Project Report on

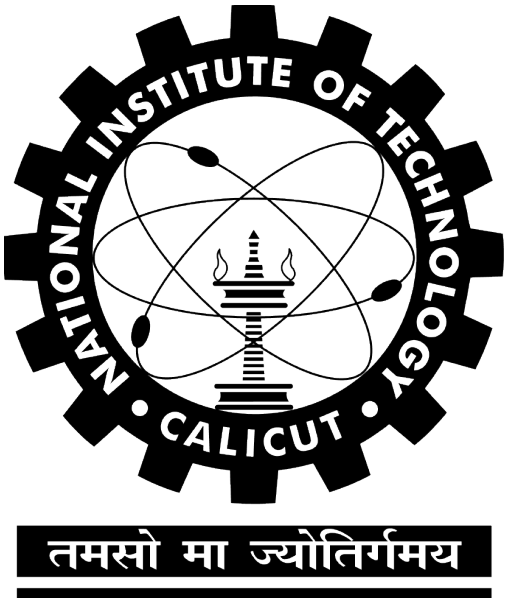
Prediction of lung cancer patient survival via supervised machine learning classification techniques

Submitted by

TGDK Sumanathilaka B150413CS
Eva Reshma Toppo B150129CS
Rakhee Poonam Lakra B150547CS
Aravind A B150263CS

Under the Guidance of

Dr. Abdul Nazeer



Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

March 28, 2019

Prediction of lung cancer patient survival via supervised machine learning classification techniques

TGDK Sumanathilaka Aravind A Eva Reshma Toppo
Rakhee Poonam Lakra

Abstract: Outcomes for cancer patients have been previously estimated by applying various machine learning techniques to large data-sets such as the Surveillance, Epidemiology, and End Results (SEER) program database. In particular for lung cancer, it is not well understood which types of techniques would yield more predictive information, and which data attributes should be used in order to determine this information. In this study, a number of supervised learning techniques is applied to the SEER database to classify lung cancer patients in terms of survival, including linear regression, Decision Trees, Gradient Boosting Machines (GBM), Support Vector Machines (SVM), CNN and Deep belief model. Key data attributes in applying these methods include tumor grade, tumor size, gender, age, stage, and number of primaries, with the goal to enable comparison of predictive power between the various methods. The prediction is treated like a continuous target, rather than a classification into categories, as a first step towards improving survival prediction. We conclude that application of these supervised learning techniques to lung cancer data in the SEER database may be of use to estimate patient survival time with the ultimate goal to inform patient care decisions, and that the performance of these techniques with this particular data-set may be on par with that of classical methods.

1 Introduction

Machine learning uses mathematical algorithms implemented as computer programs to identify patterns in large data-sets, and to iteratively improve in performing this identification with additional data. The algorithms are commonly used in different domains and diverse applications. Using these techniques to evaluate disease outcomes can be challenging. Since patient data is generally unavailable for public analysis. One exception is the SEER Program from the National Cancer Institute (NCI) at the National Institute of Health (NIH). This data-set provides information on cancer statistics of the United States population. Machine learning techniques are applied to this data-set to analyze data specific to lung cancer, with the goal to evaluate the predictive power of this techniques. Lung cancer was chosen as it ranks as a leading cause of cancer-related death, with dismal 5-year survival rates.

Given a data-set of lung cancer patients with particular demographic(e.g., age), diagnostic (e.g., tumour size), and procedural information (e.g.Radiation and/or Surgery applied), the question is whether patient survival can be computationally predicted with any precision. In this study, patients diagnosed with lung cancer during the years 2004-2009 were selected in order to be able to predict their survival time. A number of supervised learning methods was employed to classify patients based on survival time as function of key attributes and thus, help illustrate the predictive value of the various methods. The techniques chosen include linear regression, Decision Trees, Gradient Boosting Machines, Support Vector Machines,

CNN and Deep belief model. This report also enables comparing the predictive value of the methods when applied with the chosen attributes to analyze the lung cancer patient data. The data-set in this study focuses on measurements available at or near the time of diagnosis, which represents a more proactive set of survival predictors.

