


5-1-2016

# Data Driven Sample Generator Model with Application to Classification

Alvaro Emilio Ulloa Cerna

Follow this and additional works at: [https://digitalrepository.unm.edu/math\\_etds](https://digitalrepository.unm.edu/math_etds)

 Part of the [Applied Mathematics Commons](#), [Mathematics Commons](#), and the [Statistics and Probability Commons](#)

---

## Recommended Citation

Ulloa Cerna, Alvaro Emilio. "Data Driven Sample Generator Model with Application to Classification." (2016).  
[https://digitalrepository.unm.edu/math\\_etds/82](https://digitalrepository.unm.edu/math_etds/82)

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at UNM Digital Repository. It has been accepted for inclusion in Mathematics & Statistics ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact [disc@unm.edu](mailto:disc@unm.edu).

**Alvaro Emilio Ulloa Cerna**

*Candidate*

**Mathematics and Statistics Department**

*Department*

This thesis is approved, and it is acceptable in quality and form for publication:

*Approved by the Thesis Committee:*

**Erik Erhardt**

, Chairperson

**Li Li**

**Marios Pattichis**

# Data Driven Sample Generator Model with Application to Classification

by

**Alvaro Emilio Ulloa Cerna**

B.S., Electrical Engineering, Pontifical Catholic University of Peru,  
2010

M.S., Electrical Engineering, University of New Mexico, 2013

THESIS

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Master of Science  
Statistics

The University of New Mexico

Albuquerque, New Mexico

April, 2016

# Dedication

*To my wife, Jessica, and son, Gabriel, for their unconditional support and patience.*

*To my parents and family that support me unconditionally.*

*To the memory of my grandparents: Donato, Ofelia, and Florencia.*

# Acknowledgments

I would like to thank my advisor, Dr. Erik Erhardt, for his support, and advice. Thanks for introducing me to the world of statistics.

I also appreciate the feedback from the medical imaging analysis lab at MRN. As well as Dr. Calhoun's support and Dr. Plis' guidance.

Thanks to Sergey, Eduardo, and Mohammad for helping me get into machine learning.

I would also like to thank my advisor from the EE department, Dr. Pattichis, for his feedback and interest in the project.

# Data Driven Sample Generator Model with Application to Classification

by

**Alvaro Emilio Ulloa Cerna**

B.S., Electrical Engineering, Pontifical Catholic University of Peru,

2010

M.S., Electrical Engineering, University of New Mexico, 2013

M.S., Statistics, University of New Mexico, 2016

## **Abstract**

Despite the rapidly growing interest, progress in the study of relations between physiological abnormalities and mental disorders is hampered by complexity of the human brain and high costs of data collection. The complexity can be captured by machine learning approaches, but they still may require significant amounts of data.

In this thesis, we seek to mitigate the latter challenge by developing a data driven sample generator model for the generation of synthetic realistic training data. Our method greatly improves generalization in classification of schizophrenia patients and healthy controls from their structural magnetic resonance images. A feed forward neural network trained exclusively on continuously generated synthetic data produces the best area under the curve compared to classifiers trained on real data alone.

# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Structural Magnetic Resonance Imaging . . . . .	3
1.1.1 SMRI Preprocessing . . . . .	5
1.2 Machine Learning . . . . .	7
1.2.1 Deep Neural Networks . . . . .	8
1.3 Motivation for Current Work . . . . .	10
1.4 Thesis Statement . . . . .	10
1.5 Contributions . . . . .	11
1.6 Thesis Overview . . . . .	11
<b>2 Machine learning</b>	<b>13</b>
2.1 Classification methods . . . . .	13

## Contents

2.1.1	Nearest neighbors . . . . .	13
2.1.2	Decision Tree . . . . .	15
2.1.3	Random Forest . . . . .	17
2.1.4	Naive Bayes . . . . .	17
2.1.5	Logistic Regression . . . . .	19
2.1.6	Support Vector Machines . . . . .	20
2.1.7	Multilayer Perceptron . . . . .	21
2.1.8	Majority Voting Classifier . . . . .	22
2.1.9	Implementation . . . . .	22
2.2	Dataset, SMRI . . . . .	24
2.2.1	Participants . . . . .	24
<b>3</b>	<b>Sample generator model</b>	<b>27</b>
3.1	Matrix Factorization . . . . .	28
3.1.1	Principal Component Analysis . . . . .	29
3.1.2	Independent Component Analysis . . . . .	30
3.2	Random variable samplers . . . . .	33
3.2.1	Rejection sampling . . . . .	33
3.2.2	Multivariate Normal . . . . .	36
3.3	Data driven sample generator . . . . .	37



*Contents*

<b>4</b>	<b>Application to SMRI classification</b>	<b>39</b>
4.1	Data analysis . . . . .	40
4.1.1	Data set . . . . .	40
4.1.2	Linear model . . . . .	40
4.1.3	Results . . . . .	42
4.2	Classification results . . . . .	43
4.3	Size effect for data generator . . . . .	47
<b>5</b>	<b>Discussion</b>	<b>49</b>
<b>6</b>	<b>Future Work</b>	<b>54</b>
	<b>Appendices</b>	<b>55</b>
<b>A</b>	<b>Example of analytic solution for optimal envelop function</b>	<b>56</b>
<b>B</b>	<b>Code samples</b>	<b>59</b>
B.1	Baseline results . . . . .	59
B.2	Multilayer Perceptron . . . . .	60
B.3	Brain graphics . . . . .	60

