# CPTS570 Project

Chandan Dhal

December 7, 2019

## 1 Abstract

**Target Paper:** Large-Scale Bayesian Multi-Label Learning via Topic-Based Label Embeddings.

This project is an extension of Homework 3 Fortune Cookie text classifier in "Bag of words" setting, where the Label is a vector instead of taking binary values. The report is organized as follows, section II describes the complexity of this setting. Section III describes the target paper classification algorithm, Bayesian Multi-label Learning via Positive-Labels [BMLPL]. Section IV summarizes the results of BMLPL on a real-word Dataset. Section V summarizes the proposed abstract of Multi-Label classification.

## 2 Introduction:

During the lectures, we have have seen various Machine-learning settings, where the task was to assign a new example feature into 2 different classes (Binary Label classifier). But complexity increases if the task is to assign more than 2 classes (Multi-Labels classifier). We have also discussed two Multi-Labels classifier abstracts, "one-vs-one" and "one-vs-all" for Multi-Labels classifier.

Multi-Labels classifiers faces a lot of challenges in applications like image/webpage annotation, medical coding, where the Label Vector and also the Feature-vectors are very large. Any of those above mentioned abstract could require a lot of computation power to learn all the respective label weight vector for classification.

The target paper presents a scalable , fully Bayesian framework for Multi-Label Classification. The framework is based on the reduction of the dimensionality of the label space. The authors claimed the following key advantages of this framework over others which are also based same idea:

1. Computational cost of training scales in the number of ones in the label matrix.

2. likelihood model for the binary labels, based on a Bernoulli-Poisson link, more realistically models the extreme sparsity of the label matrix as compared to the commonly employed logistic/probit link

3. model is more interpretable - embeddings naturally correspond to topics where each topic is a distribution over labels.

4. In addition to the modeling flexibility that leads to a robust, interpretrable, and scalable model, our framework enjoys full local conjugacy, which allows us to develop simple Gibbs sampling, as well as an Expectation Maximization (EM) algorithm for the proposed model, both of which are simple to implement in practice (and amenable for parallelization).

This report only describes the Gibbs sampling method to generate the model, but the submitted code file can extend to EM model. Section 3 discusses more on this, as well as, an expolaratory "one-vs-one" approach.

# 3 Bayesian Multi-label Learning via Positive-Labels [BMLPL]

Often, in multi-label learning problems, many of the labels tend to be correlated with each other. To leverage the label correlations and also handle the possibly massive number of labels, a common approach is to reduce the dimensionality of the label space.

**Given:**

Label matrix, $Y \epsilon (0,1)^{LxN}$

Task, predict label vector $y_* \epsilon (0,1)^L$ given test example $x_* \epsilon \mathbb{R}^D$

The binary label vector is modeled as below equations.

The binary label vector of n-th example, $y_n$

$$y_n = \mathbb{1}(m_n \geq 1) \tag{1}$$

where $m_n = [m_{1n}, m_{2n}, ...m_{Ln}]$ belongs to latent count of vector size L, and a functional $f$ such that $y_{ln} = 1(m_{ln} \geq 1)$

$$m_n \sim \textbf{Poisson } \lambda_n \tag{2}$$

$$\lambda_n = V u_n \tag{3}$$

$$y_n = f(V u_n) \tag{4}$$

Here, $V$ can be thought of as topics over Labels and $u_n$ can be thought of embedding of label vector. Thus $u_n$ determines how actively each topics are in the n-th example. This is equivalent to a low-rank assumption on the label matrix, where K is scalable.

