Prediction of Heart Disease using Backpropagation Neural Network

Sanjana Chalavadi
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
schalava@buffalo.edu

Backpropagation

Backpropogation is an algorithm for determining how a single training example would like to nudge the weights and biases in terms of relative proportions to those changes that brings the most rapid decrease to the costs.

Method

When training an artificial neural network, we pass data into our model. This way the data flows through the model via forward propagation where we are repeatedly calculating the weight sum of the previous layers activation output with corresponding weights and then passing this sum to the next layers activation function. We do this until we reach the output layer and at this point, we calculate loss on our output.

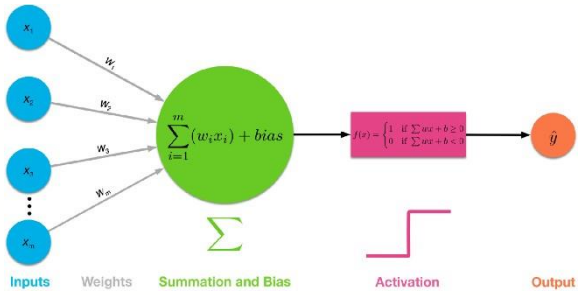


Figure1: Single layer perceptron network

Input (Z) - It's the weighted sum of the activation outputs from the previous layer(l-1)

$$\mathbf{Z} = \sum_{0}^{n-1} \boldsymbol{w^l a^{l-1}}$$

Loss (C) - It's the squared difference of desired out and activation output for that node

$$\mathbf{C} = \sum_{j=0}^{n-1} (\boldsymbol{a_j^l - y_j})$$

Activation Output (A) - It's the result of passing the input to whatever activation function(g) we choose for that layer

$$\boldsymbol{A = g^l Z^l}$$

Gradient descent works to minimize this loss. The way gradient descent does this minimization process is by first calculating the gradient of the loss function with respect to the weights and then updating the weights in the network accordingly. To do the actual calculation, gradient descent uses backpropagation.

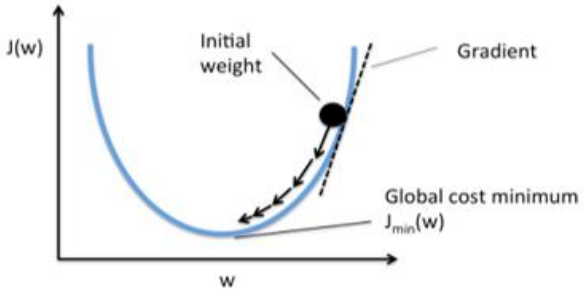


Figure2: Gradient descent

To calculate the loss with respect to some weight, w_{12} for all n training samples, we need to calculate the average derivative of the loss function for all the training samples. The same process is done for each weight in the network to calculate the derivative of C with respect to each weight.

$$\frac{dC}{dw_{12}^{(l)}} = \frac{1}{n}\sum_{i=0}^{n-1}\frac{dC_i}{dw_{12}^{(l)}}$$

Introduction

There are 60000 miles of blood vessels inside your body and on average, human heart pumps around 2000 gallons of blood in entire body. Studies show that men have more noticeable symptoms compared to women. Men experience symptoms like chest pain, discomfort, and stress or a combination of them. In addition, they could also experience pain in random areas such as arms, neck, and jaw along with shortness of breath, sweating or heart burn. In women, symptoms might be uncomfortable throbbing, pressure, fullness, pain concentrated in one area of the chest, nausea etc. It's astonishing that such a small organ is responsible for complete body functioning. Cardiovascular diseases kill around 17 million people every year and the common symptoms exhibited are myocardial infractions and heart failures. Heart failure occurs mainly due to blood not being pumped properly and when it cannot meet the needs of the body.

Complexity

Determining the probability of having cardiac disease is hard as it cannot completely just depend on risk factors. With the advancement in analytics, it's possible to use different machine learning techniques to identify patterns and forecast the heart disease that can help to reduce diagnostic time along with good accuracy and effectiveness, but it cannot be solely used without proper domain knowledge. Dataset used has data from Cleveland, Hungary, and Switzerland which is merged. The lifestyle of people from these places could be vastly different and that could impact on their heart conditions which could in turn impact our evaluation metrics.

Objective

Analysing risk factors and determine whether someone has a severe cardiac disease. Several insights are extracted from the dataset to understand the importance of each variable and their relationship.

About the data:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

Figure3: Raw Dataset

It has 1025 rows of data and attributes include "**age**", "**sex**" "**cp**" (chest pain: Atypical Angina, Non-Anginal Pain, Typical Angina, Asymptomatic), "**trestbps**" (resting blood pressure mmHg on admission to hospital), "**chol**" (cholesterol in mg/dl), "**fbs**" (fasting blood sugar >120mg/dl, 1=true, 0=false), "**restecg**" (resting electrocardiographic measurement, 0: normal, 1: ST-T wave abnormality,2:definite left ventricular hypertrophy), "**thalach**" (maximum heart rate achieved), "**exang**" (exercise induced angina), "**oldpeak**"(ST depression induced by exercise relative to rest), "**slope**"(the slope of the peak exercise ST segment, 0: upsloping, 1:flat, 2:downsloping), "**ca**" (number of major blood vessels from 0-3), "**thal**" (blood disorder thalassemia, normal, fixed defect, reversable defect), "**target**" (heart disease there or not).

Experiment and Graphs

The dataset is clean as it has no null/missing values nor any duplicate values present.

