MEng Individual Project (JMC)
Interim Report

Daren Sin (ds2912)
CID: 00732331

Supervisor: Panos Parpas

January 27, 2017

1 Introduction

1.1 Schizophrenia, etiology and genes

Schizophrenia is a complex mental disorder that displays an array of symptoms. It is commonly perceived that schizophrenia is a hereditary disease that can be passed down within the family, but some individuals diagnosed with schizophrenia do not have a family member with the disorder [1]. Thus, it is postulated that the heritability of schizophrenia might not be as high as what is commonly believed [2].

Furthermore, there is a strong indication that environmental factors - such as tobacco smoke and viruses - and genetic factors have an influence on the development of psychiatric disorders in an individual [1, 3]. This results in a hypothesis that the epigenetics (see Section 2.1.1) of an individual might have a role to play in the development of schizophrenia (see Section 2.2) [4]. However, exactly how these two factors play a part is still unclear [5].

Moreover, the current research on psychiatric disorders do not receive as much attention as other illnesses such as cancer [6]. Thus, any insight generated from this project would be beneficial to helping us understand psychiatric disorders better.

Overall, this project aims to predict Schizophrenia cases on the basis of epigenetics and epivariations.

1.2 Using machine learning to predict Schizophrenia cases

Using data from a recent study on epigenetics and schizophrenia (see Section 2.3), the project aims to use machine learning to elucidate any statistical regularity in the data, in hope that any insight into the data can help geneticists and psychiatrists understand the etiology of schizophrenia - and indeed, other psychiatric disorders - better.

As a starting point, simple classifiers can be used on the data, to determine the classification accuracy of the data (see Section 2.5). Later on, we can explore other classifiers and techniques which might produce better results.

Previous work on using machine learning on biological data (see Section 2.4) has always been plagued with the *curse of dimensionality*, where the number of biological samples is far lesser than the number of features (or dimensions) of the data (the "$p \gg n$" problem [7]). In our case, we have 847 samples (individuals) with 420374 features, resulting in about 2 gigabytes of data.

Here, we face a potential problem of a similar nature - the data has high dimensions, but not all the genes involved in the study would directly play a part in the classification of the disorder; some genes may only contribute a little to the outcome of the classification. In this case, we would need to perform feature selection to only select features that have significant contribution to the classification outcome (see Section 2.7).

Moreover, linear classifiers may not be able to capture the complexity of the data, as it is hypothesised that subsets of genes - rather than single genes - contribute to the genesis of the disorder [3].

These potential problems make the project interesting, as we cannot simply use ordinary machine learning techniques to manipulate the data. We have to adapt our algorithms and classifiers to suit the complexity and context of the problem.

2 Background

2.1 Biological review

2.1.1 Molecular biology and definitions

This section outlines the necessary biology that will be relevant to the discussion in this project [8, 9, 10, 11].

- **DNA:** Deoxyribonucleic Acid, also known as DNA, is a molecule that contains all the hereditary material in all living things. It serves as the fundamental unit of heredity.
- **DNA bases:** The hereditary information stored in DNA molecules are made up of four bases - Adenine (A), Thymine (T), Cytosine (C) and Guanine (G). These bases pair up in a specific way: A with T and C with G. Along with other types of molecules, these pairs form a nucleotide. Nucleotides are then arranged in a double helix structure.
- **Genes:** A gene is the fundamental building block of heredity. Genes consist of DNA, and encode instructions to produce proteins. These instructions are

used to produce proteins through the process of transcription and translation. This process is often called the “central dogma” of molecular biology.

- **Gene expression:** Gene expressions behave like a switch to determine when and what kind of proteins are produced by cells. All cells in a human being carry the same genome. Thus, gene expression allows cells to specialise into different functionalities (e.g. differentiate between a brain cell and a skin cell).
- **Epigenome:** The epigenome is a set of chemical compounds and modifications that can alter the genome, and thus alter DNA and the proteins that it produces. The epigenome can thus alter the “on/off” action in gene expression and control the production of proteins. The epigenome arises naturally, but can be affected by external factors (e.g. environmental factors, disease), which might explain why even though twins have the same genome, it often happens that one twin inherits a disease, while the other does not [12].
- **DNA methylation:** A common chemical modification is DNA methylation, where methyl groups ($-CH_3$) are attached to the bases of DNA at specific places. These methyl groups switch off the gene which they are attached to in the DNA, and thus no protein can be generated from that gene.

2.2 Relationship between epigenetics and disease

Studies that involve monozygotic twins (twins who share the same set of genomes) are useful to discover the effect of epigenetics on the phenotypes (observable, physical characteristics) of these twins [13].

A study that focuses on monozygotic twins and their susceptibility to disease found that the genes that make up an individual cannot fully explain how likely he/she would be diagnosed with a disease [14]. It is thus interesting to ascertain, using epigenetics data and machine learning, whether epigenetics has any influence on being diagnosed with psychiatric disorders, by detecting any statistical regularity in the data.

2.3 The data

This project makes use of data from a recent genetic-epigenetic analysis of schizophrenia, conducted in 2016 [15]. There are high-throughput methods that enable genomics researchers to perform epigenome-wide association studies (EWAS). In this study in particular, its researchers aimed to use these methods to identify positions in the genome that display DNA methylations associated with environmental exposure and disease. It turns out that there are significant differences in DNA methylation between individuals diagnosed with schizophrenia, and those who were not (controls).

