CS199:

Classification of Bioinformatics Research Papers

Shashank Khanna

University of California, Los Angeles

Centre of Excellence for Big Data Computing at UCLA

Department of Bioinformatics

University of California, Los Angeles

Contents

[1 Description of the Problem 1](#_Toc474324564)

[1.1 Background 1](#_Toc474324565)

[1.2 Problem Statement 1](#_Toc474324566)

[2 Part 1 : Algorithms 2](#_Toc474324567)

[2.1 Decision Tree 2](#_Toc474324568)

[2.1.1 Algorithm Summary 2](#_Toc474324569)

[2.1.2 Application to the classification problem 2](#_Toc474324570)

[2.2 SVM (Linear/RBF) 2](#_Toc474324571)

[2.2.1 Algorithm Summary 3](#_Toc474324572)

[2.2.2 Application to the classification problem 3](#_Toc474324573)

[2.3 Nearest Neighbors 3](#_Toc474324574)

[2.3.1 Algorithm Summary 3](#_Toc474324575)

[2.3.2 Application to the classification problem 3](#_Toc474324576)

[2.4 Neural Networks 3](#_Toc474324577)

[2.4.1 Algorithm Summary 3](#_Toc474324578)

[2.4.2 Application to the classification problem 3](#_Toc474324579)

[2.5 Random Forest 4](#_Toc474324580)

[2.5.1 Algorithm Summary 4](#_Toc474324581)

[2.5.2 Application to the classification problem 4](#_Toc474324582)

[2.6 Naïve Bayes 4](#_Toc474324583)

[2.6.1 Algorithm Summary 4](#_Toc474324584)

[2.6.2 Application to the classification problem 4](#_Toc474324585)

[3 Testing algorithms on Iris Training Data 5](#_Toc474324586)

[3.1 Decision Tree 5](#_Toc474324587)

[3.1.1 Algorithm Summary 5](#_Toc474324588)

[3.1.2 Application to the classification problem 5](#_Toc474324589)

[3.2 SVM (Linear/RBF) 5](#_Toc474324590)

[3.2.1 Algorithm Summary 5](#_Toc474324591)

[3.2.2 Application to the classification problem 6](#_Toc474324592)

[3.3 Nearest Neighbors 6](#_Toc474324593)

[3.3.1 Algorithm Summary 6](#_Toc474324594)

[3.3.2 Application to the classification problem 6](#_Toc474324595)

[3.4 Neural Networks 6](#_Toc474324596)

[3.4.1 Algorithm Summary 6](#_Toc474324597)

[3.4.2 Application to the classification problem 6](#_Toc474324598)

[3.5 Random Forest 6](#_Toc474324599)

[3.5.1 Algorithm Summary 7](#_Toc474324600)

[3.5.2 Application to the classification problem 7](#_Toc474324601)

[3.6 Naïve Bayes 7](#_Toc474324602)

[3.6.1 Algorithm Summary 7](#_Toc474324603)

[3.6.2 Application to the classification problem 7](#_Toc474324604)

[4 Feature Identification: Creating The Training Set 8](#_Toc474324605)

[4.1 Hypothesizing Features 8](#_Toc474324606)

[4.1.1 Presence of Links 8](#_Toc474324607)

[4.1.2 Sentences from Abstract 8](#_Toc474324608)

[4.1.3 Presence of Repositories 8](#_Toc474324609)

[4.1.4 Presence of Programming Language Names 9](#_Toc474324610)

[4.1.5 Presence of Journal Types 9](#_Toc474324611)

[4.1.6 Frequency of Links 9](#_Toc474324612)

[4.2 Quantifying Features 9](#_Toc474324613)

[4.2.1 Algorithm Summary 9](#_Toc474324614)

[4.2.2 Application to the classification problem 9](#_Toc474324615)

[5 Extracting the Features And Producing Test Data 10](#_Toc474324616)

[5.1 Decision Tree 10](#_Toc474324617)

[5.1.1 Algorithm Summary 10](#_Toc474324618)