$V = [v_1 ... v_K] \epsilon \mathbb{R}_+^{LxK}$

$U = [u_1 ... u_N] \epsilon \mathbb{R}_+^{KxN}$

Thus, the modeling parameter using this frame-work is summarized below equations

$$v_k \sim Dirichlet(\eta \mathbf{1}_L) \tag{5}$$

$$u_{kn} \sim Gamma(r_k, P_{kn}(1 - p_{kn})^{-1}) \tag{6}$$

$$p_{kn} = \sigma(w_k^T x_n) \tag{7}$$

$$w_k \sim Nor(0, \Gamma \tag{8}$$

**Key-Idea:** Since columns of $V$ are Dirichlet Drawn, they correspond to topics over labels. The dependence of the label embedding $u_n$ on the feature vector $x_n$ is achieved by making the scale parameter of the gamma prior on $u_n$ depend on $P_{kn}$ which in turn depends on features $x_n$ via regression weight $W$. Figure 1 depicts the BMLPL framework.

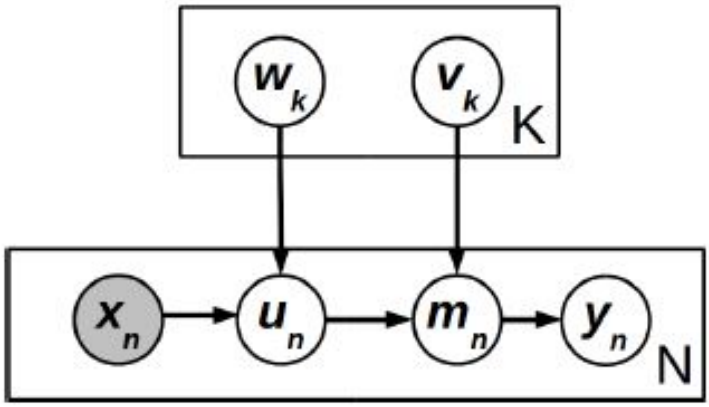


Figure 1: BMLPL

**Observations**

1. Marginalizing $m_n$ from Equation 1, leads to $y_n \sim Bernoulli(1-exp(-\lambda_n))$

2. For bernoulli-poisson likelihood for binary labels, the conditional posterior of the latent vector $m_n$, is

$$Pr(m_n|y_n, V, u_n) \sim y_n \odot Poisson_+(Vu_n) \tag{9}$$

   $Poisson_+$positive zero truncated Poisson distribution and $\odot$ is the element wise product.

   Thus, label with 0 entry will sure have corresponding latent vector $m_n$ also 0.

3. Bernoulli-Poisson likelihood is also a more realistic model for highly sparse binary data as compared to the commonly used logistic/probit likelihood.

Bernoulli-Poisson link will encourage a much fewer number of nonzeros in the observed data as compared to the number of zeros. On the other hand, a logistic and probit approach both 0 and 1 at the same rate, and therefore cannot model the sparsity/skewness of the label matrix like the Bernoulli-Poisson link.

4. Gibbs sampling is commonly used as a means of statistical inference, especially Bayesian inference. It is a randomized algorithm (i.e. an algorithm that makes use of random numbers), and is an alternative to deterministic algorithms for statistical inference such as the expectation-maximization algorithm (EM).

# 4 Application of BMLPL

**Application Paper:** Confronting Data Sparsity to identify potential sources of Zika virus spillover infection among primates

Since BMLPL is expected to perform better on highly sparse Data set. This section summarizes how the proposed framework was implemented in the **Application Paper**. The inference section of the target paper is omitted in the report. The BMLPL framework in this framework is based on Gibbs Sampling method. The application paper has an open access github account for the results generated in the paper. This section is based on their R code. The figures 2-4 (snaps from Target Paper) summaries the Gibbs sampling method for estimating the BMLPL model parameters.

The application paper collected samples of 365 spicies with a label vector of size 7. The labels are flavlabel, Zlabel, Zdiv, Order, family, Genus, Species. Where later topics have arbitrarily many possibility. The output csv file Imputed1 was used to analyze the proposed Multi-Class classifier.

Further, an exploratory results on this dataset was implemented using Naive Bayes Gaussian classifier. I considered only one label for e.x. favlabel as the binary classification training label and compared the application paper with this exploratory results.

