Project Report on

Classification and Prediction of Lymphoma Subtypes

Submitted by

Mathew George B160008CS
Vignesh Krishnan B160104CS
Varshini Naik B160382CS
Neema George B160081CS

Under the Guidance of

Dr Abdul Nazeer K A

Department of Computer Science and Engineering
National Institute of Technology, Calicut
Calicut, Kerala, India - 673601

Classification and Prediction of Lymphoma Subtypes

Mathew George Vignesh Krishnan Varshini Naik Neema George

Abstract— **Lymphoma is the abnormal growth of Immune System Cells, also called Lymphocytes. There are predominantly two types of Lymphoma - Hodgkin lymphoma and Non-Hodgkin lymphoma, further divided into many subtypes. Since the diagnosis and treatment of the different subtypes of lymphoma often differ substantially, efficient detection of the type of lymphoma is often detrimental. Automated classication of lymphocytes can save time, thus leading to a faster identication of the treatment required by the patient. Through this paper, we aim to use K-Nearest Neighbours, Support Vector Machines and Random Forest Classifiers to classify datasets of lymphoma patients into subtypes with respect to the similarity of gene expression data. We will also be analysing the accuracy and eciency of each model and proposing the best model for Lymphoma subtype prediction and discovery.**

I. INTRODUCTION

Lymphoma, or lymphatic cancer, is a general term for all cancers affecting the lymphatic system. The two main types of lymphoma in humans are Hodgkin's lymphoma(HL) and the non-Hodgkin's lymphoma (NHL). Hodgkin's lymphoma is a rare type of cancer and only about 14 percent of all lymphomas belong to this category. The other group, non-Hodgkin lymphoma, accounts for about 3 percent of all malignancies. In both kinds of lymphoma, cells in the lymphatic system become abnormal, dividing rapidly and growing without any order or control. Since lymphatic tissue is present in many parts of the body, lymphoma can start almost anywhere. It may occur in a single lymph node, a group of lymph nodes, or, sometimes, in other parts of the lymphatic system such as the bone marrow and spleen. Hodgkin lymphoma tends to spread in a fairly orderly way from one group of lymph nodes to the next group (Contiguous Spread). Non-Hodgkin lymphoma can spread to almost any part of the body, including the liver, bone marrow, and spleen (Non-Contiguous spread). Machine learning techniques can be used in the classification of lymphoma into Hodgkin Lymphoma and Non-Hodgkin Lymphoma. Through this paper, we aim to classify lymphoma gene expression data into four different classes, Hodgkin's Lymphoma and three subtypes of NHL - T Cell Lymphoma, B Cell Lymphoma and Follicular Lymphoma, so as to minimize the time and cost required in diagnosing patients efficiently and accurately.

II. PROPOSED METHODS

A. Support Vector Machines(SVM)

Support vector machine (SVM) is a method for the classication of both linear and non linear data. In case the data is non linear, it uses a non linear mapping to transform the original training data into a higher dimension. In this new dimension, it searches for the linear optimal separating hyperplane (i.e., a decision boundary separating the tuples of one class from another), as shown in Figure 3. With an appropriate non linear mapping to map lower dimensional data to a suciently higher dimension, data from two classes can always be separated by a hyperplane. The SVM nds this hyperplane using support vectors (essential training tuples) and margins (dened by the support vectors). During the training phase, the algorithm is called with two parameters, D and attribute list. D is the complete set of training tuples and their associated class labels. The parameter, attribute list, is a list of attributes describing the tuples. N represents the size of the data set D. The goal of the algorithm is to maximize w (width of the street). Then getStreetDetails function takes D, attributeList and w as inputs and returns the gutter points as output. Using these gutter points, the bias b can be calculated. Finally, w (width of the street) and bias, b are returned by the training phase algorithm. During the classication phase, the trained SVM is used to predict the class label for a given tuple using the width of the street(w) and bias, b which are calculated during the training phase. For a given tuple we calculate K(w, tuple) + b, where K is the kernel function, which is computed in the lower dimensional space and is equal to the inner product of the 2 vectors in the higher dimensional space. If this value is greater than or equal to 1, then the tuple belongs to Class 1, else if this value