[5.1.2 Application to the classification problem 10](#_Toc474324619)

[5.2 SVM (Linear/RBF) 10](#_Toc474324620)

[5.2.1 Algorithm Summary 10](#_Toc474324621)

[5.2.2 Application to the classification problem 11](#_Toc474324622)

[5.3 Nearest Neighbors 11](#_Toc474324623)

[5.3.1 Algorithm Summary 11](#_Toc474324624)

[5.3.2 Application to the classification problem 11](#_Toc474324625)

[5.4 Neural Networks 11](#_Toc474324626)

[5.4.1 Algorithm Summary 11](#_Toc474324627)

[5.4.2 Application to the classification problem 11](#_Toc474324628)

[5.5 Random Forest 11](#_Toc474324629)

[5.5.1 Algorithm Summary 12](#_Toc474324630)

[5.5.2 Application to the classification problem 12](#_Toc474324631)

[5.6 Naïve Bayes 12](#_Toc474324632)

[5.6.1 Algorithm Summary 12](#_Toc474324633)

[5.6.2 Application to the classification problem 12](#_Toc474324634)

# Description of the Problem

## Background

Currently, Bioinformatics software tools are present in silos with no common endpoint which could serve as a one size fits all solution for researchers to access the software they need. This present condition, leads us to create a solution to collate all present bioinformatics software based resources and provide a common access point for them. In order to achieve that, all publications that could potentially contain software related to Bioinformatics, which is henceforth referred to as a tool in this paper, would need to be inspected, and identified as a tool or a not a tool. Given the scale of this problem, it is physically impossible to individually study every single publication, which leads the problem towards a machine learning solution.

## Problem Statement

The issue of identifying publications as a tool or a non-tool can be looked upon as a traditional classification problem. Given that the publications can be modified to fit the correct consumption format, Python scripts can potentially be used to implement several possible classification algorithms.

# Part 1 : Algorithms

## Decision Tree

Decision Trees classify a pattern through a sequence of questions where the next question depends on the answer to the current question. The questions are asked in a true/false style. (MSU, 2017)

### Algorithm Summary

Classic decision tree works by building a tree and making a decision on a dataset corresponds to a traversal of the tree from root to one of the leaves which corresponds to the decision.

Given a set S, we can organise the test data into a tree where a class is applied to each test. Hence, we can continually split the training set into smaller subsets with the subsets defined by a particular property, until we have reached a node where there is no need to divide the set further and all the data corresponding to that node have the same class label.

### Application to the classification problem

The efficacy in application of the decision tree algorithm to this problem would depend on the relative strength of the features picked in deciding whether a particular publication can be classified as a tool or otherwise. This is because the algorithm works based on quantifying the importance of features to decide which decision to make.

## SVM (Linear/RBF)

Support vector machine is a popular binary classifier that can classify a set of data by separating it into two groups at a time.

### Algorithm Summary

The algorithm works by finding the best separating hyperplane, i.e. the greatest distance between the nearest positive and negative value. In this context the positive value means the case where a particular data point is a particular classification whereas the negative value refers to the other classifications. The new data points are subsequently provided a classification based on the side of the hyperplane that they lie in.

### Application to the classification problem

Since our particular problem is a binary classification problem, this strategy can be applied to classifying tools and non-tools by mapping the positive value to a tool and vice versa.

## Similar Binary Classifiers

Aside from the two main algorithms considered, there are several major classifying algorithms considered for this given problem. These include the Nearest Neighbors algorithm, the Random Forest Classification, Naïve Bayes, Logistic Regression and Neural Networks.

## Picking the best strategy

In order to decide on the best algorithm for the given set of data, the algorithms were first experimented on the iris data set, which lead to the following results in efficiency.

