forest classifier.

For each entity pair m_a, m_b generated in a sentence s, two relation mentions $z_1 = (m_a, m_b, s)$ and $z_2 = (m_b, m_a, s)$ are formed.

For syntax and distributional characteristics, lexical features of entities and its contexts from POS-tagged corpus is extracted for entitis and relation mention(F_m and F_z respectively).

Relation mentions are linked with type labels and features in a d-dimensional vector space. The solution propose a novel global objective, which extends a margin-based rank loss to model noisy mention-type associations and leverages the second-order proximity idea to model corpus-level mention-feature co-occurrences.

It assumes that two relation mentions sharing many text features (i.e., with similar distribution over the set of text features F_m) likely have similar relation types; and text features co-occurring with many relation mentions in the corpus tend to represent close type semantics. Using these two assumptions, a loss function is modelled.
Same steps are repeated for entity mentions.
It not only embed entities with their types, but also with text features in a d-dimensional vector space.

For representing entity relation hypothesis, "translating operation" is used, which follows the assumption that embedding vector of m_1 should be a nearest neighbor of the embedding vector of m_2 plus the embedding vector of relation mention z. Error function is written using: $\tau(z) = \|m_1 + zm_2\|_2^2$

Loss function calculated from entity and relation embedding, and their interaction is captured independently using $\{O_z, O_m, O_{zm}\}$, and are minimized using a joint optimized function containing a combination of all.

6.4 Technical Ingenuity

LSTM-RNN based approach incorporates two enhancements over traditional end-to-end relation extraction model by using entity pretraining, which pretrains the entity model, and scheduled sampling, which replaces (unreliable) predicted labels with gold labels in a certain probability. These enhancements increase the performance of entity detection in early stages of training, also thus improving relation classification task.

Moreover, this model builds feature based on both sequence and parse-tree, and as bidirectional LSTMs are used, dependency information is captured from both direction.

Cotype based framework captures efficiently the constraints for each level i.e., entity extraction, relation mention extraction, and the cross constraints between their joint modelling. It is also domain independent.
Cotype based model embeds not only types but also text features with each entity. The loss function modelled for entity typing step captures the distributional as well as semantic information.

Noise Mitigation for Neural Entity Typing and Relation Extraction paper showed that probabilistic calculation is better than discrete prediction.
Also it showed how MIML problem can be applied to entity typing step. All previous papers used to consider entity typing as distant supervision problem only. It helped to reduce the noise due to distant supervision.

6.5 Limitation

LSTM-RNN based method for relation extraction works for a single instance problem only.

Cotype based framework, though used a hierarchy for types, it can't find the correlation between them. As it depends on pre-trained POS-taggers, extending it to different languages become difficult.

Chapter 7

Miscellaneous

7.1 Motivation

This chapter groups different techniques for classification which doesn't use any specific method but just employs basic ML tools like, SVM, CNN to get the dependency between input and output.

7.2 Work Done

7.2.1 Relation Extraction with Multi-instance Multi-label Convolutional Neural Networks

Jiang *et al.* (????) introduced a a multi-instance multi-label convolutional neural network architecture. The model takes two entity pairs $< e_1, e_2 >$ and input sentences. For each sentence, it creates an input representation by mapping each word of a sentence to a vector, containing concatenation of word embedding (dimension d_s) and position features from each entity(dimension d_p), thus input dimension $d_s = d_w + 2.d_p$.

Features are extracted from input matrix using convolution window(W) of size w_cxd_s. The convolution window is slided down the sentence to get a feature map $c = [c_1, c_2, \ldots, c_{h-w_c+1}]$.The process is repeated n times with different W to get n features.

Then a piecewise max-pooing is performed to capture the 3 most important features from each feature maps, thus getting $3n$ features in total for each sentences.

After feature generation, to detect relation labels, a cross-sentence max pooling is performed. The idea behind this step was inferred from an assumption that *"A relation holding between two entities can be either expressed explicitly or inferred implicitly from*

all sentences that mention these two entities".