Figure4: Heart Disease Distribution with respect to Age

Upon checking the heart disease distribution with respect to age, the age of 54 seems like the most common age for heart disease followed by 51,52 and 41.

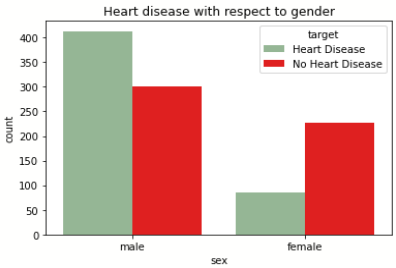


Figure5: HD with Gender

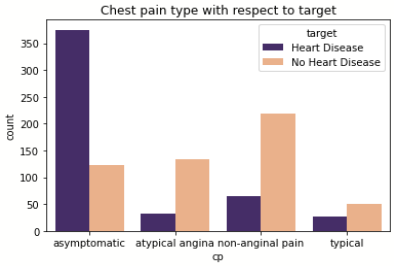


Figure6: HD with Chest Pain

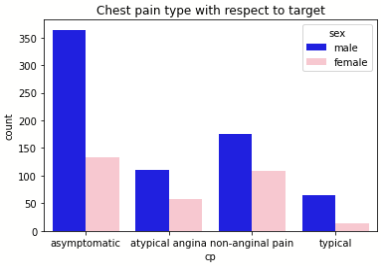


Figure7: HD with Gender

Majority of the patients having the heart disease are males. Most of the heart disease patients have asymptomatic chest pain which also means that the symptoms are silent, so in case the patient has any heart disease, they would lack the intensity of a classic heart attack such as extreme chest pain and pressure, stabbing pain in the arm, neck, or jaw, sudden shortness of breath, sweating, and dizziness. The only way to tell if the patient had asymptomatic attack is by an electrocardiogram or echocardiogram.

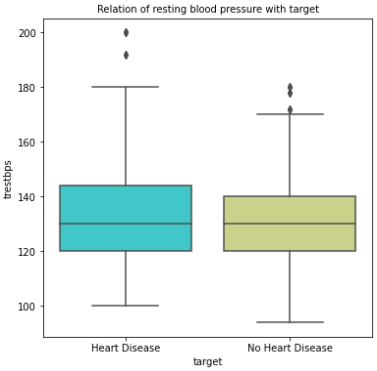


Figure8: HD with Resting BP

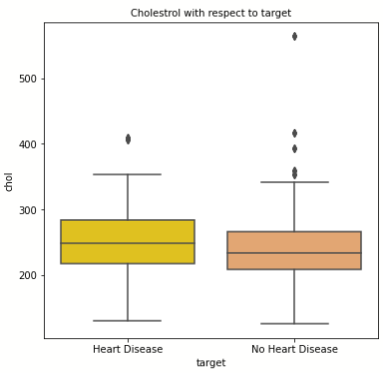


Figure9: HD with Cholesterol

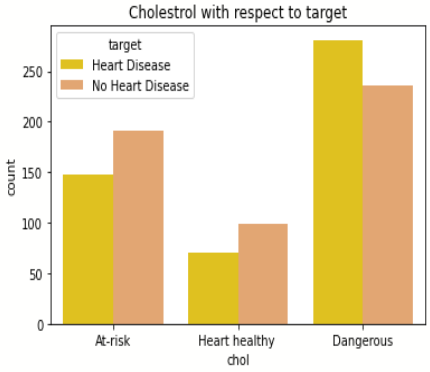


Figure10: HD with Chol classes

Figure8 clearly suggests that the patients who are most likely to not suffer from the disease have a slightly lower blood pressure than the patients who have heart diseases. In general, low LDL and high HDL cholesterol levels

are good for heart health. A healthy heart has a total cholesterol level of less than 200mg/dL whereas at risk is considered as 200-239 and dangerous is considered as 240 and higher. From Figure9, it looks like most of the heart disease patients have dangerous levels of cholesterol followed by at-risk category. Cholesterol seems like one of the significant factors in determining a healthy heart.

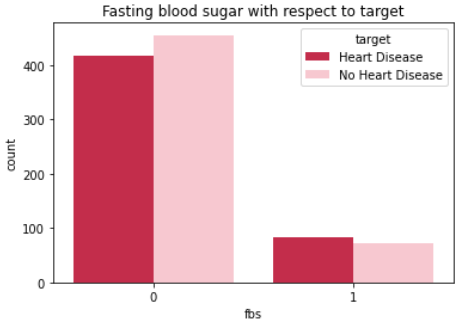


Figure11: HD with fbs

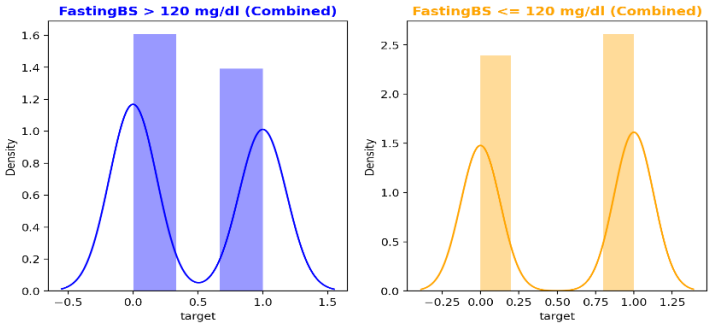


Figure12: HD with fbs levels (<>120)

There is no insight to check if fasting blood sugar levels impact heart disease, but you can notice that if blood sugar is low (<=120) in figure12, it could be a factor contributing to heart disease.