**Conclusion of Application Paper:** BMLPL was implemented on a large data set which is also highlly sparse and also assumes the Label vector structure as "Topics over Labels". This report only analyzes one key findings of the paper. Using BMLPL they were able to detect some species [Species 88 Cebus Albifrons] who have been predicted as positive Zlabel, but actual recorded data is negative. Figure 5 summarizes Table 2 from the application paper.

**Remark:** Only one of the Output Dataset is used for further discussions. The submitted zip files contains the R code for BMLPL framework. These resources are available in the Github link mentioned in the application paper.

**Sampling V:** Using Eq. 11 and the Dirichlet-multinomial conjugacy, each column of $\mathbf{V} \in \mathbb{R}_+^{L \times K}$ can be sampled as

$$\boldsymbol{v}_k \sim \text{Dirichlet}(\eta + m_{1k}, \ldots, \eta + m_{Lk}) \tag{12}$$

where $m_{lk} = \sum_n m_{lnk}, \forall l = 1, \ldots, L$.

**Sampling U:** Using the gamma-Poisson conjugacy, each entry of $\mathbf{U} \in \mathbb{R}_+^{K \times N}$ can be sampled as

$$u_{kn} \sim \text{Gamma}(r_k + m_{kn}, p_{kn}) \tag{13}$$

where $m_{kn} = \sum_l m_{lnk}$ and $p_{kn} = \sigma(\boldsymbol{w}_k^\top \boldsymbol{x}_n)$.

**Sampling W:** Since $m_{kn} = \sum_l m_{lnk}$ and $m_{lnk} \sim \text{Poisson}_+(v_{lk} u_{kn})$, $p(m_{kn}|u_{kn})$ is also Poisson. Further, since $p(u_{kn}|r, p_{kn})$ is gamma, we can integrate out $u_{kn}$ from $p(m_{kn}|u_{kn})$ which gives

$$m_{kn} = \text{NegBin}(r_k, p_{kn})$$

where NegBin(.,.) denotes the negative Binomial distribution.

Figure 2: Sampling V,U, M

$$\omega_{kn} \sim \text{PG}(m_{kn} + r_k, \boldsymbol{w}_k^\top \boldsymbol{x}_n)$$

where PG(.,.) denotes the Pólya-Gamma distribution [17].

Given these PG variables, the posterior distribution of $\boldsymbol{w}_k$ is Gaussian $\mathcal{N}or(\mu_{w_k}, \Sigma_{w_k})$ where

$$\begin{aligned} \Sigma_{w_k} &= (\mathbf{X}\Omega_k\mathbf{X}^\top + \Gamma^{-1})^{-1} \\ \mu_{w_k} &= \Sigma_{w_k}\mathbf{X}\kappa_k \end{aligned}$$

where $\Omega_k = \text{diag}(\omega_{k1}, \ldots, \omega_{kN})$ and $\kappa_k = [(m_{k1} - r_k)/2, \ldots, (m_{kN} - r_k)/2]^\top$.

Figure 3: Sampling W from M

$$p(y_{l*} = 1|\boldsymbol{x}_*) = 1 - \prod_{k=1}^{K} \frac{1}{[V_{lk} \exp(\boldsymbol{w}_k^\top \boldsymbol{x}_*) + 1]^{r_k}}$$

Figure 4: BMLPL framework Label Prediction

# 5 Results interpretation

## 5.1 Exploratory Experiment results:

Considered a "one-vs-one" approach to design prediction models for each labels flavlabel, Zlabel and Zdiv. Rest of the label topics are not numeric, due to the time constraints a naive-gaussian classifier are trained on these three label topics. At the end of this section we will compare the results obtained by Exploratory and BMLML for a particular example.

Fgure 6, 7 and 8 depicts the naive-gaussian classifier Training accuracy when only flavlabel, Zlabel and Zdiv label topics are considered respectively.