In particular, we are interested in the data offered in “phase 2” of the study, where schizophrenia-associated differentially methylated positions (DMPs) - positions on the genome where there is a difference in patterns of DNA methylation between two sets of genomes - were tested among 847 individuals, 414 of whom were schizophrenia cases.

Throughout this project, we shall identify the data produced from this study as *the data*. At the time of writing, some work has already been done to the data. See section 3.1.

2.4 Cancer classification

There is a significant amount of literature on cancer classification using gene expression data. These works primarily aim to uncover biological or medical insights using biological data obtained from microarrays, which are tools to measure the gene expression of thousands of genes simultaneously [16]. For example, using neural networks, gene expression data can be used to distinguish between tumour types, which helps in solving cancer diagnosis problems [17, 18]. We can draw lessons from these studies to apply to this project.

What is similar about this project and previous work on cancer classification is that the data for both cases are plagued with high dimensionality (*Curse of dimensionality*). For example, in cancer studies, microarrays are used, with a large number of genes (features) but a small number of samples (observations) [18].

Furthermore, only a (small) subset of the features are relevant for the studies, as not all genes are relevant for determining the type of cancer a patient has. This is known as biological noise [19]. As such, a feature/dimensionality reduction on the data has to be performed to select only the relevant genes/features for the classification problem. In other words, the solution for our situation (and also for cancer classification) would ideally be sparse, as we seek to identify the features that are most relevant to the classification.

However, what is fundamentally different about studies on psychiatric disorders and cancer, is that the latter is observable, such that we can know for sure that an individual has cancer using medical methods, such as conducting a blood test. However, it is not obvious that an individual has a psychiatric disorder, as its symptoms might not be straightforward.

For example, the manual "*Diagnostic and statistical manual of mental disorders*" discusses culture-related diagnostic issues of schizophrenia: "the assessment of disorganized speech may be made difficult by linguistic variation in narrative styles across cultures". Furthermore, "ideas that appear to be delusional in one culture (e.g., witchcraft) may be commonly held in another" [20]. These highlight how diagnosing a psychiatric disorder like schizophrenia is not at all easy.

2.5 Review of machine learning classifiers

Eventually, we would need to select an ideal machine learning method to detect statistical regularity in our data. This section reviews machine learning classifiers that might be relevant for this project.

2.5.1 Decision Trees

In our context, the task is to classify the data according to whether a sample (individual) has a psychiatric disorder - in particular, schizophrenia - or not. In other words, the classification task is binary. An intuitive solution is to use decision trees; problems with discrete output values can be solved using decision trees [21].

A decision tree algorithm is capable of sorting the instances - in our context, samples with different features - down the tree until the algorithm reaches a leaf node, during which a classification is given to the node. At each level of the tree, the intermediate node is split according to some attribute.

One variant of the decision tree algorithm is the ID3 algorithm [22]. The ID3 algorithm makes use of a statistical quantitative measure, the information gain, to determine the attribute to classify the samples with. Using definitions from [21], let S be the set of all the samples that we want to classify at a particular node. The samples can also be separated into two groups, those with a positive classification and those with a negative classification. Define the entropy of S as:

$$Entropy(S) \equiv -p_{(+)} \log_2 p_{(+)} - p_{(-)} \log_2 p_{(-)}$$

where $p_{(+)}$ and $p_{(-)}$ represents the proportion of samples with positive and negative classification respectively.

Then, the information gain with respect to the set S and an attribute (feature) A is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where $Values(A)$ is the set of all possible values of attribute A, and $|S_v|$ is the number of elements in the set S with value v for its attribute A.

We then classify the samples in the node according to the attribute with the highest information gain. Intuitively, we want to choose the attribute that can give the most distinct separation between the positive and negative classification (instead of choosing an attribute that, say, splits the samples into half according to their classification).

Although the decision tree algorithm is said to be robust to errors [21] and the resulting decision tree can be easily interpreted, it might be difficult to classify samples according to features that are highly correlated. This also happens to be a potential characteristic of our dataset, and we would expect some of the features to be correlated.

Furthermore, the features of our dataset are vectors of real numbers. We would thus need to discretise the range of real numbers into intervals. This would then give rise to another problem of defining a suitable interval for these values.

2.5.2 Random forest

The random forest method [23], a form of "ensemble learning", is an extension of the decision tree algorithm described above, and it has been used in areas such as multi-class object detection in images [24]. Overall, a random forest algorithm can be outlined as such:

- Split the dataset into distinct subsets.
- Using each subset, train a decision tree using a relevant algorithm, such as the ID3, as outlined above.
- Combine all the trees together to create a forest.
- Suppose we have an unseen sample $\boldsymbol{x}$. Put the $\boldsymbol{x}$ through each tree, and obtain the resulting classification for each tree.
- Based on a "majority vote" system, determine the final classification of $\boldsymbol{x}$; that is, choose the classification that is the most popular among the decision trees.

Even though random forests have been shown to outperform decision trees [25], the limitations of decision trees as described above would still be inherent in the random forest method. Besides, Random Forest requires more parameters in general. For example, we would need to determine the number of trees to be trained. This would require numerical experiments.

It has been shown that the number of trees grow with the number of features that directly affect the classification outcome [26], and we do not know beforehand what these features are. As such, we might potentially have to train a lot of trees. This might take up a lot of memory and time.