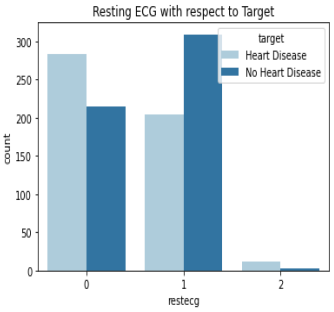


Figure13: HD with restecg

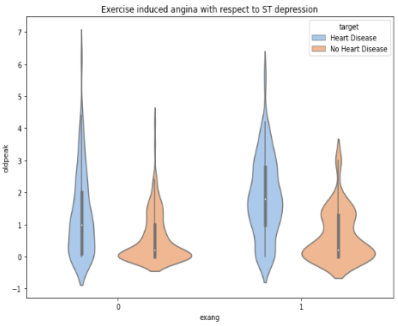


Figure14: HD with exang

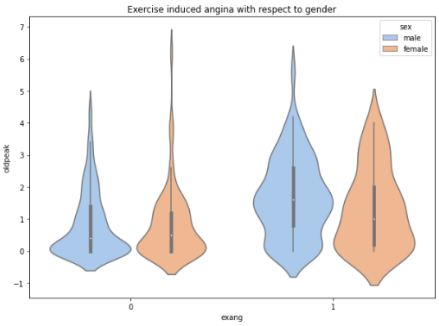


Figure15: HD with exang and gender

Resting 12-lead ECG is a non-invasive test that can detect abnormalities including arrhythmias, evidence of coronary heart disease, left ventricular hypertrophy and bundle branch blocks. In this case, level 0 means normal, level 1 means ST-T wave abnormality and level 2 means definite left ventricular hypertrophy. From the figure13, it shows that most of the heart disease patients have either normal ECG or some ST-T wave abnormality, so it seems like ECG is not the sole factor in estimating if a person has a heart disease.

Upon checking exercise induced angina with respect to ST depression induced by exercise relative to rest, people have exang (AP) is a common complaint of cardiac patients, particularly when exercising in the cold. If you notice figure14, you can see that if exercise induced angina is present and if old peak is greater than zero, then it could be a possible case for heart disease and more commonly seen in males which can be seen in figure15.

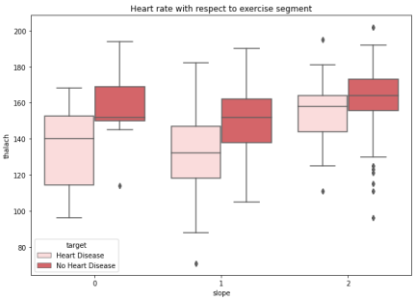


Figure16: HD with Slope

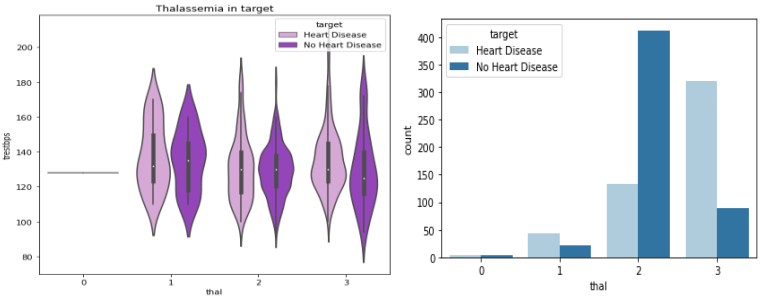


Figure17/18: HD with Thalassemia & rest bps

ST depression (horizontal or down sloping) is the most reliable indicator of exercise-induced ischemia. In figure16, for most of the heart disease patients, the ST segment is either flat or down sloping which is as expected. This method of testing could be a good way to diagnose heart diseases.

Thalassemia is an inherited blood disorder that causes your body to have less haemoglobin than normal. Haemoglobin enables red blood cells to carry oxygen. Thalassemia can cause anaemia leaving the patient fatigued and t-major leads to heart failure and liver problems. Figure17 shows that patients of reversible defect have blood pressure (> 120) which could be possible reasons for heart disease and notice the one in fixed defect too (value 1), they seem to have higher BP than 120 as well. In addition, figure18 shows abnormal levels of thalassemia indicate heart disease.

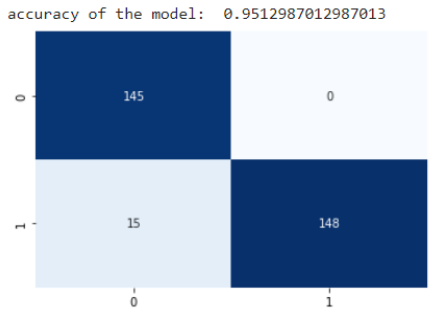
Results

Using the prediction method in artificial neural network, we are seeing that prediction of accuracy level of 95% for 14 neurons in hidden layers with sigmoid activation function using 1000 iterations.