**Table 2**
New world primate species whose risk scores for testing positive for Zika virus ($ZIKV^+$) were above the 90th percentile, and other mosquito-borne flaviviruses for which each species has tested positively (YFV = yellow fever virus; SLEV = St. Louis encephalitis virus; Undetected = the primate species is currently unknown to be positive for any mosquito-borne flaviviruses).

| Species | Status | Percentile risk $ZIKV^+$ |
|---|---|---|
| *Cebus apella* | YFV+ | 99.7 |
| *Cebus albifrons* | Undetected | 97.3 |
| *Alouatta seniculus* | YFV+ | 95.9 |
| *Alouatta caraya* | YFV+, SLEV+ | 94.2 |
| *Saimiri boliviensis* | Undetected | 92.9 |
| *Cebus capucinus* | Undetected | 90.7 |

Figure 5: Table 2: Application Paper

```
In [3]:
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings('ignore')

MyData = pd.read_csv("/Users/Chandan/Documents/CPTS 570/Project/Predict-zika-res-master/Outputs/Imputed_data_1.csv", header=N
Y_train = MyData[MyData.columns[0:1]].values
X_train = MyData[MyData.columns[7:]].values



scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
X_train = scaling.transform(X_train)

gnb = GaussianNB()
GnBTraining_acc = []
GnBTest_acc = []


count = 0
for i in range(len(X_train)):
    y = gnb.fit(X_train,Y_train).predict([X_train[i,:]])                # Laplace Smoothing by default
    if y != Y_train[i]:
        count = count + 1
GnBTraining_acc.append((len(X_train)-count)/len(X_train)*100)

GnBTraining_acc

Out[3]: [79.45205479452055]
```

Figure 6: Flavlabel as only Training label

Each of the individual classifier prediction for species 88 is depicted in Figure 9. Out[20],Out[22],Out[24] represents the predicted value for labels Flavlabel, Zlabel, Zdiv respectively.

**Observation 1:** Species 88 was predicted correctly by all individual classifiers based on Scikit Naive Bayes. Since the nature of Zika virus spillover depends on the geographical location, surrounding natural habitat conditions. The scalable $K$ paramater can be predict the **"Risk Factor"** on other species who are likely to get infected from Zika virus spillover. These characteristics were able to be recorded by using BMLPL or any "Topics over labels" Label

```
In [2]:
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings('ignore')

MyData = pd.read_csv("/Users/Chandan/Documents/CPTS 570/Project/Predict-zika-res-master/Outputs/Imputed_data_1.csv", header=
Y_train = MyData[MyData.columns[1:2]].values
X_train = MyData[MyData.columns[7:]].values



scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
X_train = scaling.transform(X_train)

gnb = GaussianNB()
GnBTraining_acc = []
GnBTest_acc = []


count = 0
for i in range(len(X_train)):
    y = gnb.fit(X_train,Y_train).predict([X_train[i,:]])                # Laplace Smoothing by default
    if y != Y_train[i]:
        count = count + 1
GnBTraining_acc.append((len(X_train)-count)/len(X_train)*100)

GnBTraining_acc

Out[2]: [70.68493150684931]
```

Figure 7: Zlabel as only Training label

```
In [4]:
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings('ignore')

MyData = pd.read_csv("/Users/Chandan/Documents/CPTS 570/Project/Predict-zika-res-master/Outputs/Imputed_data_1.csv", header=
Y_train = MyData[MyData.columns[2:3]].values
X_train = MyData[MyData.columns[7:]].values



scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
X_train = scaling.transform(X_train)

gnb = GaussianNB()
GnBTraining_acc = []
GnBTest_acc = []


count = 0
for i in range(len(X_train)):
    y = gnb.fit(X_train,Y_train).predict([X_train[i,:]])                # Laplace Smoothing by default
    if y != Y_train[i]:
        count = count + 1
GnBTraining_acc.append((len(X_train)-count)/len(X_train)*100)

GnBTraining_acc
```

Figure 8: Zdiv as only Training label

classifier.

## 5.2 BMLPL Gibbs Sampling Results:

The inference model based on Gibbs Sampling took lot of time to estimate the parameters for $W$. Later subsection we discuss how EM algorithm can achieve that computation much faster than Gibbs Sample. Figure 10 depicts the BMLPL-Gibbs sampling framework prediction.