is less than or equal to -1, then the tuple belongs to Class 2.

```
function GenSVM(D, attributeList)
N = D.size
Compute:
w=Sumover(i=1 to N)(iyiDi)
Subject to:
Sumover(i=1 to N) iyi = 0
(y, gutterPoints) =
getStreetDetails(D, attributeList, w)
equations = []
foreach xi belongs to gutterPoints
do
equations.add(yi(K(w,xi) + b) - 1 = 0)
end
b = solve(equations) return (w, b)
end function

function Classify((w, b),tuple)
if (K(w,tuple) + b) \geq 1 then
return getClassLabel1()
end
if K(w,tuple) + b  -1
then
return getClassLabel2()
end
end function
```

B. Random Forest

There are two stages in Random Forest algorithm, one is random forest creation, the other is to make a prediction from the random forest classifier created in the first stage. The whole process is shown below, and its easy to understand using the figure. Here the author firstly shows the Random Forest creation pseudocode:

- Randomly select K features from total m features where k much less than m
- Among the K features, calculate the node d using the best split point
- Split the node into daughter nodes using the best split
- Repeat the a to c steps until l number of nodes has been reached
- Build forest by repeating steps a to d for n number times to create n number of trees

prediction psuedocode:

- Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)

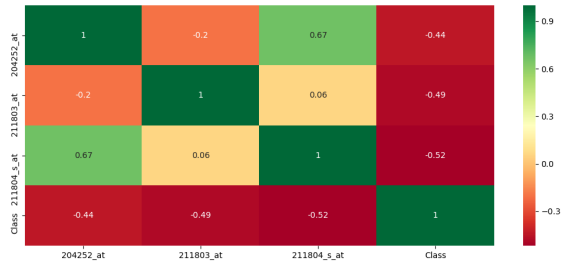


Fig. 1. Correlation Heatmap of Reporter Genes

- Calculate the votes for each predicted target
- Consider the high voted predicted target as the final prediction from the random forest algorithm

III. Literature Review

A. Towards Designing an Automated Classification of Lymphoma subtypes using Deep Neural Networks

[1]This paper uses Convolutional Neural Networks to predict the possible subtypes of lymphoma on an NIA curated dataset of Raw Microarray data saved as pictures.

1) Convolutional Neural Networks - introduction: A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

2) Input:

- The input layer or input volume is an image that has the following dimensions: [width x height x depth].It is a matrix of pixel values.
- Example- Input: [32x32x3],i.e., (width=32, height=32, depth=3)The depth here, represents R,G,B channels.

- The input layer should be divisible many times by 2 . Common numbers include 32, 64, 96, 224, 384, and 512.

3) Convolution: The objective of a Conv layer is to extract features of the input volume. A part of the image is connected to the next Conv layer because if all the pixels of the input is connected to the Conv layer, It will be too computationally expensive. So we are going to apply dot products between a receptive field and a filter on all the dimensions. The outcome of this operation is a single integer of the output volume (feature map). Then we slide the filter over the next receptive field of the same input image by a Stride and compute again the dot products between the new receptive field and the same filter. We repeat this process until we go through the entire input image. The output is going to be the input for the next layer.