Parameter	Value
Iterations	1000
Number of neurons in input layer	8
Number of neurons in hidden layer	14
Number of neurons in output layer	1

Table: Parameter Settings

It's understandable because neural networks are prone to overfitting because they learn several parameters while building the model. In this case, we should be looking for more correct classifications of heart disease. Incorrect classifications can lead to wrong medical diagnosis which can be followed up with severe life changing consequences. Recall should be considered as focus should be more on correctly diagnosing the disease.




```
print(classification_report(y_test,y_pred.round()))
```

	precision	recall	f1-score	support
0	0.91	1.00	0.95	145
1	1.00	0.91	0.95	163
accuracy			0.95	308
macro avg	0.95	0.95	0.95	308
weighted avg	0.96	0.95	0.95	308

Figure19: Confusion Matrix and Evaluation Metrics for Backpropagation

References

Datasource: https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/

https://my.clevelandclinic.org/health/articles/11920-cholesterol-numbers-what-do-they-mean

https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f

Prediction of Heart Disease using Support Vector Machine

Sanjana Chalavadi
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
schalava@buffalo.edu

Background

For 2-dimensional data, when we use the threshold that gives us the largest margin to make classifications, it will use a maximal margin classifier. But if the training data was not well spread and if the there was an outlier observation that was very close towards the opposite class, the maximum margin classifier would be near that outlier which would be very far from the other class observations. So maximum margin classifiers are super sensitive to outliers in the training data

Figure1: Maximum margin classifier

To make a threshold that is not sensitive to outliers, we must allow misclassifications. Choosing a threshold that allows misclassifications is related to Bias/Variance Trade off which is a solid foundation principle in machine learning. When we allow misclassifications, the distance between the observations and thresholds is called soft margin. To know which soft margin is better, we use cross validation to determine many misclassifications and observations to allow inside of the soft margin to get a good classification.

Figure2: Soft margin in 1D

When we use a soft margin to determine the location of threshold, we will be using a support vector classifier to classify observations. The observations within the edge and within the margin are called support vectors through which the name originates support vector classifier. When the data is 2-dimensional, support vector classifier is a line and in that, the soft margin is measure from the points. The parallel lines tell us where all the points are in relation to soft margin.

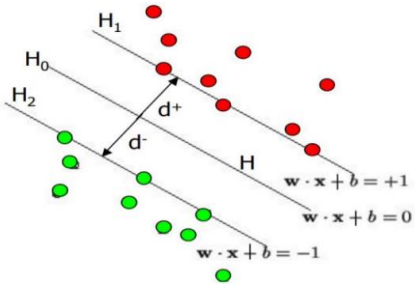


Figure3: Support vector classifier in 2D

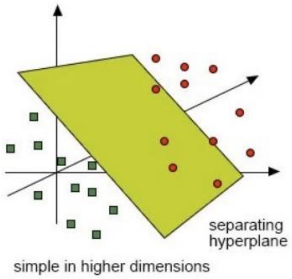


Figure4: Support vector classifier in 3D

In 2-dimensional, the classifier has the form $\boldsymbol{f(x) = W^T x + b}$

Geometric margin is distance between the point and the decision line which can be denoted by

$$\gamma = y\frac{\boldsymbol{w^T x} + b}{||w||}$$

When comparing hyperplanes, hyperplane with the largest γ should be selected and for finding optimal hyperplane, values of **w** and **b** should be found for which training error is zero.

The distance between H_1 and H_2 (refer Figure2) which is the size of the margin, is given by $\frac{\mathbf{2}}{\|\boldsymbol{W}\|}$

For 3-dimensional data, the support vector classifier forma a plane instead of a line and we classify new observations by determining which side of the plane they are on. When the data are in 4 or more dimensions, the support vector classifier is a hyperplane. Support vector classifier might seem correct because they can handle outliers and in addition, they allow misclassifications along with overlapping classifications. If there are multiple overlaps? Since maximal margin classifiers and support vector classifiers cannot handle that, support vector machine comes into play.

Support Vector Machine Methodology

Support vector machine starts with data in a relatively low dimension followed by moving the data into a higher dimension. Next, it finds a support vector classifier that separates the higher dimensional data into two groups. But to decide on how to transform the data, support vector machines use something called kernel functions to automatically find support vector classifiers in higher dimensions.

In polynomial kernel, the degree d (relationship between each point), it systematically increases the dimensions by setting d and the relationships between each pair of observations are used to find a support vector classifier. To find a good value for d, cross validation can be used. Another common kernel is radial basis function (RBF) kernel. This kernel finds support vectors classifiers in infinite dimensions which uses the logic of weighted nearest neighbour model. It's important to remember that kernel functions only calculate the relationships between every pair of points assuming they are in higher dimension, they don't do the transformation. The trick of calculating the high dimensional relationships without any transformations is called **the kernel trick**. The kernel trick reduces the amount of support vector machines by avoiding the math that transforms the data from low to high dimensions along with calculating the relationships in infinite dimensions used by the radial kernel possible.

Polynomial Kernel

To calculate the high-dimensional relationships, we must calculate the dot products between each pair of points. The dot product values should be plugged into the kernel to get the high-dimensional relationships.

$$(\boldsymbol{a}\ \boldsymbol{X}\ \boldsymbol{b} + \boldsymbol{r})^{\boldsymbol{d}}$$

"a" and "b" refer to the two observations for which we must calculate the high dimensional relationship for, "r" is the polynomial's coefficient and "d" is the degree of the polynomial. The values "r" and "d" are determined using cross validation.