**Observation 2:** From figure 5, the authors reported that species 88 (Cebus albifrons), species 90 (cebus capucinus), species 336 (Saimiri bolivienis) detected as positive Zlabel using BMLPL framework. This prediction model is based

```
In [20]: y = gnb.fit(X_train,Y_train).predict([X_train[79,:]])
         y
Out[20]: array([0], dtype=int64)

In [22]: y = gnb.fit(X_train,Y_train).predict([X_train[79,:]])
         y
Out[22]: array([0], dtype=int64)

In [24]: y = gnb.fit(X_train,Y_train).predict([X_train[79,:]])
         y
Out[24]: array([0], dtype=int64)
```

Figure 9: Species 88 Prediction "one-vs-one"

```
> model = bmlml.Gibbs(Y1, X1, K, nsamp=50, maxit=1e1, tol=1e-3)
> newX = as.matrix(MyData[81,8:45])
> Ypred = predict.bmlml(model, newX, transform="scale", standardize=NULL)
> cat("The Zlabel prediction is ",Ypred[1,1])
The Zlabel prediction is  1
> newX = as.matrix(MyData[83,8:45])
> Ypred = predict.bmlml(model, newX, transform="scale", standardize=NULL)
> cat("The Zlabel prediction is ",Ypred[1,1])
The Zlabel prediction is  1
> newX = as.matrix(MyData[325,8:45])
> Ypred = predict.bmlml(model, newX, transform="scale", standardize=NULL)
> cat("The Zlabel prediction is ",Ypred[1,1])
The Zlabel prediction is  1
```

Figure 10: BMLPL Zlabel prediction for Cebus albifrons,cebus capucinus,Saimiri bolivienis

on the BMLPL frame inference mode which assumes that "Topic-over-labels" prediction structure.

**Remark: Hyper-parameters** The BMLPL framework requires tuning for scalable parameter "$K$", which corresponds to size of the matrix parameters $V$ and $U$. From equation 4, we defined $K$ as topics over labels , i.e., how many of the labels are correlated. For the application database, I assumed 2 labels are correlated.

For Gibbs Sampling, we need to specify the number of samples to run the inference model. The complexity of this method depends on the number of non-zeros label entries. Although not described in this report, BMLPL-EM framework is much faster than BMLPL-Gibbs sampling framework. This parameter should be in order of $1e3$ for better inference results parameters.

**Note:** The github BMLPL R source code is not compatible with window versions. The following packages were changed from the actual source code to run on windows platform. These might have affected the results. The submitted

code works for windows implementation.

1. Omega.k = pgdraw( b, c) for prediction model
2. b(is.nan(b)) = 1, if using pgdraw() then b $\geq$ c and b $\geq$ 1

## 5.3 BMLPL Computation Complexity:

The computational cost for Predicting Label Vector $Y$ for the n-th example depends on determining the latent vector $m_{ln} = (m_{l1n}, .., m_{lKn})$. Each latent vector element takes O($K$) time, thus the computational cost of computing Y is O($nnz(Y)K$). It could very efficient if label Vector $Y$ is known to have few non-zeros, which is observed in most real-world multi-label cases.

Estimating the parameter $V, U$ are cheap to generate using Gibbs and EM method because the Polya-Gamma expectations are available in closed form and can be easily computed. The most complex step is estimating the parameters $W$ which takes O($KD^3$) using Gibbs sampling. This computation hurdle can be avoided by using EM algorithm, which is not analyzed in this report. Since both Gibbs sampling and EM algorithm inferences are computing the two independent parameters, the two algorithms could be extended in parallized/block update to take advantage of both algorithms.

## 5.4 Comparison with other algorithms:

The target paper evaluates the BMLPL on four Bench-mark multi-Label Data set, namely, bibtex, delicious, compphys, eurlex. The figure below summarizes the Data sets used in the paper. The following state-of-the-art methods were compared against BMLPL, which are also based on the "Topics-over-label" Label structure.