- Filter, Kernel, or Feature Detector is a small matrix used for features detection. A typical filter on the first layer of a ConvNet might have a size [5x5x3].
- Convolved Feature, Activation Map or Feature Map is the output volume formed by sliding the filter over the image and computing the dot product.
- Receptive field is a local region of the input volume that has the same size as the filter.
- Depth is the number of filters.
- Depth column (or fibre) is the set of neurons that are all pointing to the same receptive field.
- Stride has the objective of producing smaller output volumes spatially. For example, if a stride=2, the filter will shift by the amount of 2 pixels as it convolves around the input volume. Normally, we set the stride in a way that the output volume is an integer and not a fraction. Common stride: 1 or 2 (Smaller strides work better in practice), uncommon stride: 3 or more.
- Zero-padding adds zeros around the outside of the input volume so that the convolutions end up with the same number of outputs as inputs. If we dont use padding the information at the borders will be lost after each Conv layer, which will reduce the size of the volumes as well as the performance.

4) Output Volume Computation [W2xH2xD2]:

- W2=(W1F+2P)/S+1
- H2=(H1F+2P)/S+1
- D2=K

Where,
[W1xH1xD1] : input volume size
F: receptive field size
S: stride
P: amount of zero padding used on the border.
K: depth

B. Computational modeling of early T-cell precursor acute lymphoblastic leukemia (ETP-ALL) to identify personalized therapy using genomics

[2] This paper makes use of Computational Biological Modelling to classify the dataset into ETP-ALL and non ETP-ALL types.

1) Computational Biological Modelling(CBM) - Definition: Exponential advancements in computer memory and performance has ushered in the use of system modelling, especially so in the field of bioinformatics.

2) Pseudocode:

- A conceptual model must be formulated. To do this, an understanding of the biological system being modelled, the components involved and the interaction between components has to be established. Assumptions must be made sagaciously to ensure that the model matches the system it strives to mimic. Also, the choices and simplifications that can be made at different situations must be specified and charted for the model to recognize an use.
- The model created above must be converted to mathematical form, containing a full listing of the state variables and the appropriate representations used for each.Each interaction must be represented as a mathematical relation.
- The created mathematical model must be converted to computer code. Various approaches like phase-field, lattice-Boltzmann, boundary element, finite volume, Monte Carlo, Gillespie, molecular dynamics and dissipative particle dynamics can be used.

IV. IMPLEMENTATION

The dataset has four different classes with 3 major attributes that directly contribute to the final accuracy. The dataset is then applied on with the above code.

V. RESULTS

- The accuracy values of Support Vector Machine analysis using Radial Basis Function Kernel was 95.57%
- The accuracy values of K-Nearest Neighbours analysis of the same dataset was 92.982%