Radial Kernel

a and b are two observations. The amount of influence one observation has on another is a function of the squared distance. Gamma is determined by cross validation, it scales the squared distance, and thus, it scales the influence. The further two observations are from each other, the less influence they have on each other. When we plug values in radial kernel, we get the high dimensional relationship.

$$\mathbf{e}^{-\gamma(a-b)^2}$$

Experiment and Graphs

Upon doing SVM with radial kernel (because of multiple features) without optimization of parameters (refer Figure4), it shows that 145 patients did not have heart disease and in that **97%** were correctly classified and out of 163 that have heart disease, **96%** were correctly classified in test data. The support vector did good without

any optimization. There is a possibility to improve predictions using cross validation for finding optimal parameters.

```
Accuracy of Support Vector Machine: 96.42857142857143

              precision    recall  f1-score   support

           0       0.95      0.97      0.96       145
           1       0.97      0.96      0.97       163

    accuracy                           0.96       308
   macro avg       0.96      0.96      0.96       308
weighted avg       0.96      0.96      0.96       308
```

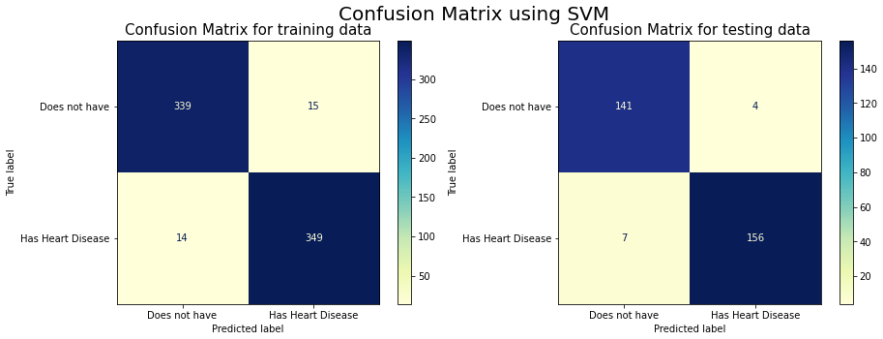


Figure4: Confusion Matrix and Evaluation Metrics for SVM without optimization

After optimizing parameters with cross validation by using **GridSearch** method, values for γ and C are found

```
print(optimal_params.best_params_) #print the best parameters
```

```
{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
```

Figure5: Optimal Parameters

Another SVM is built using the optimal γ and C and the optimized SVM performs perfectly. All the patients were correctly classified in train and test data. There's no overfitting either.

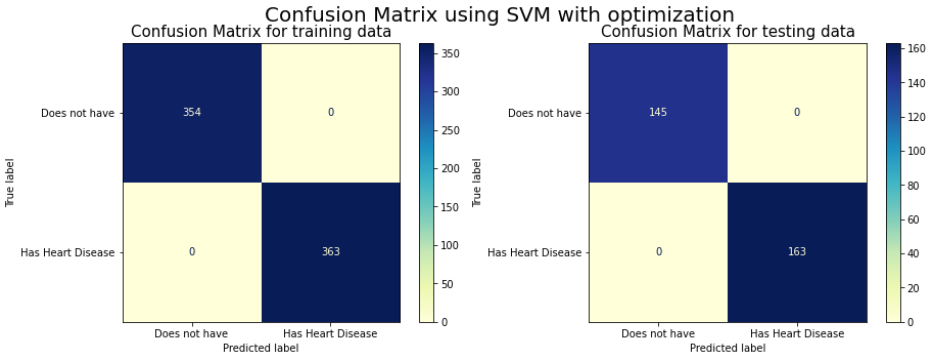


Figure6: SVM with optimization

Since there are 22 features, it's impossible to project it in 22-dimensional graphs so used principal component analysis to check 2D and 3D graph. From the scree plot, it looks like more than 10 PC's capture 90% of the variance but PC1 captures majority of variance followed by PC2 and PC3.