So, overall, the decision tree and random forest methods might not be the best methods for our context, even though they are considered to be popular machine learning techniques [25].

2.5.3 Lasso and Elastic net

In Section 1.2, we discussed how, in this project, not only are we seeking low classification errors, we also have to select features/variables in the data that are relevant in producing accurate predictions. An obvious, but naive, solution is to consider all the features in different combinations. But this solution is evidently computationally expensive, much less with data as large as the one we consider in this project.

One method to overcome this problem is by Lasso regression [27], which is a regularised least squares scheme that imposes an l_1-norm penalty on an error function that it tries to minimise. More importantly, in the context of big data and especially this project, the Lasso is an appealing solution because it produces a sparse solution, by shrinking the coefficients of insignificant features to 0.

However, Zou and Hastie [7] examined the limitations of the Lasso method, especially in the context of microarray data. In particular, Lasso has some limitations in variable selection if a subset of features have high correlation with one another. This is precisely a characteristic of genes, as genes often interact with one another.

As a result, Zou and Hastie proposed the *elastic net*, which imposes a linear combination (weighted) of the l_1-norm and the square of the l_2-norm. This method performs feature selection, presents a sparse solution and takes into account variables with high correlation, where groups of correlated variables are not known in advance [28]. Furthermore, Zou and Hastie showed that the elastic net method outperforms Lasso. As such, elastic net might be applicable for our data set.

Besides, we can also utilise the elastic net library in `scikit-learn` implemented in Python. This allows us to experiment with elastic net easily, to see if it would be suitable for our dataset.

2.6 Support vector machines

This section explains and discusses Support Vector Machines[1].

Consider our dataset that comprises m features and n samples. Then, the data can be written as a set: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^m$ is an m dimensional vector that corresponds to the i-th sample. Moreover, $y_i = \pm 1$ is the label of the i-th sample: $y_i = 1$ if the classification is positive (e.g. sample does not have schizophrenia) and $y_i = -1$ otherwise, for $i = 1, \ldots, n$.

Suppose we have data points that correspond to either class 1 or class 2. The Support Vector Machine (SVM) [29] uses a separating hyperplane to distinguish between data points that belong to class 1 and class 2. Using Statistical Learning Theory, by Vapnik et. al [29], the SVM finds the hyperplane with the largest margin between the two classes.

The hyperplane is parameterised with weight vector $\boldsymbol{w}$ and a bias b. We thus solve classification problems using linear models:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b \tag{1}$$

Finding the hyperplane with maximum margin amounts to solving the following optimisation problem:

$$\min_{w,b} \quad \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} \tag{2}$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \tag{3}$$

for $i = 1, \ldots, n$, where, as defined above, $y_i = \pm 1$, depending on the classification of the vector of features $\boldsymbol{x}_i$. We then get the result:

$$\begin{aligned} \boldsymbol{w}^\top \boldsymbol{x}_i + b &\geq 1 \quad \text{if } y_i = 1 \\ \boldsymbol{w}^\top \boldsymbol{x}_i + b &\leq -1 \quad \text{if } y_i = -1 \end{aligned}$$

[1] Most of the Mathematics here is with reference to notes from the Department of Computing, course CO496 - Mathematics for Inference and Machine Learning.

2.6.1 Dual representation

To solve the (primal) optimisation problem in (2) subject to the conditions in (3), we can formulate the following Lagrangian equation:

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \sum_{i=1}^{n} a_i(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1) \tag{4}$$

where $\boldsymbol{a}$ is the n-dimensional vector containing the Lagrangian multipliers ($a_i \geq 0$) corresponding to the inequality conditions in (3). The (primal) optimisation problem in (4) can be written as:

$$\min_{\boldsymbol{w},b} \max_{\boldsymbol{a}\geq 0} L(\boldsymbol{w}, b, \boldsymbol{a}) \tag{5}$$

This can be written as its dual equivalent:

$$\max_{\boldsymbol{a}\geq 0} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{a}) \tag{6}$$

It can shown that:

- The equation L in (4) is convex, and thus any optimal solution found is guaranteed to be the global optimal solution [30].
- The primal (5) and dual (6) problems have the same optimal solutions, if any [31].

To solve the problem in (6), we must first minimise L with respect to $\boldsymbol{w}$ and b for fixed $\boldsymbol{a}$. To do this, we can take the derivative of L with respect to $\boldsymbol{w}$ and b. Then, set the derivatives to 0. Doing this would result in the constraints $a_i \geq 0$ and $\sum_{i=1}^{n} a_i y_i = 0$.

We would then obtain an expression of L with respect to $\boldsymbol{a}$ that we wish to maximise:

$$L(\boldsymbol{a}) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j$$

Combining the above together with the constraints $a_i \geq 0$ and $\sum_{i=1}^{n} a_i y_i = 0$, we would get the following quadratic optimisation problem:

$$\begin{aligned} \max_{\boldsymbol{a}} \quad & \boldsymbol{1}^\top \boldsymbol{a} - \frac{1}{2}\boldsymbol{a}^\top \boldsymbol{K}_y \boldsymbol{a} \\ \text{subject to} \quad & a_i \geq 0, \quad i = 1, \ldots, n \\ & \boldsymbol{a}^\top \boldsymbol{y} = 0 \end{aligned}$$

where $\boldsymbol{1}$ is an n dimensional vector of ones, and $\boldsymbol{K}_y = y_i\, y_j\, \boldsymbol{x}_i^\top \boldsymbol{x}_j$.