1. CPLST: Conditional Principal Label space Transformation
2. BCS: Bayesian Compressed Sensing for Multi-Label Learning
3. WSABIE: Based on optimizing a weighted approximate ranking loss.
4. LEML: Low rank Emperical risk minimization for multi-label learning

| | | | Training set | | | Test set | | |
|---|---|---|---|---|---|---|---|---|
| Data set | $D$ | $L$ | $N_{train}$ | $\bar{L}$ | $\bar{D}$ | $N_{test}$ | $\bar{L}$ | $\bar{D}$ |
| bibtex | 1836 | 159 | 4880 | 2.40 | 68.74 | 2515 | 2.40 | 68.50 |
| delicious | 500 | 983 | 12920 | 19.03 | 18.17 | 3185 | 19.00 | 18.80 |
| compphys | 33,284 | 208 | 161 | 9.80 | 792.78 | 40 | 11.83 | 899.20 |
| eurlex | 5000 | 3993 | 17413 | 5.30 | 236.69 | 1935 | 5.32 | 240.96 |

Table 1: Statistics of the data sets used in our experiments. $\bar{L}$ denotes average number of positive labels per example; $\bar{D}$ denotes the average number of nonzero features per example.

Figure 11: Evaluation Data-set Table1 from the paper

The figures below summarizes efficiency of BMLPL with respect to the 4 frameworks described above. These experiment characterizes the advantages of the BMLPL Framework earlier described in section 2. [Figures from Paper].

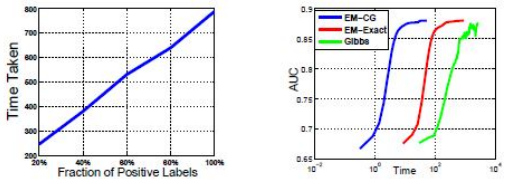


Figure 2: (Left) Scalability w.r.t. number of positive labels. (Right) Time vs accuracy comparison for Gibbs and EM (with exact and with CG based M steps)

Figure 12: Gibbs is slower than EM, but EM-CG parallized/block update is faster than EM (Section 5.3)

| | CPLST | BCS | WSABIE | LEML | BMLPL |
|---|---|---|---|---|---|
| bibtex | 0.8882 | 0.8614 | 0.9182 | 0.9040 | **0.9210** |
| delicious | 0.8834 | 0.8000 | 0.8561 | 0.8894 | **0.8950** |
| compphys | 0.7806 | 0.7884 | 0.8212 | **0.9274** | 0.9211 |
| eurlex | - | - | 0.8651 | 0.9456 | **0.9520** |

Table 2: Comparison of the various methods in terms of AUC scores on all the data sets. Note: CPLST and BCS were not feasible to run on the eurlex data, so we are unable to report those numbers here.

Figure 13:

| | BCS | LEML | BMLPL |
|---|---|---|---|
| bibtex | 0.7871 | 0.8332 | **0.8420** |
| compphys | 0.6442 | 0.7964 | **0.8012** |

Table 3: AUC scores with only 20% labels observed.

Figure 14:

## 5.5 Did we follow the 10 simple rules of Big Data Research.

1. Data are used to identify potential "primates" with positive Zika test
2. The data for the results section are presented from source code cited by the authors of the application paper

| Topic 1 (Nuclear) | Topic 2 (Agreements) | Topic 3 (Environment) | Topic 4 (Stats & Data) | Topic 5 (Fishing Trade) |
|---|---|---|---|---|
| nuclear safety | EC agreement | environmental protection | community statistics | fishing regulations |
| nuclear power station | trade agreement | waste management | statistical method | fishing agreement |
| radioactive effluent | EC interim agreement | env. monitoring | agri. statistics | fishery management |
| radioactive waste | trade cooperation | dangerous substance | statistics | fishing area |
| radioactive pollution | EC coop. agree. | pollution control measures | data transmission | conservation of fish stocks |

Table 4: Most probable words in different topics.