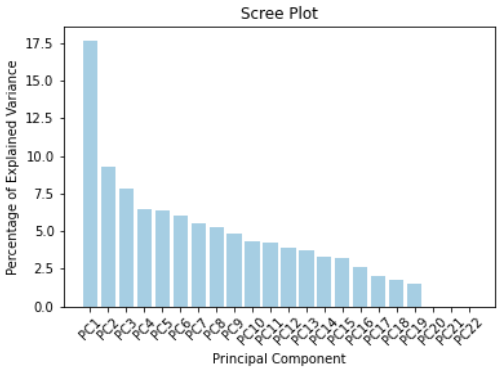


Figure7: Scree plot using PCs

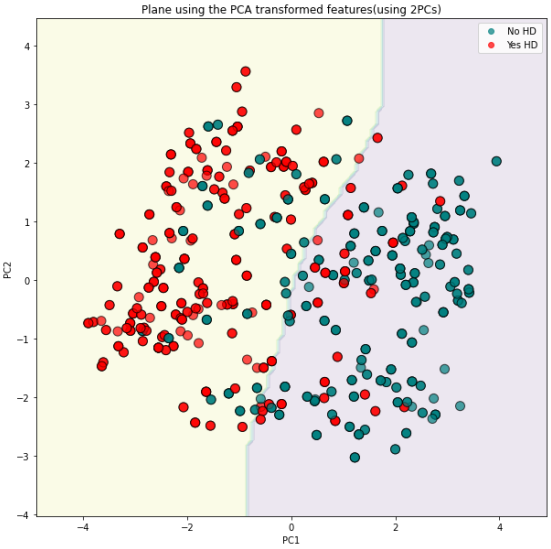


Figure8: 2D Plot using 2PCs

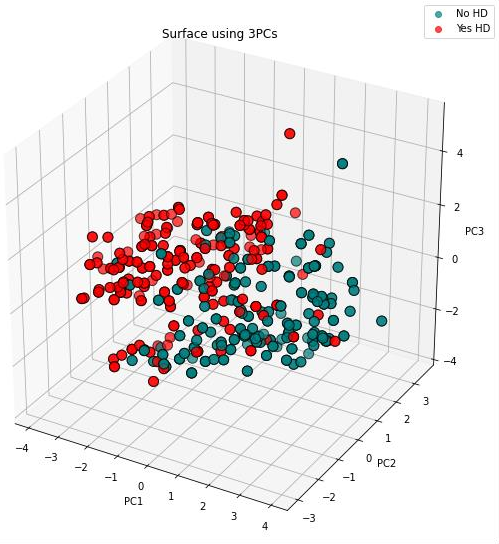


Figure9: 3D plot using 3PCs

In Figure8, the left side of the graph shows patients with heart disease and the right side with no heart disease. There seems to be some misclassifications, but it should be noted that we are just using 2 features here. In Figure9 (if rotated in plotly), the boundaries can be seen properly. Overall, SVM with parameter optimization performs perfectly (even when compared to neural networks).

References

Professor Changyou Chen's CSE 574 SVM Slides

SVM lecture in https://statquest.org/video-index/

https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Prediction of Heart Disease using Naïve Bayes

Sanjana Chalavadi
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
schalava@buffalo.edu

Introduction

Naïve Bayes is a probability-based algorithm used for binary classifications. For classifications, we need X features where X = $(x_1, x_2, x_3, x_4, \ldots . x_n)$ to predict the dependant variable Y. So, in layman terms, given X, event that Y=y is determined by P (Y=y | X = $(x_1, x_2, x_3, x_4, \ldots . x_n)$). For calculating the probabilities of discrete values, the probabilities are called likelihoods. Start with an initial guess say p(x) which can be any probability but usually the common guess is estimated from the training data. The initial guess is called **prior probability.**

$$\mathbf{P(Y|X)} = \frac{\boldsymbol{P(X \mid Y)\, P(Y)}}{\boldsymbol{P(X)}} \rightarrow \text{Posterior probability} = \frac{\text{likelihood x prior probability}}{\text{marginal probability}}$$

In this case of heart prediction problem, X = (age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal) and Y = target. As the number of parameters increase in the model, naïve assumption should be made which is assume all the parameters we have are independent. It's naïve because, there is a very less possibility for features to be independent. With this assumption, we can classify as follows

$$\boldsymbol{P(Y|X)} = \frac{\boldsymbol{P(Y)\prod_i P(X_i|Y)}}{\boldsymbol{P(X)}}$$

For categorical features, some feature values may never show up so their likelihood is zero which means the posterior probability in turn will be 0. For that, we can do Laplace estimator where we add imaginary sample per category. So, the likelihood of categorical features can be represented as

$$\boldsymbol{P(X_i = x \mid Y = y)} = \frac{\boldsymbol{\#(X_i=x,\ Y=y)+1}}{\boldsymbol{\#Y=y+p}}$$, p= number of data pts such that $X_i = x,\ Y = y$

For continuous features, naïve bayes can be written as $P(Y|X) = \frac{P(Y)\prod_i f(X_i|Y)}{f(X)}$ where f is the probability density function assuming we have features which are normally distributed.

Gaussian naïve bayes follows gaussian normal distribution while supporting continuous variables with the assumption that each class is distributed normally with a mean μ and standard deviation σ given by

$$f(X_i = x|Y = y) = \frac{1}{\sqrt{2\pi\sigma'^2}} e^{\frac{(x-\mu')^2}{2\sigma'^2}}$$

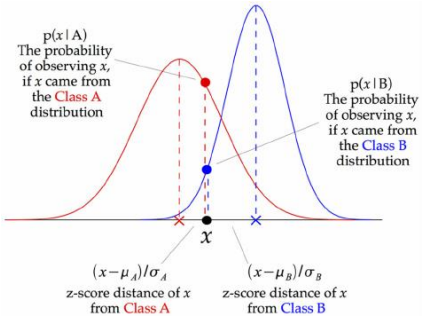


Figure1: Gaussian Naïve Bayes

About the data

Followed up on the same dataset used for back propagation, I implemented gaussian naïve bayes model because the dataset has both numerical and categorical features.

Experiment and Graphs

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.88	0.83	0.86	145
1	0.86	0.90	0.88	163
accuracy			0.87	308
macro avg	0.87	0.87	0.87	308
weighted avg	0.87	0.87	0.87	308