While setting the derivatives of L with respect to $\boldsymbol{w}$ and setting to 0, we would obtain the following condition:

$$\boldsymbol{w} = \sum_{i=1}^{n} a_i y_i \boldsymbol{x}_i$$

Substituting this into the linear model equation in (1), we get:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} a_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + b \tag{7}$$

In order to determine the classification of a new point $\boldsymbol{x}$, we simply have to determine the sign of $f(\boldsymbol{x})$ in (7).

2.6.2 Mapping to higher dimensional space

When the relationship between data points in the input space is not linear, the (linear) SVM with the description above would not be able to learn these non-linear relations. This would result in underfitting.

As such, we would need to map the input data points into a higher dimensional space (feature space). We can then build an SVM based on this high dimensional space such that the points in the feature space is linearly separable.

We first define a mapping $\phi : X \to F$ where ϕ is a non-linear mapping from the input space X to a higher dimensional feature space F. We then define the *kernel* function K such that $\forall \boldsymbol{x}, \boldsymbol{y} \in X$,

$$K(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$$

The optimisation problem can then be written as:

$$\begin{aligned} \min_{\boldsymbol{a}} \quad & \frac{1}{2} \boldsymbol{a}^\top \boldsymbol{K}_y \boldsymbol{a} - \boldsymbol{1}^\top \boldsymbol{a} \\ \text{subject to} \quad & a_i \geq 0, \quad i = 1, \ldots, n \\ & \boldsymbol{a}^\top \boldsymbol{y} = 0 \end{aligned}$$

where $\boldsymbol{K}_y = y_i \, y_j \, \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) = y_i \, y_j \, K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Similarly, equation (7), which determines the classification of a new data $\boldsymbol{x}$, can be written as:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} a_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b = \sum_{i=1}^{n} a_i y_i \, \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}) + b$$

Thus, for a new vector $\boldsymbol{x}$, we simply need to test the sign of $f(\boldsymbol{x})$. If $f(\boldsymbol{x})$ is positive, then we can classify it as class 1, and class 2 if $f(\boldsymbol{x})$ is negative.

2.6.3 Slack variables

Real-life data may not be perfectly linearly separable in the feature space $\phi(\boldsymbol{x})$, especially due to the presence of noise [32]. Slack variables, ξ, are introduced to allow some form of error when training data points are misclassified. These slack variables allow data points to be classified on the wrong side of the decision hyperplane, but the further away a point is from the decision boundary, the larger the penalty imposed. We then need one slack variable per input data point [33], defined as such:

- $\xi_i = 0$: data point is correctly classified.
- $0 < \xi_i \leq 1$: data point lies inside the margin, but is on the correct side of the boundary.
- $\xi_i > 1$: data point is wrongly classified.

This is often referred to as the 1-norm soft margin constraint in the literature [32]. Now, we would need to maximise the margin of the hyperplane, while penalising the misclassified points. We can thus formulate our problem as such:

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C \sum_{i=1}^{n} \xi_i \tag{8}$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for } i = 1, \ldots, n \tag{9}$$

where $C > 0$ is the *penalty parameter*. Similarly, to state the dual of (8) subject to the conditions in (9), we need to compute the Lagrangian:

$$L(\boldsymbol{w}, b, \xi_i, a_i, r_i) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} a_i(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} r_i \xi_i$$

where $a_i \geq 0$ and $r_i \geq 0$ are the Lagrangian multipliers.

Similarly, we compute the derivative of the above with respect to $\boldsymbol{w}$, b and ξ_i to get the following dual problem:

$$\min_{\boldsymbol{a}} \quad L(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^\top \boldsymbol{K}_y \boldsymbol{a} - \boldsymbol{a}^\top \boldsymbol{1} \tag{10}$$

$$\text{subject to} \quad \boldsymbol{a}^\top \boldsymbol{y} = 0, \quad 0 \leq a_i \leq C \tag{11}$$

where $K_y = [y_i \, y_j \, \boldsymbol{x}_i^\top \boldsymbol{x}_j]$ and C is the penalty parameter.

Now, suppose we choose to map the input space into a higher dimensional feature space, we simply modify K in the above problem:

$$K_y = [y_i \, y_j \, \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)]$$

We then need to solve the quadratic optimisation problem in (10).

2.6.4 Model selection

There are 4 widely used kernels:

- Linear kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$
- Polynomial kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\gamma \, \boldsymbol{x}_i^\top \boldsymbol{x}_j + r)^d$
- Radial basis function (RBF): $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \, \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$
- Hyperbolic tangent kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\gamma \, \boldsymbol{x}_i^\top \boldsymbol{x}_j + r)$

where r, d and $\gamma > 0$ are kernel parameters.

First, we would pick a kernel before the training process. Then, using a procedure such as k-fold cross validation, we would then find the most optimal kernel and penalty parameters (C).

For example, if we choose the RBF kernel, we would perform cross validation to obtain the most optimal parameters γ and C.

A study in [34] noted that the RBF kernel has less numerical difficulties than the polynomial kernel. We note that, $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in X$,

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \geq 0 \quad \text{and} \quad \gamma > 0 \quad \Rightarrow \quad 0 < K(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1$$

On the other hand, for large d, the polynomial kernel might go to infinity or close to 0. Furthermore, the hyperbolic tangent kernel is not valid under certain kernel parameters. We should then focus on the linear and RBF kernels.