Figure 15:

3. These results does not claim to solve any of thing big.

4. This project was to get insights on an extension to a problem we covered in lectures. The checklist is to understand the "Topics over labels" concept

# 6 Conclusion:

This project is similar in spirit to Multi-Label classification examples we have discussed in lectures. In real-world setting these Multi-Label classifier algorithm face scalability and intractability time-complexity issues. These two main abstract ideas we have discussed in lectures, namely, "one-vs-one" and "one-vs-all" depends on the number of the example features and the size of the label vector. Hence, in real world Data the version space of classifiers is arbitarily large and intractable, which makes the "Bag-of-words" type classification more challenging than binary classification. The report is briefly analyzing the Multi-label classification extension to HW-3("Fortune-cookie question").

In section 2 we introduced BMLPL framework for the "Bag-of-words" setting classification. This framework assumes that the label vector follows a "Topics-over-Labels" pattern. The framework exploits this assumption to design a prediction model for such Label vector patterns. The general idea is that, such label vector $Y \epsilon R^{LxN}$ can be split into two parameters $V = [v_1...v_K] \epsilon R^{LxK}$ and $U = [u_1...u_N] \epsilon R^{K x N}$, where $V$ can be thought of "Topics for Labels" and $U$ can be thought of embedding of label vector. Equations (5-8) and figure 2 illustrates training model for BMLPL. Section 3 discusess the inference modelfor computing parameters $V$ and $W$ by Gibbs sampling, EM algorithm or EM-CG parallized/block.

In section 4, the proposed framework was implemented in a real-world Data to predict Zika virus label in primates. Each example has 45 features and a label vector of size 7. We assume these label vectors of this data set is highly correlation. For example, a lot of the primates have label as "primates" or many example share the same "Genus" and "Family Name". This correlation is exploited in BMLPL Framework, the Multi-labels are classified based of correlation between the features. In Zoology most of such feature matrices are known to be very sparse. Hence application of BMLPL on this Zika Primate Dataset makes it interesting could give some interesting observations.

The result of the exploratory experiment which was based on "one-vs-one"

using Naive Bayassian classifier have good training accuracy and correctly predicted the sero-negative Zika virus for species Cebus albifrons,cebus capucinus,Saimiri bolivienis. But using BMLPL the same species were predicted sero-positive Zika virus.

Both the ML algorithms, BMLPL and Naive-Bayes performed correctly. The task for both ML algorithms are completely different. In exploratory experiment, the ML algorithm is trained to predict the sero-negative/positive Zika virus based on the primates dataset based on each label vector classifier. In BMLPL framework the correlation of each features was used to predict the sero-negative/positive Zika virus.

For example, species Cebus albifrons have labels (Primates, Cebidae, Cebus ) which is sero-negative Zika virus. But Cebus albifrons share the same labels with species Cebus apella, which has been detected as sero-positive Zika virus. The BMLPL framework predicts potential sero-positive species if there is **"Spill-over of Zika virus"**.

Application of BMLPL on Zooligical Dataset was interesting, as it levarages two broad field of study. This report was an effort to explore the benefits of ML/AI techniques in different field of study. Particularly not involving humans as Data. Based on topics covered in WSU Math579, ML/AI techniques have potential for more interesting observations on such biological dataset. The application paper demonstrates the use of AI to make better disease spread model for primates.

# 7 References


1. CPLST: Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In NIPS, 2012.

2. BCS: Ashish Kapoor, Raajay Viswanathan, and Prateek Jain. Multilabel classification using bayesian compressed sensing. In NIPS, 2012.

3. WSABIE: Jason Weston, Samy Bengio, and Nicolas Usunier. WSABIE: Scaling up to large vocabulary image annotation. In IJCAI, 2011.

4. LEML: Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S Dhillon. Large-scale multi-label learning with missing labels. In ICML, 2014.

5. Paper ideas from Fall 2019 MATH579 Lectures