So instead of, sentence layer representation, entity pair level relation extraction is performed. For this, it aggregates each components of the sentences into one components, thus creating three entity-pair level components by taking a max-pool on them.

If, $p_i^(j)$ is the i^{th} component of sentence j, then entity-pair level representation $g = [g_1, g_2, \ldots, g_{3n}]$ can be constructed as:

$$g_i = \max(p_i, p_i, \ldots, p_i) \tag{7.1}$$

Relation modelling is done by applying a sigmoid function on each entity-level representation for each label(i), by the equation

$$p(i|M, \theta) = \frac{1}{1 + e^{W_i \cdot g + b_i}} \tag{7.2}$$

7.2.2 M^3MIML : A Maximum Margin Method for Multi-Instance Multi-Label Learning

Zhang and Zhou (2008) proposed a method called M^3MIML, which uses maximum margin method for solving MIML problem. This model assumes that the systemâĂŹs output for dataset (X_i, Y_i) for the l^{th} class is determined by the maximum prediction of X_i âĂŹs instances with respect to (w_l, b_l), i.e.,

$$Y = \{l|\max_{x \in X}(< w_l, x_i > +b_l)0, l \in Y\} \tag{7.3}$$

So, the margin for a classification system can be expressed as minimum margin of a dataset over all classes as:

$$\min_{l \in Y} \frac{Y_i(l).\max_{x \in X}(< w_l, x_i > +b_l)}{||w_l||} \tag{7.4}$$

For the whole dataset equation can be rewritten as:

$$\min_{1 \le i \le n} \min_{l \in Y} \frac{Y_i(l).\max_{x \in X}(< w_l, x_i > +b_l)}{||w_l||} \tag{7.5}$$

Since, minimum margin value is assumed to be 1 , under the assumption that all labels can be properly classified by the system

$$\Delta S = \min_{l \in Y} \frac{1}{||w_l||} \tag{7.6}$$

So the optimization function can be rewritten as:

$$\begin{gathered} \min_{\{(w_l, b_l)|l \in Y\}} \frac{1}{2} \max_{l \in Y} ||w_l||^2 \\ \text{subject to: } \forall i \in \{1, \ldots N\}, l \in Y \text{s.t.} \\ \max_{x \in X_i}(< w_l, x_i > +b_l)1, \text{ if } l \in y_i \\ \forall x \in X_i : (< w_l, x_i > +b_l)1, \text{ if } l \in \bar{y}_i \end{gathered} \tag{7.7}$$

Since, summation over all datasets is greater than maximum value, and average is less than or equal to the maximum value, the function can be rewritten as:

$$\begin{aligned} &\min_{\{(w_l,b_l)|l\in Y\}} \frac{1}{2}\max_{l\in Y} \sum_{l=1}^{m} \|w_l\|^2 \\ &\text{subject to: } \forall i \in \{1,\dots N\}, l \in Y \text{s.t.} \\ &\frac{\sum_{j=1}^{n_i}(< w_l, x_i > +b_l)}{n_i} 1, \text{ if } l \in y_i \\ &\forall x \in X_i : (< w_l, x_i > +b_l)1, \text{ if } l \in \bar{y_i} \end{aligned} \tag{7.8}$$

The equation is solved for optimal value of W by getting the dual value of the equation.

7.3 Technical Ingenuity

Traditional relation extraction algorithm extracts feature from contexts using various NLP algorithms or are hand-crafted, and thus have errors. The errors become more severe for long sentences. Faulty errors shows degraded performance in distant supervision task due to error propagation. The model formed by using CNN automatically extracts task specific features from context, thus reduces the error probability.
Moreover, unlike other CNN models [Zeng *et al.* (2014)], it trains on entity pair-level representation, rather than sentence pair level representation, thus achieving multi instance target.

So MIMLCNN can be used when feature set is unknown for relation extraction purpose.

M^3MIML is an extended version of previous MIMLBOOST, and MIMLSVM. The methods at that time used degenerated method to solve MIML problem, i.e., by converting MIML problem to either single instance or to single label, and then combining to get the MIML output. Those method had a serious issue that many of the information used to get lost in the degenerating process. That time, M^3MIML brought an advantage by directly introducing relation between input and output.