Although the RBF kernel is more commonly chosen than the rest of the kernels listed above, the study also revealed that if the number of features is large, using the linear kernel may just suffice, as the nonlinear mapping provided by the RBF kernel may not necessarily improve performance. Furthermore, using the linear kernel would mean that we only need to investigate the optimal C penalty parameter value.

Nevertheless, we should view this conclusion with scepticism and still proceed to investigate the better kernel method (between linear and RBF) by using cross-validation, as the conclusion made in [34] might be data-dependent.

2.6.5 Libraries available for SVM

Chang and Lin [35] developed LIBSVM, a library for SVMs at the National Taiwan University. LIBSVM utilises the Sequential Minimal Optimisation (SMO) algorithm to solve the quadratic optimisation problem in (10). SMO allows the problem to be solved analytically [33]. Furthermore, the library can be interfaced to other programming languages like Java, MATLAB and Python.

The Python library `scikit-learn` also allows SVMs to be trained for classification purposes (`SVC`) and allows various kernels to be used. Furthermore, the library uses LIBSVM internally to handle computations.

We can try both libraries to see if either one outperforms the other, in terms of accuracy and computation time.

2.7 Feature selection

As mentioned in section 1.2, the data suffers a *curse of dimensionality*, where the number of features exceeds the number of samples available. This usually causes *over-fitting* on the classifier. This refers to the situation where the classifier can classify the training data perfectly, but performs poorly with unseen data.

Furthermore, there is a possibility that not all of the genes in our data set have a role to play in the classification of schizophrenia. As such, the feature selection method is necessary to select the significant features and eliminate the others. Besides, once we obtain only the relevant features, it would be less computationally expensive to train our classifier just on these features.

In 2010, a study was conducted to investigate the impact of feature selection methods (Kernel Principal Component Analysis (PCA), Greedy Kernel PCA and Generalised discriminant analysis) on real microarray gene expression data. It concluded that applying feature selection methods improved the performance of the SVM classifier [36]. This strengthens our belief that the feature selection process would enable our classifier to perform better with our data set.

2.7.1 Feature selection methods

Feature selection methods can be classified as [37, 38]:

- **Filters:** Ranks genes based on a univariate measure, and selects only the top ranking genes. No learning is involved. Interaction between features is also not considered.
- **Wrappers:** Use learning to decide which subset(s) of features are relevant.
- **Embedded:** Incorporates feature selection process into the classifier.
- **Hybrid:** Combination of the above approaches.

Several reviews of feature selection methods on microarray data [37, 38] came to the conclusion that filter methods are not as appropriate as the others, as, for example, the method might rank similar features highly, and thus pose a lot of redundancy after processing the data. Besides, filter methods do not take into account the classifier that we train, and they do not consider the interaction between features [39].

Furthermore, since embedded methods are executed together with the training process of the classifier, the resulting optimal set of features is coupled with the classifier that is trained. In other words, this set of features is classifier dependent [38]. Since we want the classifier to eventually generalise to other epigenetic datasets (see Section 3.6 on possible extensions), we might want to avoid this class of methods.

2.7.2 Wrappers

In general, wrapper methods are expected to yield better features. However, one disadvantage of wrappers is that it is computationally expensive, and the cost of the method increases with the feature space.

Wrapper methods can be classified into optimal and suboptimal search algorithms [40]. The former searches the whole space of features and their subsets, while the latter only considers part of this space. The optimal algorithms can give us better results since they consider all combinations of the features, but they are obviously computationally too expensive. Although the suboptimal algorithms does not guarantee the best result, they are much more practical to execute. Such algorithms include:

- **Sequential forward selection:** start from an empty set of features. At each iteration, add in a new feature that maximises the selection criterion (e.g. training error produced by classifier). Stop when the criterion stops improving.
- **Sequential backward selection:** start from the whole set of features. Delete one feature at a time, until the number of features required is reached.

There are also "floating" versions of the above selection methods, which allows backtracking to remove (or add) features that might improve the selection criterion [41].

There also has been work that combines genetic algorithms with SVMs [40]. However, these algorithms are deemed to be more time consuming, although it can cover more combinations of feature subsets (see Section 3.6 for possible extensions).

2.7.3 Novel methods in feature selection

In [39], Tang et. al suggested two novel methods for feature selection:

- Gradient-based leave-one-out gene selection (GLGS) algorithm
- Leave-one-out calculation sequential forward selection (LOOCSFS) algorithm

The GLGS uses a gradient-based algorithm, while the LOOCSFS incorporates the Sequential Forward Selection method and uses the leave-one-out cross validation error (LOOE) of the SVM as a selection criterion. Experiments on different datasets by Tang et. al seem to come to the conclusion that the GLGS algorithm might be a good choice for small number of samples (individuals in our context), with large d and t, where d is the number of features and t is the number of features to be selected. We can take this approach in our context, and see if either method outperforms the other with our data.

Furthermore, Tang et. al pointed out that the number of features to be selected by the algorithm (t) must be defined beforehand for the algorithm to work. The experimenters set this number to be 100 for all the datasets that were explored. However, the paper later recommends different values of t for two of the datasets that were used.

This suggests that t might be dependent on the dataset used. We can thus select t using the approaches listed by Tang et. al:

- Terminate the algorithm if the selection criterion does not improve much when more features are added.
- Plot a graph of the error against t. We can inspect visually the most optimal value of t.