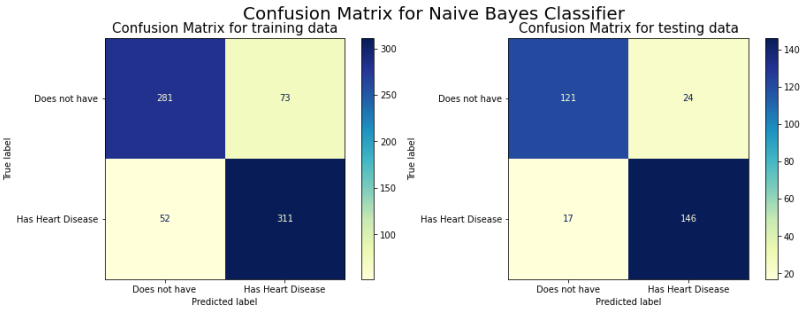


Figure2: Evaluation metrics and confusion matrix for Gaussian Naïve Bayes

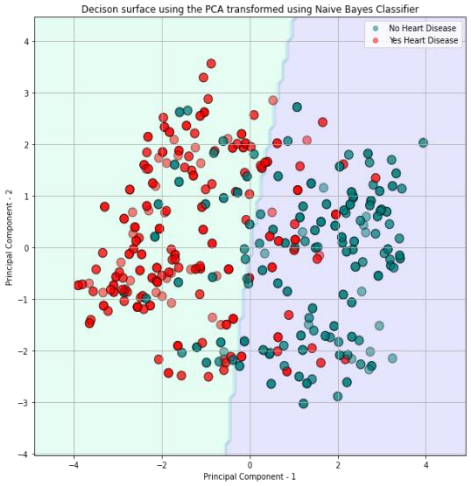


Figure3: Plot using 2PC's

Upon doing Gaussian Naïve Bayes, it shows that 121 patients did not have heart disease and in that **83%** were correctly classified and out of 163 that have heart disease, **90%** were correctly classified in test data. Gaussian Naïve Bayes performed reasonably ok but when compared to SVM and Neural Network, it didn't perform that well. If you notice Figure3, there are lot of mis classifications too using the first two PC's, but it should also be noted that the 90% of the variance is captured in 10PC's.

References

https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0

Analysis of Crime Rates by US State using Gaussian Mixture Model

Sanjana Chalavadi
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
schalava@buffalo.edu

Introduction

Gaussian mixture model is another clustering technique which is based on probability density estimation that uses a procedure called expectation minimization (EM) for fitting the model parameters. For groups that overlap in feature space, it's hard to interpret which assignment is right and if the clusters are defined in some non-circular shape, normal clustering algorithms like k-means won't be able to discover that common overlap assignment.

Gaussian mixture models are built upon k-means model more like an extension where clusters are modelled using gaussian distributions so in addition to mean, there will be covariance which will help figure out the cluster shapes. In this way, we can fit the model by maximizing the likelihood of the data using EM algorithm. EM algorithm will assign data to each cluster with probability.

In gaussian mixture model, we begin with several components linked by **c** where each one is described by gaussian distribution where each one has a mean μ, covariance σ and size π where each component could have a different shape like height, area, etc., so the joint distribution is defined by the weighted average of those components.

$$\mathbf{p\ (x)} = \sum_c \boldsymbol{\pi}_\mathbf{c}\ \mathbf{N}(\mathbf{x}\ ;\ \mu_c, \boldsymbol{\sigma}_c)$$

Components with large π are selected often so select component with probability π called z so p (z = c) = π_c, we can draw value x from the gaussian distribution so p (x | z = c) = N (x; μ_c, σ_c) . Presence of unknown values of z helps identify patterns in x like groups.

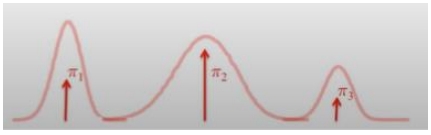


Figure1: Example distribution showing curves with different π

Most of the time features are high dimensional so in that case, multivariate gaussian should be used which indeed uses vector mean μ (with size as number of features say n), data points x, and nxn covariance matrix ∑. It's represented as follows

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}}|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu})\right\}$$

Expectation minimization proceeds iteratively in two steps called expectation step and maximation step.

In expectation step, the gaussian parameters μ, ∑ and size π will be fixed. For each data point x_i and each cluster c, responsibility value r_{ic} which measures relative probability for the data point x_i *belonging to cluster* c. For doing that, first probability is computed under model c and normalized over clusters. If c ends up being a wrong cluster value for the point x_i, r_{ic} value will be smaller and if it's ends up being the best, r_{ic} will be equal to 1.

$$r_{ic} = \frac{\sum_c \boldsymbol{\pi}_\mathbf{c}\ \mathbf{N}(\mathbf{x}\ ;\ \mu_c, \boldsymbol{\sigma}_c)}{\sum_{c^i} \boldsymbol{\pi}_{c^i}\ \mathbf{N}(\mathbf{x}\ ;\ \mu_{c^i}, \boldsymbol{\sigma}_{c^i})}$$

In maximization step, r_{ic} will be fixed and parameters of clusters like μ, ∑ and π will be updated. For each cluster c, parameters are updated using r_{ic} weighted probabilities for data points i, so for total number of data points m, it will be the sum of r_{ic} and size is the same value normalized by total number of data points. The weighted mean μ is just the weighted average of the data so each point x_i is given by r_{ic}

$$\boldsymbol{m_c} = \sum_i \mathbf{r_{ic}}\ ,\ \boldsymbol{\pi_c}\ =\ \frac{\boldsymbol{m_c}}{\mathrm{m}},\ \mu_\mathbf{c} = \frac{1}{\boldsymbol{m_c}}\sum_i \mathbf{r_{ic}} x^{(i)}$$

Log likelihood is the log probability of the data points under the mixture model so its sum by data points.

$$\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \, \mathcal{N}(x_i \, ; \, \mu_c, \Sigma_c) \right]$$

We need to iterate until convergence and initialization is important for local minima. Like k-means, we also need to choose the number of clusters which can be done log likelihood of the test data. All in all, in gaussian mixture model, we compute soft assignments for each data point and then use those soft assignments in maximum likelihood estimates and EM will converge but most likely on local minima. If EM is difficult to implement, there are other alternatives like stochastic EM and hard EM.

About the data

This dataset contains arrests per 100000 residents for murder, assault, and rape in each of the 50 US states from the year 1973. In addition, percent population living in urban areas is given. Through gaussian mixture model, we can analyse patterns in crimes by groups.