### Testing the algorithms on Iris: a popular dataset

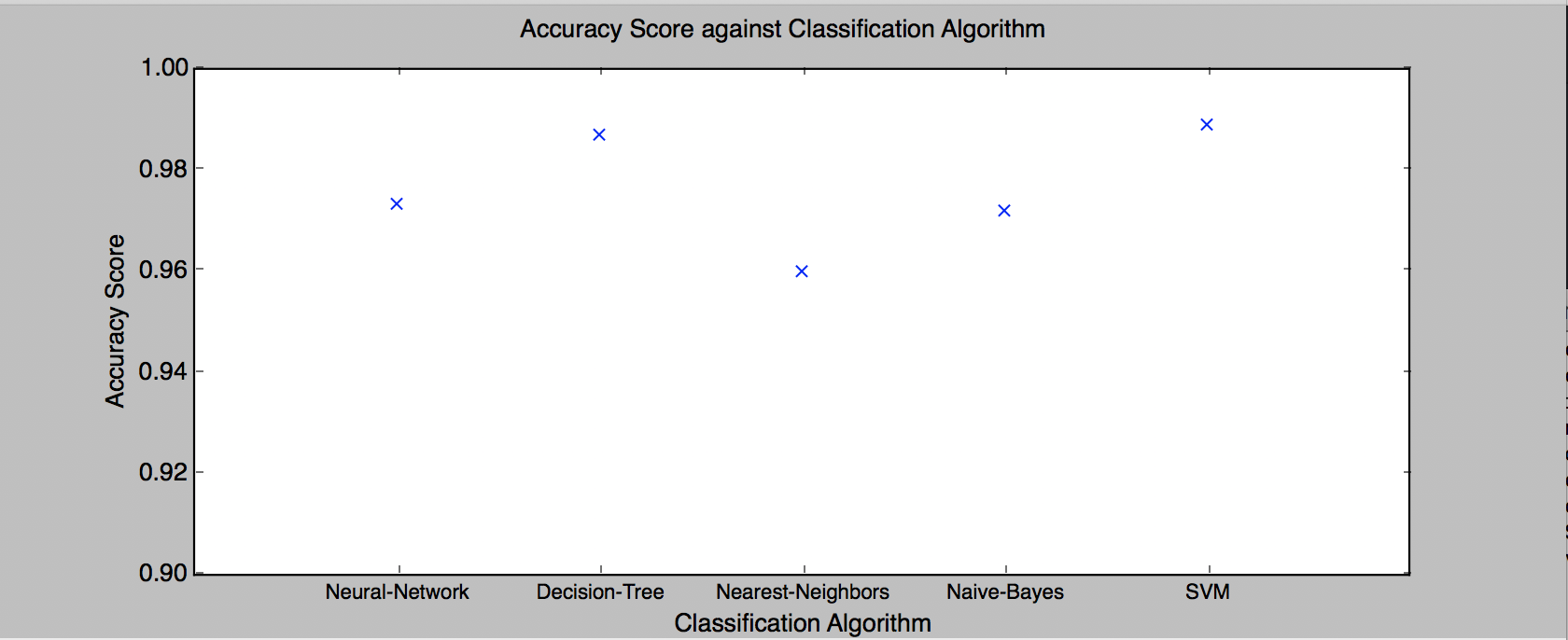


Figure 1: Comparing Classification Algorithms on the IRIS Dataset

In this comparsion we compared

# Feature Identification: Creating The Training Set

## Hypothesizing Features

The files are presented as a pdf format, assuming the files can be converted to a format where we can dissect the file into presence of certain features, the following are the ones hypothesized, which are later tested and improved upon.

### Presence of Links

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Sentences from Abstract

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

### Presence of Repositories

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Presence of Programming Language Names

### Presence of Journal Types

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

### Frequency of Links

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

## Quantifying Features

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

# Extracting the Features And Producing Test Data

## Decision Tree

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

## SVM (Linear/RBF)

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

## Nearest Neighbors

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

## Neural Networks

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

## Random Forest

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.

## Naïve Bayes

We require a bound copy and an electronic copy of your dissertation. .

### Algorithm Summary

We require two canvas bound copies with hard or soft cover. On the spine the lettering should be in the format:

### Application to the classification problem

These should be in Word or PDF format. It is useful to have it on a CD, but if the whole lot can be zipped to under 10MB then you can also email it to your tutor.