2 Problem statement

To predict the survival time of the patient suffering from lung cancer using various machine learning techniques like Multi Layer Perceptron(MLP), Deep Belief Network(DBN) and Long Short-Term Memory(LSTM).

3 Literature Review

Previously published work has analyzed the SEER database via statistical methods as well as classification techniques. In earlier work, the concept of agglomerative clustering was applied to generate groups of cancer patients. The algorithm of clustering of cancer data (ACCD) was proposed to predict outcomes, with any number of factors as input and with the goal of grouping patients uniformly in terms of survival. The approach was applied to a large breast cancer data-set from the SEER database using information concerning tumor size, tumor extension and lymph node status. The results showed the approach to be more effective than the traditional TNM (tumor-node-metastasis) cancer staging system.

Prediction models for breast cancer survivability using a large data-set were developed applying two popular data mining algorithms, artificial neural networks and Decision Trees, as well as a commonly used statistical method, logistic regression. Ten-fold cross validation methods were employed to measure the unbiased estimate of the three prediction models for performance comparison purposes. The results showed that Decision Tree (C5) was the best predictor with 93.6% accuracy on the holdout sample, artificial neural networks were second best with 91.2% accuracy, and logistic regression models obtained 89.2% accuracy. A study was performed to develop prediction models for prostate cancer survivability, employing support vector machines (SVM) in addition to the previously mentioned three techniques. In this case, the results singled out SVM as the most accurate predictor (92.85% accuracy), followed by artificial neural networks and Decision Trees. Similarly, prostate cancer survivability was evaluated using artificial neural networks, Decision Trees, and logistic regression. Various techniques were compared using SEER colon cancer patient data to predict survival, finding that neural networks were the most accurate. Ensemble voting of three best performing classifiers resulted in optimal prediction and area under the Receiver Operating Characteristic (ROC) curve for colon cancer survival.

Few studies have evaluated lung cancer patient survival by analyzing the SEER database with machine learning techniques, including ensemble clustering-based approaches SVM and logistic regression and unsupervised methods. Data classification techniques were evaluated to determine the likelihood of patients with certain symptoms to develop lung cancer. The performance of C4.5 and Naive-Bayes classifiers was compared applied to lung cancer data from the SEER database, achieving 90% precision in predicting patient survival. Ensemble voting of five Decision Tree based classifiers and meta-classifiers was determined to yield the best prediction of lung cancer survivability in terms of precision and area under the ROC curve.

Association rule mining techniques have been employed to determine interesting association or correlation relationships among a large set of items; different techniques to extract the rules and standard criteria have been proposed, suggesting how to choose the best rules and select optimization based on a given data-set. An automated technique to create a tree of rules for lung cancer was implemented, some of which were redundant and were manually removed based on domain knowledge. Three factors were considered: the maximum branching factors, adding a new branch, and the factor to be used when adding a new branch. The authors proposed a tree-based algorithm using the entire data-set from the very be-

ginning, and descending into the data in a depth-first fashion using a greedy approach. Each node of the tree represented a segment and hence an association rule. The attributes included: age, birth place, cancer grade, diagnostic confirmation, farthest extension of tumor, lymph node involvement, type of surgery performed, reason for no surgery, order of surgery and radiation, scope of regional lymph node surgery, cancer stage, number of malignant tumor, and total regional lymph nodes examined. Measuring the efficacy of treatments and surgery is a desired result from analyzing the SEER data-set, even though the data-set lacks information regarding chemotherapy.

The effectiveness of treatment was taken into consideration. The study explored the question whether lung cancer patients survive longer with surgery or radiation, or both. A Propensity Score was used, representing a conditional probability that a unit will receive a treatment given a set of observed covariates. Two methods were applied for estimating the score, namely, logistic regression and classification tree. Since patients can receive surgery or radiation separately or together, the score was calculated for each group and then the attributes were ranked. Statistical information related to the combination of survival time and radiation was extracted, and a classification tree was generated for each group. The results showed that patients who did not receive radiation with or without surgery had the longest survival time.