3 Project plan

3.1 Work that has been done

3.1.1 Understanding the data

The first step that was done for this project was to find ways to understand the data. At the time of writing this report, the Python library `pandas` is used to manipulate the csv data set. `pandas` is a high-performance data analysis tool, which is suitable to be used on large datasets.

On a typical laptop, attempting to read the csv file row by row would result in a memory error in Python. The `pandas` library allows the file to be read chunk by chunk. This method then enables us to find out exactly how many rows and columns there are in the csv file. Moreover, we found that the rows represent the sites in the DNA that can be methylated (features), while the columns represent the samples (individuals).

3.1.2 Data preprocessing

Furthermore, in the National Center for Biotechnology Information (NCBI) database, a "series matrix" file explains what the data in the csv file represents. In particular, the columns (individuals) can be divided into the "control" (individuals with no schizophrenia) and the "cases" (individuals known to have schizophrenia). This is labelled by `disease_status=1` and `disease_status=2` respectively. This is helpful for classification, as we know beforehand what the label of each column (individual) is. This is necessary for supervised learning algorithms to be applied.

We also have to convert the data that can be read by our classifier. For example, we have to convert the csv file into matrices, and convert the classification (control or case) into a (binary) vector.

3.2 Performance of classifiers

Similar to previous work on cancer classification, we would need to employ different types of supervised learning algorithms on the dataset. Although several experiments have shown that support vector machines (SVMs) are superior compared to other methods [42], we should still consider other classifiers, such as elastic net, to ascertain their suitability on the data. We then need to determine the classification accuracy obtained by each classifier, and select the best classifier for our data.

The classification accuracy of each classifier can be determined using methods such as k-fold or leave-one-out cross validation.

Furthermore, classifiers such as the SVM requires several parameters. We would then need to run numerical experiments to find out what the optimal values for these parameters are. These optimal values should give us the best classification accuracy.

Overall, this entire process of training the right classifier can be summarised in three steps:

- Training of the classifier using data
- Model selection: finding the best set of parameters for the classifier
- Classification error: accessing the performance of the classifier

In the case that we use the SVM as our ideal method, we also have to select the most appropriate kernel.

3.3 Feature selection

As briefly mentioned in section 2.7, once we get the classifier working on the dataset to a satisfactory accuracy, we can then proceed to perform feature selection on the data. This allows us to select only the most informative and relevant genes. It also helps to reduce the size and amount of data needed for the classifier.

I intend to start with the GLGS and LOOCSFS (see section 2.7.3) algorithms, as they have been proven to work well with biological data with small sample size and large number of features. We can then also find the most optimal value of t, which was defined as the number of features to be selected by the algorithm.

We can then compare the performance of the classifier when it is training on the full data, with when it is trained only on the selected features. We can also investigate if other classifiers would perform better when the feature set is reduced.

3.4 Biological implications

Once the above are obtained, we can then see if we are able to select the most informative features. This has biological implications, as we can understand which part(s) of the genome has the most impact on the outcome of data (i.e. whether an individual has schizophrenia).

Next, after any insight found using our preferred classifier on the data set, we can extend our methodology to other datasets. In this way, we can investigate how well the classifier (together with the optimal parameters) generalises to other datasets.

3.5 Timeline and planning of project

The significant dates relevant for this project are stated in the table below.

Date	Significant event
27 Jan	Submission of interim report
17 Feb	Project review deadline
13 Mar	Spring term exam revision
20 Mar	Spring term exam
25 Mar	Start of Easter break Math summer exam revision
30 Apr	End of Easter break Math summer exam
15 May	Project health check-up
31 May	Latest date to start final report (3 weeks period to work on report)
21 Jun	Final report due

I would give myself the following dates to complete the following:

- *Before 17 February:* Complete experiments with classifiers, determine the best one (in terms of accuracy, computational cost etc.) (3 weeks).
- *Before 3 March:* Experiment with parameters of classifier(s); determine the most optimal set of parameters (2 weeks).
- *Before 13 March:* Experiment with feature selection; we can see if it is possible to select the most informative features (1+ weeks).

The following are the fundamental components of this project. It is evident that there is plenty of time (including the Summer term) to complete this. In case any of the above steps takes a longer time than anticipated, there will be enough time to finish them. I would also need to consider the ongoing courseworks and deadlines during Spring term.

Besides, I would have 1 Math exam in Summer, whose date is not finalised yet, but it should occur in the month of May. This also means that I should be able to juggle between the exam and this project fairly well during the Easter break and in Summer.

Taking the above into consideration, if the above schedule works, I would then proceed with the extensions (see Section 3.6).

3.6 Possible extensions

First, more complex algorithms, such as the neural network, which requires more time to train due to the high dimensionality of the data, can be used. We can then, similarly, compare the classification accuracy of the neural network, and decide if we should adopt the classifier that was deemed the best, or the neural network. We also need to take into account the time that the network takes to be trained.

Second, we can create a toolbox or script that consists of an analysis pipeline for biomedical researchers to analyse similar epigenetic data. In other words, we can automate the process of manipulating the data and training the classification algorithm, such that future epigenetic data can be similarly analysed. We can then obtain biological and/or medical insight from the data much quicker. However, to do this, we must ensure that the algorithm generalises to other datasets, not just the data that we use in this project.