	Country	Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6

Figure2: Dataset

Experiment and Graphs

The data is clean with no missing values present. The distribution of numerical variables is below.

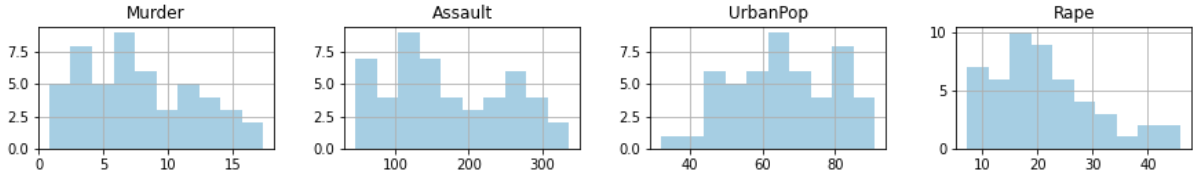


Figure3: Distribution of the numerical attributes

Only the attribute UrbanPop seems close to gaussian distribution. From Figure4, you can see that the attributes Murder and Assault are highly correlated, and Assault is also significantly correlated with Rape. So, there is a chance that data points might share clusters which are more than the number of attributes, but we can confirm this through the GMM process.

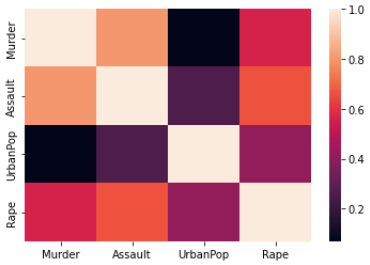


Figure4: Correlation plot

To begin with, I randomly chose 3 clusters across first two principal components, but it does not seem like a right choice as there are few data points which fall beyond the grouping and its not distinct.

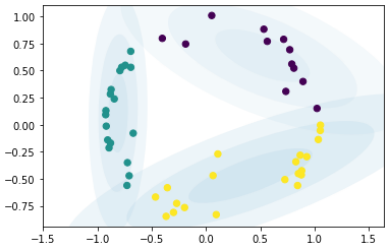


Figure4: GMM using 3 clusters

To select clusters using a more structure and in a accurate way, I used silhouette method for picking k. This method checks how well the clusters are compact and well distinguished. The score nearer to 1 indicates a better clustering.

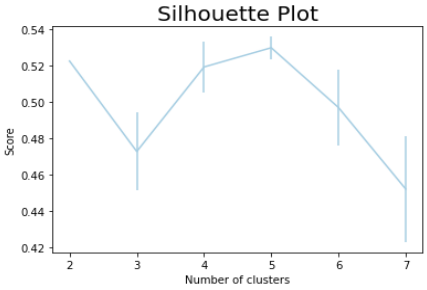


Figure5: Silhouette plot for cluster pick

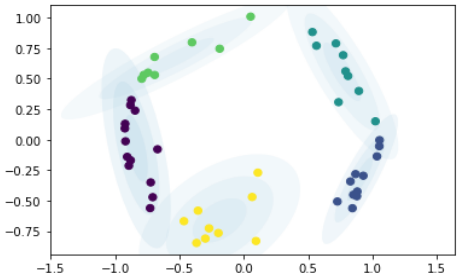


Figure6: GMM using clusters from Silhouette plot

So, from Figure5, it shows 5 clusters as a good pick and when used that in GMM, it shows good capture of all data points (refer Figure6). Based on covariance, the model used "full" shaped ellipse for fitting the clusters which makes sense because the dataset is small. So, its interesting to note that the attributes Murder, Assault, Urbanpop and Rape share common links so the number of clusters will be equal to or more than attributes confirms the assumption I made earlier.

Evaluation

Bayesian Information Criterion is used for evaluation. This criterion will give estimate on how good the gaussian mixture model is. The lower the BIC, the better the model is for prediction. Furthermore, to avoid overfitting, BIC penalizes the model with big number of clusters. From Figure7, BIC score is low for 5 clusters which aligns with our process before. To check the slope of the BIC curve change, gradient of BIC scores is calculated. In Figure8, it can be noticed that cluster 5 and 6 almost closely have the same value so the gradient there could be near to zero so that backs up the fact that the correct number of cluster choice is 5.

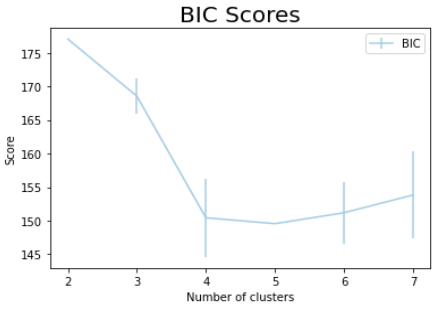


Figure7: BIC evaluation

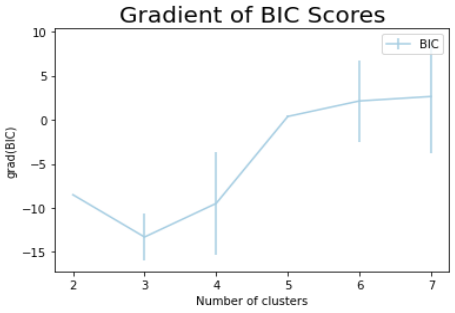


Figure8: Gradient of BIC Scores

References

Data source: https://www.kaggle.com/datasets/halimedogan/usarrests

UC Irvine Professor Alexander Ihler Video Lecture

https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95