3.1 Linear Regression

The simplest method implemented is linear regression, one of the oldest and most widely used correlational techniques. The goal of the method is to fit a straight line to a set of data points using a series of coefficients multiplied to each input, like a weighting function, and an intercept. The weights are decided within the linear regression function in a way to minimize the mean error. These weight coefficients multiplied by the respective inputs, plus an intercept, give a general function for the outcome, patient survival time.

3.1.1 Algrithm of Linear Regression

Input: Pre-processed dataset with chosen features.
Output: predicted survival time (in months) of the patient

Steps:

m examples $\{(\mathbf{x}^i, y^i)\}_i$

example $\mathbf{x} = \langle x_0, x_1, .., x_n \rangle$

$h_{\mathbf{a}}(\mathbf{x}) = a_0 x_0 + a_1 x_1 + .. + a_n x_n = \sum_{j=0}^{n} a_j x_j = \mathbf{x}a$

$J(\mathbf{a}) = \frac{1}{2m} \sum_{i=1}^{m} (h_{\mathbf{a}}(x^i) - y^i)^2$

$\frac{\partial J(\mathbf{a})}{\partial a_j} = \frac{1}{m} \sum_{i=1}^{m} x_j^i (h_{\mathbf{a}}(x^i) - y^i) = \frac{1}{m} \mathbf{X}_j^T (\mathbf{Xa} - \mathbf{y})$

$\nabla J(\mathbf{a}) = \frac{1}{m} \mathbf{X}^T (\mathbf{Xa} - \mathbf{y})$

Pseudocode: Given α, $\mathbf{X}$, $\mathbf{y}$

Initialize $\mathbf{a} = \langle 1, .., 1 \rangle^T$

Normalize $\mathbf{X}$

Repeat until convergence

- $\mathbf{a} = \mathbf{a} - \frac{\alpha}{m} \mathbf{X}^T (\mathbf{Xa} - \mathbf{y})$

Output $\mathbf{a}$

3.2 Decision tree

Decision tree builds regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Leaf nodes are nothing but indication of survival time.For regression decision trees there are more classifications than in a typical classification decision tree that makes the outcomes near continuous.

3.3 Random Forest

The Random Forest technique generates a number of decision trees during training which are allowed to split randomly from a seed point. This results in a forest of randomly generated decision trees whose outcomes are ensemble by the Random Forest Algorithm to predict more accurately than a single tree does alone. It is a meta estimator that ts a number of classifying decision trees on various sub-samples of the data set and use averaging to improve the predictive accuracy and control over-fitting. The number of trees was set to 500.

3.4 Support Vector Machine

Epsilon-Support Vector Regression. Assuming that a set of training data has been labeled as belonging to one of two sets, the algorithm represents them in space and specifies a hyper-plane maximally distant from both to separate them. The plane is called the maximal margin hyper-plane. If a linear separation is not possible, the algorithm employs kernel methods to obtain a non-linear mapping to a feature space. Kernel method selected is rbf. A drawback of SVM is that the method can be subject to over-fitting when the data is noisy.

3.5 Gradient Boosting Machine

Gradient Boosting for regression. GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function. Loss function to be optimized i.e least squares regression. Learning rate shrinks the contribution of each tree by learning rate. There is a trade-off between learning rate and n-estimators The number of boosting stages to perform are 100. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance. The maximum depth of the individual regression estimators is 3. It limits the number of nodes in the tree.

3.6 Ensemble

A custom ensemble method was used to bring all these models together for a more accurate prediction. The results were expected to be better with the custom ensemble than with any single approach, and the ensemble was simple to implement and easily adaptable to model adjustments.

Model	RMSE	Standard Deviation	Standard Deviation of Residuals	Mean	Weighting Factor for Custom Ensemble
Custom Ensemble – Regression Weighting	15.30	6.39	15.30	19.41	(Intercept of 0.613)
GBM	15.32	6.47	15.31	19.38	0.620
Linear Regression	15.38	6.80	18.35	19.47	0.118
Random Forests	15.63	6.77	15.63	19.43	0.057
Decision Trees	15.81	4.84	15.81	19.45	0.096
SVM	15.82	5.89	15.39	14.59	0.158

Figure 1: Comparison of modeling techniques ranked from best to worst based on RMSE values. Both the standard deviation of the predictions and the standard deviation of the difference between predictions and the actual values (Standard Deviation of Residuals) are shown along with the ensemble weighting factors.