Next, when conducting k-fold cross validation when selecting the most optimal set of parameters for our classifier, we would fix the value of k beforehand, typically 5, 10 or 20 [43]. However, we can go one step further to consider k as a hyperparameter, as suggested in [43]. We can then find the most optimal k by, say, using a grid search.

Moreover, we can extend our investigation into the best feature selection method for our dataset. In particular, we can choose to focus our attention on probabilistic wrappers. These involve genetic algorithms to select the most optimal set of features [38]. As discussed above, feature selection with genetic algorithms tend to take more time to train, so this can only be done after our initial investigation of the ideal feature selection method.

4 Evaluation plan

4.1 An investigation into the use of epigenetic data

Essentially, this project is about investigating whether the use of epigenetic data is relevant in helping us to identify individuals with psychiatric disorders. In other words, the project might conclude that epigenetic data is not able to help us to predict individuals that potentially have schizophrenia. Nevertheless, this can also be seen as a beneficial development.

4.2 Extending beyond our dataset

The most important measure of success is that our classifier can generalise beyond the data. In other words, even if we use the classifier on other sources of data, our classifier would still give correct predictions and classify the examples with high accuracy.

To achieve this, we would need to train a classifier that is independent of our current dataset, and that would be a challenge for this project.

4.3 Creating an analysis pipeline

If time permits (see Section 3.6 on extensions) us to create an analysis pipeline for epigenetic data, we would need to ensure that it is user-friendly, and the analysis it produces is biologically relevant and useful for further analysis.

To do this, we would need several experts in the field to try out the pipeline. Feedback can then be obtained from them (preferably iteratively) to improve the design of the pipeline. To make the pipeline user friendly is especially important, especially if we decide to create a GUI-based toolbox.

References

[1] National Institute of Mental Health. Schizophrenia, 2016. Available from https://www.nimh.nih.gov/health/topics/schizophrenia/index.shtml.

[2] O. J. Bienvenu, D. S. Davydow, and K. S. Kendler. Psychiatric 'diseases' versus behavioral disorders and degree of genetic influence. *Psychological medicine; Psychol.Med.*, 41(1):33–40, 2011. ID: TN_cambridgeS003329171000084X.

[3] Schizophrenia.com. Heredity and the genetics of schizophrenia, 2004. Available from http://www.schizophrenia.com/research/hereditygen.htm.

[4] Bob Weinhold. Epigenetics: The science of change. *Environmental health perspectives*, 114(3):A160–A167, 2006. ID: TN_pubmed_central1392256.

[5] Florence Thibaut. Why schizophrenia genetics needs epigenetics: a review. *Psychiatria Danubina*, 24(1):25, 2012. ID: TN_medline22447081.

[6] Heidi Ledford. If depression were cancer. *Nature*, (515):182–184, 2014. Available from http://www.nature.com/news/medical-research-if-depression-were-cancer-1.16307.

[7] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. ID: TN_wj10.1111/j.1467-9868.2005.00503.x.

[8] Lister Hill National Center for Biomedical Communications. *Help Me Understand Genetics*. 2017. Available from https://ghr.nlm.nih.gov/primer.

[9] National Human Genome Research Institute. Epigenomics, 2016. Available from https://www.genome.gov/27532724/.

[10] YourGenome. What is gene expression?, 2016. Available from http://www.yourgenome.org/facts/what-is-gene-expression.

[11] Heidi Chial, Carrie Drovdlic, Maggie Koopman, Sarah Catherine Nelson, Angela Spivey, and Robin Smith. Essentials of genetics, 2014. Available from http://www.nature.com/scitable/ebooks/essentials-of-genetics-8.

[12] University of Utah. Insights from identical twins. Available from http://learn.genetics.utah.edu/content/epigenetics/twins/.

[13] Arturas Petronis. Epigenetics and twins: three variations on the theme. *Trends in Genetics*, 22(7):347–350, 2006. ID: TN_sciversesciencedirect_elsevierS0168-9525(06)00126-0.

[14] Pernille Poulsen, Manel Esteller, Allan Vaag, and Mario F. Fraga. The epigenetic basis of twin discordance in age- related diseases. *Pediatric research*, 61(5):38R, 2007. ID: TN_medline17413848.

[15] Eilis Hannon, Emma Dempster, Joana Viana, Joe Burrage, Adam R. Smith, Ruby Macdonald, David St Clair, Colette Mustard, Gerome Breen, Sebastian Therman, Jaakko Kaprio, Timothea Toulopoulou, Hilleke E. Hulshoff Pol, Marc M. Bohlken, Rene S. Kahn, Igor Nenadic, Christina M. Hultman, Robin M.

Murray, David A. Collier, Nick Bass, Hugh Gurling, Andrew McQuillin, Leonard Schalkwyk, and Jonathan Mill. An integrated genetic-epigenetic analysis of schizophrenia: evidence for co-localization of genetic associations and differential dna methylation. *Genome biology*, 17(1):176, 2016. ID: Hannon2016.

[16] W. P. Kuo, E. Y Kim, J. Trimarchi, T. K Jenssen, S. A. Vinterbo, and L. Ohno-Machado. A primer on gene expression and microarrays for machine learning researchers. *Journal of Biomedical Informatics*, 37(4):293–303, 08 2004.