7.4 Limitation

One serious issue with MIMLCNN method is that it requires sentences to be of same length, but can be tackled by padding. As CNN training time is very high, and its complete dependence on CNN, make it less applicable to larger datasets.

M^3MIML uses a QP optimization problem, which gets very difficult to solve, and is a very basic method method. Lots of advancement has been done till now.

Chapter 8

Conclusion

The seminar work was concluded with a overall study of MIML problem and different approaches of solving it. The summary of all work is presented below:

Method	Pros	Cons
Embedding Based Approach:	Identifies input output correlation	Loss of information
Label Specific Approach:	Learns label specific features	Increases complexity.
Label Dependency Approach:	Identifies label dependency	Skewness problem
End-to-End Extraction:	Reduces error due to pipelining of input	Complex Model

Table 8.1: Comparision among the methods

So every system has some positive points as well as drawbacks, Correct model is chosen as per the requirements.

References

Andrew, G., Arora, R., Bilmes, J., and Livescu, K., 2013, "Deep canonical correlation analysis," in *International Conference on Machine Learning*, pp. 1247–1255.

Fu, B., Xu, G., Wang, Z., and Cao, L., 2013, "Leveraging supervised label dependency propagation for multi-label learning," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on* (IEEE). pp. 1061–1066.

Graves, A., Mohamed, A.-r., and Hinton, G., 2013, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on* (IEEE). pp. 6645–6649.

Jiang, X., Wang, Q., Li, P., and Wang, B., ????, "Relation extraction with multi-instance multi-label convolutional neural networks,"

Miwa, M., and Bansal, M., 2016, "End-to-end relation extraction using lstms on sequences and tree structures," *arXiv preprint arXiv:1601.00770*

Pham, A. T., Raich, R., Fern, X. Z., and Arriaga, J. P., 2015, "Multi-instance multi-label learning in the presence of novel class instances.." in *ICML*, pp. 2427–2435.

Ren, X., Wu, Z., He, W., Qu, M., Voss, C. R., Ji, H., Abdelzaher, T. F., and Han, J., 2016, "Cotype: Joint extraction of typed entities and relations with knowledge bases," *arXiv preprint arXiv:1610.08763*

Yaghoobzadeh, Y., Adel, H., and Schütze, H., 2016, "Noise mitigation for neural entity typing and relation extraction," *arXiv preprint arXiv:1612.07495*

Yaghoobzadeh, Y., and Schütze, H., 2017, "Multi-level representations for fine-grained typing of knowledge base entities," *arXiv preprint arXiv:1701.02025*

Yang, S.-J., Jiang, Y., and Zhou, Z.-H., 2013, "Multi-instance multi-label learning with weak label," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence* (AAAI Press). pp. 1862–1868.

Yeh, C.-K., Wu, W.-C., Ko, W.-J., and Wang, Y.-C. F., 2017, "Learning deep latent spaces for multi-label classification,"

Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., *et al.*, 2014, "Relation classification via convolutional deep neural network.." in *COLING*, pp. 2335–2344.

Zhang, M.-L., and Wu, L., 2015, "Lift: Multi-label learning with label-specific features," *IEEE transactions on pattern analysis and machine intelligence* **37**, 107–120.

Zhang, M.-L., and Zhou, Z.-H., 2008, "M3miml: A maximum margin method for multi-instance multi-label learning," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (IEEE). pp. 688–697.

Zhou, Z.-H., Zhang, M.-L., Huang, S.-J., and Li, Y.-F., 2012, "Multi-instance multi-label learning," *Artificial Intelligence* **176**, 2291–2320.

Acknowledgements

This report is a summary of selected readings undertaken and solved problems while working under the guidance of Prof. Ganesh Ramakrishnan on "Multi Instance Multi Label Learning". I would like to thank him for the guidance. It was immensely helpful in gaining the understanding required for writing this report. Also I would like to thank Ankith MS and Suhit Sinha for all the help throughout this learning experience.

Khushboo Agarwal
IIT Bombay
1 May 2017