4 Pseudo code for implemented models

Experimental Framework

1. Read the SEER data-set.
2. Preprocessing the data-set.
   - Removing redundant data.
   - Removing empty values.
   - Removing Noise.
3. Divide data-set into test data and training data.
4. Train the Developed model with training data.
5. Run the trained model for testing data.
6. Calculate Mean value and RMSE.

4.1 Multi Layer Perceptron

A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in

between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function.

Multilayer perceptrons are often applied to supervised learning problems3: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Backpropagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).

Feedforward networks such as MLPs are like tennis, or ping pong. They are mainly involved in two motions, a constant back and forth. You can think of this ping pong of guesses and answers as a kind of accelerated science, since each guess is a test of what we think we know, and each response is feedback letting us know how wrong we are.

In the forward pass, the signal flow moves from the input layer through the hidden layers to the output layer, and the decision of the output layer is measured against the ground truth labels.

In the backward pass, using backpropagation and the chain rule of calculus, partial derivatives of the error function w.r.t. the various weights and biases are back-propagated through the MLP. That act of differentiation gives us a gradient, or a landscape of error, along which the parameters may be adjusted as they move the MLP one step closer to the error minimum. This can be done with any gradient-based optimisation algorithm such as stochastic gradient descent. The network keeps playing that game of tennis until the error can go no lower. This state is known as convergence.

4.1.1 Algorithm Multi Layer Perceptron(MLP) (Forward pass)

Require: pattern , MLP, enumeration of all neurons in topological order
Ensure: calculate output of MLP

1. **for all** input neuron i **do**
2. set $a_i = x_i$
3. **end for**
4. **for all** hidden and otput neurons i in topological order **do**
5. set $\text{net}_i = w_{i0} + \sum_{j \epsilon Pred(i)} \text{w}_{ij} a_j$
6. set $a_i = f_{log}(net_i)$
7. ***end for***
8. ***for all*** *output neutron i* ***do***
9. *assemble* a_i *in output vector y*
10. ***end for***
11. ***return*** *y*

4.2 Deep Belief Model

Deep belief network (DBN) is a generative graphical model, or alternatively a class of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers, but not between units within each layer. DBNs can be viewed as a composition of simple, unsupervised networks such as restricted Boltzmann machines (RBMs) or auto-encoders, where each sub-network's hidden layer serves as the visible layer for the next layer.

$$P(x, h^1, ...h^l) = \prod_{k=1}^{(k-2)} \text{P}(\text{h}^k | h^{(k+1)})$$

4.2.1 Algorithm for Deep Belief Model(DBN)

1. Train the first layer as an RBM that models the input x as its visible layer

2. The first layer is used as the input data for the second layer which is chosen either by mean activations of $[P(h^1 = 1|h^0)$ or samples of $P(h^1|h^0)$

3. Iterate for the desired number of layers, each time propagating upward either samples or mean values.

4. Fine-tune all the parameters of this deep architecture with respect to log-likelihood or with respect to a supervised training criterion

4.3 Long Short-Term Memory

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer". It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

4.3.1 Algorithm for Long short-term memory(LSTM)

An Long short-term memory(LSTM) network computes a mapping from an input sequence x = $(x_1...x_T)$ to an output sequence y = $(y_1...y_T$) by calculating the network unit activations using the following equations iteratively from t = 1 to T:

1. $i_t = \sigma(\mathrm{W}_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i)$
2. $f_t = \sigma(\mathrm{W}_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f)$
3. $c_t = f_t \bigodot c_{t-1} + i_t \bigodot \mathrm{g}(\mathrm{W}_{cx}x_t + \mathrm{W}_{cm}m_{t-1} + b_c)$
4. $o_t = \sigma(\mathrm{W}_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o)$
5. $m_t = o_t \bigodot h(c_t)$
6. $y_t = \phi(\mathrm{W}_{ym}m_t + b_y)$

where the W terms denote weight matrices (e.g. W_{ix} is the matrix of weights from the input gate to the input), W_{ic}, W_{fc}, W_{oc} are diagonal weight matrices for peephole connections, the b terms denote bias vectors (b_i is the input gate bias vector), σ is the logistic sigmoid function and i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the cell output activation vector m, $\bigodot$ is the element-wise product of the vectors, g and h are the cell input and cell output activation functions, generally and in this paper *tanh*, and ϕ is the network output activation function.

5 Observation

We have taken the lung cancer dataset from SEER(Surveillance, Epidemiology, and End Results). 19 out of 134 attributes were selected from the SEER database for the implementation of Multilayer perceptron(MLP), Deep Belief Nework(DBN) and Least Short-Term Memory(LSTM) models.
We are taking Root Mean Squared Error (RSME) value into consideration for the prediction of lung cancer survival. LSTM has the least RSME value of 10.53, MLP with 1 layer has RSME as 14.8787,

MLP with 2 layer 14.9684 and DBN has the highest RSME with a value of 16.399. Hence, we conclude that, from the output of Least Short-Term Memory, meaningful predictions can be made with reasonable accuracy bands describable by the resulting statistics.

Table 1: Comparison of modeling techniques ranked from best to worst based on RMSE values

Models	RMSE	Standard Deviation	Mean of Predictions	Mean of residuals
LSTM	10.53	14.2652	42.8517	7.5264
MLP with 1 layer	14.8787	11.5504	45.4631	9.3820
MLP with 2 layer	14.9684	11.6146	46.3205	9.4452
DBN	16.399	7.4902	40.0	14.5900

6 Conclusion

The results from this study suggest that a correlational approach via supervised machine learning may be applicable to lung cancer patient survival prognosis, in the sense that meaningful predictions can be made with reasonable accuracy bands describable by the resulting statistics. The only model that may be non-applicable is Decision Trees, as it has too few discrete outputs. Despite the issues with the other models investigated, no model other than Decision Trees seems truly lacking, with the Least Short-Term Memory(LSTM) model displaying stronger performance, and the Deep Belief Network(DBN) AND MultiLayer Perceptron(MLP) being worthy of independent attention as it predicts scores similarly to the others.

7 Future Work

Future work could reevaluate the inputs for the selected models. While RMSE was chosen during our up-front design, other metrics may be warranted; the scores and standard deviations of LSTM, MLP and MLPs suggest that a deeper analysis may prove fruitful.

A new effort to evaluate each individual criteria and how it relates to patient survival, especially in the case of longer-lived patients, could be key to more accurate and predictive correlational supervised machine learning algorithms.

References

[1] Chip M. Lynch, Behnaz Abdollahi et.alPrediction of lung cancer patient survival via supervised machinelearning classification techniques, International Journal of MedicalInformatics 108 (2017) 18.

[2] Mangai JA, Wagle S, Kumar VS.An improved k nearest neighbour classifier using interestingnessmeasures for medical image mining, International Journal of MedicalHealth Biomedical Bioengineering and Pharmaceutical Engineering2013; 7(9):23640.

[3] P. Thamilselvan* and J. G. R. SathiaseelanDetection and Classification of Lung Cancer MRI Images by usingEnhanced K Nearest Neighbor Algorithm, Indian Journal of Scienceand Technology, Vol 9(43),DOI:0.17485/ijst/2016/v9i43/104642, November 2016.

[4] Thamilselvan P, Sathiaseelan JGRA comparative study of data mining algorithms for imageclassification. International Journal of Education and ManagementEngineering. 2015; 5(2):19.

[5] Chen CKThe classification of cancer stage in microarray data. ComputerMethod and Programs in Biomedicine. 2012; 108(3):10707.

[6] Ramteke RJ, Monali YK.Automatic medical image classification and abnormality detectionusing k nearest neighbor. International Journal of Advanced ComputerResearch. 2012; 2(4):1906.