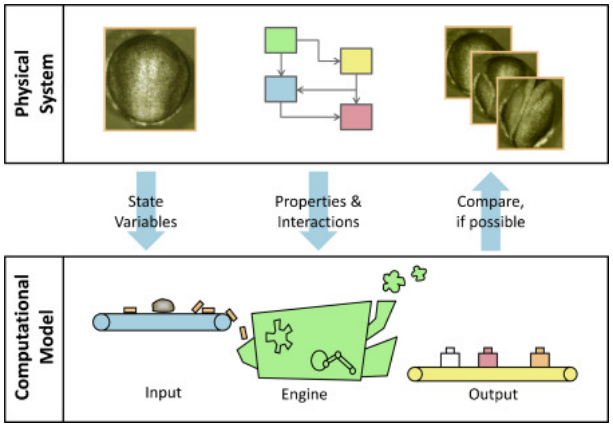


Fig. 2. Computational Biological Model Building

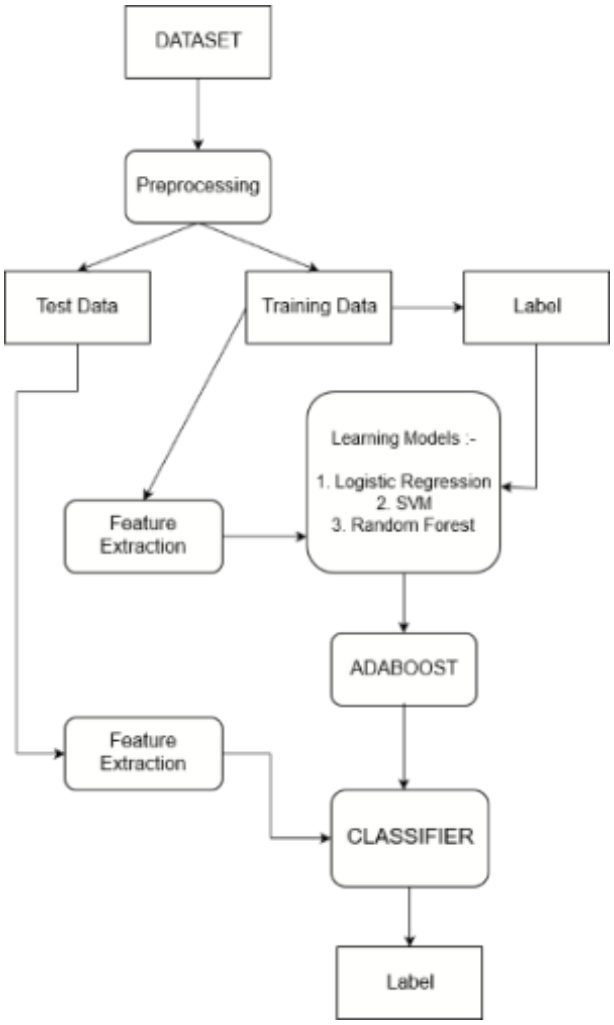
- The accuracy values of Random Forest Classifier analysis was 97.78%

Hence, the best classifier for this data is the Random Forest Classifier.

VI. Conclusion

The results obtained are very promising hence,the model depicted in this paper can be put to use in various practical applications.The method presented in this paper demonstrates that predicting the type of disease can be done with high accuracy and quick response time, making it suitable for practical purposes.

VII. Design Diagram



References

[1] Tambe, Rucha, et al. "Towards Designing an Automated Classification of Lymphoma subtypes using Deep Neural Networks". ACM India Joint International Conference, 143-149. 10.1145/3297001.3297019.

[2] Kumar, Ansu, et al. Computational Modeling of Early T-Cell Precursor Acute Lymphoblastic Leukemia (ETP-ALL) to Identify Personalized Therapy Using Genomics. Leukemia Research, Elsevier,Volume 78, 7 Jan. 2019.

[3] Lymphoma — CDC. Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, www.cdc.gov/cancer/lymphoma/index.htm.

[4] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics. New York, NY, USA: Springer-Verlag, 2011.

[5] N. V. Orlov et al., "Automatic Classification of Lymphoma Images With Transform-Based Global Features," in IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 4, pp. 1003-1013, July 2010. doi: 10.1109/TITB.2010.205069 Curr. Hematol. Malig. Rep. (September) (2017), p. 14

[6] N.Pochet, F.D.Smet, J.A.K.Suykens, and B.L.R.D.Moor,Systematic benchmarking of microarray data classication: Assessing the role of non linearity and dimensionality reduction, Bioinformatics, vol. 20, no. 17, pp. 31853195, Jul. 2004.

[7] O. Chapelle, V. Sindhwani, and S. S. Keerthi, Optimization techniques for semi-supervised support vector machines,J.Mach.Learn.Res.,vol.9, pp. 203233, Jan. 2008.

[8] D. C. Koestler et al., Semi-supervised recursively partitioned mixture models for identifying cancer subtypes, Bioinformatics, vol. 26, no. 20, pp. 25782585, 2010.

[9] B. Schlkopf, C. J. C. Burges, and A. J. Smola, Advances in Kernel Methods: Support Vector Learning. Cambridge, MA, USA: MIT Press, 1999.

[10] C.Hsu, C.Chang,and C.Lin.(2013).A Practical Guide to Support Vector Classication. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin

[11] E. Kreyszig, Introductory Mathematical Statistics. New York, NY, USA: Wiley, 1970.