[17] A. Bharathi and A. M. Natarajan. Microarray gene expression cancer diagnosis using machine learning algorithms. In *3rd IEEE International Conference on Signal and Image Processing, ICSIP 2010, December 15, 2010 - December 17*, pages 275–280, Chennai, India, 2010 2010. Bannari Amman Institute of Technology, Tamilnadu (State), India, IEEE Computer Society. Compilation and indexing terms, Copyright 2016 Elsevier Inc.; T3: Proceedings of the 2010 International Conference on Signal and Image Processing, ICSIP 2010.

[18] Chang Kyoo Yoo and Krist V. Gernaey. Classification and diagnostic output prediction of cancer using gene expression profiling and supervised machine learning algorithms. *Journal of Chemical Engineering of Japan*, 41(9):898–914, 2008. Compilation and indexing terms, Copyright 2016 Elsevier Inc.

[19] Y. Lu and J. Han. Cancer classification using gene expression data. *Information Systems*, 28(4):243–68, 06 2003.

[20] American Psychiatric Association. Diagnostic and statistical manual of mental disorders, 2013. ID: 44IMP_ALMA_DS5172131690001591.

[21] Tom M. (Tom Michael) Mitchell 1951. *Machine learning.* International 1997 edition, 1997. Includes bibliographical references and index.; ID: 44IMP_ALMA_DS2143719110001591.

[22] J.R. Quinlan. Induction of decision trees, 1986. ID: RS_60168743381inductionofdecisiontrees.

[23] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ID: TN_springer_jour1010933404324.

[24] Juergen Gall, Nima Razavi, and Luc Van Gool. An introduction to random forests for multi-class object detection. In *15th International Workshop on Theoretical Foundations of Computer Vision, June 26, 2011 - July 1*, volume 7474 LNCS, pages 243–263, Dagstuhl Castle, Germany, 2011 2012. Computer Vision Laboratory, ETH Zurich, SwitzerlandMax Planck Institute for Intelligent Systems, GermanyESAT/IBBT, Katholieke Universiteit Leuven, Belgium, Springer Verlag. Compilation and indexing terms, Copyright 2016 Elsevier Inc.; T3: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

[25] Songul Cinaroglu. Comparison of performance of decision tree algorithms and random forest: An application on oecd countries health expenditures. *International Journal of Computer Applications*, 138(1):37–41, March 2016.

[26] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, December, 2002.

[27] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, pages 267–288, 1996.

[28] C. De Mol, E. De Vito, and L. Rosasco. Elastic-net regularization in learning theory. *Journal of Complexity*, 25(2):201–30, 04 2009.

[29] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–99, 1999.

[30] Edwin K.P. Chong and Stanislaw H. Zak. *Convex optimization problems.* An introduction to optimization. Wiley, Hoboken, New Jersey, 4th; fourth edition, 2013.

[31] Edwin K. P. Chong and Stainlaw H. Zak. *Duality.* An introduction to optimization. Wiley, Hoboken, New Jersey, 4th; fourth edition, 2013. Includes bibliographical references and index.; ID: dedupmrg214935921.

[32] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines: and other kernel-based learning methods.* Cambridge University Press, Cambridge, U.K. ; New York, 2000. Includes bibliographical references (p. 173-186) and index.; ID: 44IMP_ALMA_DS2146175590001591.

[33] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, New York, 2006. Includes Bibliography: p. 711-728 and index; ID: 44IMP_ALMA_DS2144511690001591.

[34] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, 2016. National Taiwan University, Taipei 106, Taiwan.

[35] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[36] C. M. M. Wahid, A. B. M. S. Ali, and K. Tickle. Impact of feature selection on support vector machine using microarray gene expression data. In *2009 Second International Conference on Machine Vision (ICMV 2009)*, pages 189–93, Piscataway, NJ, USA, 28-30 Dec. 2009 2009. Sch. of Comput. Sci., CQ Univ., QLD, Australia, IEEE. T3: Proceedings of the 2009 Second International Conference on Machine Vision (ICMV 2009);.

[37] P. K. Ammu and V. Preeja. Review on feature selection techniques of dna microarray data. *International Journal of Computer Applications*, 61(12), 2013.

[38] Z. M. Hira and D. F. Gillies. A review of feature selection and feature extraction methods applied on microarray data, 2015. ID: 44IMP_DSP_DS10044/1/25192.

[39] E. K. Tang, PN Suganthan, and Xin Yao. Gene selection algorithms for microar-

ray data based on least squares support vector machine. *BMC Bioinformatics*, 7:95, 2006.

[40] Li Li, Wei Jiang, Xia Li, Kathy L. Moser, Zheng Guo, Lei Du, Qiuju Wang, Eric J. Topol, Qing Wang, and Shaoqi Rao. A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. *Genomics*, 85(1):16–23, 2005. ID: TN_sciversesciencedirect_elsevierS0888-7543(04)00271-X.

[41] Andrew R. Webb and Keith D. Copsey. *Statistical Pattern Recognition.* Chichester, UK. ID: TN_wilbookl10.1002/9781119952954.

[42] Michael P. S. Brown, David Lin, Terrence S. Furey, David Haussler, Charles Walsh Sugnet, Manuel Ares Jr., William Noble Grundy, and Nello Cristianini. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262–267, 2000. ID: TN_scopus2-s2.0-0034602774.

[43] Davide Anguita, Luca Ghelardoni, Alessandro Ghio, Luca Oneto, and Sandro Ridella. The 'k' in k-fold cross validation. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, April 2012.