# List of Figures

1.1	Siemens magnetic resonance scanner. Example of brain scanning for SMRI. Retrieved from <a href="http://www.siemens.co.uk">http://www.siemens.co.uk</a> . . . . .	4
1.2	Raw SMRI example. This image was taken at the Mind Research Network Institute and belongs to the author of this thesis. . . . .	5
1.3	Example diagram of a single layer neural network with 3 input features, 4 hidden units, and 3 outputs. . . . .	9
1.4	Example diagram of a multi-layer neural network with 4 input features, 3 hidden layers of 5 units each, and 3 output. . . . .	9
2.1	Nearest neighbors example for $k = 3$ . The green sample will be assigned class B because there are 2 samples on class B and 1 sample in class A . . . . .	14
2.2	Decision tree example for coffee classification. Note that the order of the decisions determines the importance of a feature. . . . .	16
2.3	Bayes classifier example on Fisher's iris dataset. Obtained from <a href="http://www.mathworks.com/help/stats/fitnaivebayes.html">http://www.mathworks.com/help/stats/fitnaivebayes.html</a> . . . . .	18

*List of Figures*

2.4	Example of a linear decision boundary for a support vector machine. (a) The green lines denote some of the infinity of planes that can divide blue samples from red samples in the feature space of $\mathbf{x}$ . (b) The green line shows the optimal plane which has the maximum margin. Obtained from <a href="http://docs.opencv.org">http://docs.opencv.org</a> . . . . .	20
2.5	Moon dataset for classifier evaluation. . . . .	23
2.6	Moon dataset results on various classifiers. . . . .	24
2.7	Predicted labels across all 10 folds for each classifier . . . . .	26
3.1	Visual example of principal component analysis decomposition. . .	30
3.2	Visual example of an independent component analysis decomposition. The plot shows the (a) true sources, (b) mixed sources, (c) estimated sources computed with ICA, and (d) estimated sources computed with PCA . . . . .	31
3.3	Rejection sampling efficiency visualized. (a) Plot of density function to sample from, (b) histogram of accepted samples, (c) plot of rejected samples in blue, and accepted samples in green for $e(x) = 0.3 \times \text{Uniform}(0, 1)$ , and (d) same plot as in (c) with $e(x) = 1.5 \times \chi^2(4)$ . 35	35
3.4	Data driven generator block diagram. The dataset is factorized into components and loading matrix $A$ . The RV block denotes the RV generator that fits from $A$ and generates new samples to reconstruct the synthetic dataset. . . . .	38
4.1	(a) Image taken from the author of this thesis as an example of a raw SMRI image. (b) Image after pre-processing steps described in Section 2.2.1. (c) Mask used to keep intracranial voxels. . . . .	40

*List of Figures*

4.2	Three-way ANOVA results. Voxels passing fdr correction for multiple comparison at the 0.01 level for the (a) diagnosis, (b) age, and (c) gender effects. Significant interactions between (d) gender and diagnosis; (e) age and diagnosis; and (f) age, gender, and diagnosis. We show the significance as $-\log_{10}(p)$ . . . . .	43
4.3	Three dimensional view of all effects on the GMC mean. Red marks the diagnosis effect after FDR correction at 0.01 level. Blue shows the age effect after FDR correction at 0.01 level. Green shows the age-diagnosis effect at 0.01 level. Pink shows the gender-diagnosis effect at 0.01 level. White shows the three-way interaction effect at 0.01 level. . . . .	44
4.4	Average classification score results for raw, ICA reduced, PCA reduced, and augmented data grouped by type of classifier. . . . .	47
4.5	Size effect for various classification methods trained on synthetic data. . . . .	48
A.1	Beta(2,2) probability density function. . . . .	56

# List of Tables

4.1	Participants demographics distribution. . . . .	41
4.2	Participants demographics distribution for three factors: age, gender, and diagnosis. . . . .	41
4.3	Three way ANOVA group means for the main effects of schizophrenia dataset. . . . .	44
4.4	Three way ANOVA group means for the effects of interactions on schizophrenia dataset. . . . .	45
4.5	Classification methods and parameters for grid search . . . . .	46
4.6	Classification results on raw data, ICA reduced, PCA reduced, and augmented dataset. . . . .	46

# Chapter 1

## Introduction

Mental illnesses alter normal behavior and may provoke hallucinations in individuals. Depending on severity, they can also impair the person and significantly degrade their quality of life. According to the national survey on drug use and health for Behavioral Health Statistics and Quality (2015), there were an estimated 9.8 million adults aged 18 or older in the United States with severe mental illness. This number represented 4.2% of all U.S. adults. Thus, the need to better understand, diagnose, and treat these disorders generates a vast interest in the study of mental illnesses at the behavioral and biological levels.

Even though the literature presents many efforts of multidisciplinary areas to understand the underlying mechanisms of the brain, it remains an open problem. Researchers use several data modalities, and a wide range of data analysis methods (Ozer et al., 2008; Müller et al., 2011; Frodl and Skokauskas, 2012) to collect evidence in favor of a study's hypothesis, nonetheless, both data modalities and methods remain a challenge due to various factors (Akil et al., 2011).

Current technology allows us to extract brain behavior information in a non-invasive manner by exploiting measurable electrical properties of the brain. These

## *Chapter 1. Introduction*

electrical properties are magnetic and electric potential fields. Magnetic fields can be measured with magnetic resonance scanners that induce a strong magnetic field to the brain and provides 3D images of brain tissue contrast, or magneto-encephalograms that sense fields triggered by brain activity. On the other hand, the electric potential is measured on the scalp, where probes measure voltage differences that respond to brain activity. All these technologies bring a lot of information to explore and help investigate the brain, however, at the time of writing this thesis, these imaging technologies are still very sophisticated and expensive.

State-of-the-art methods used for brain data analysis are often challenged by the limited sample size that the expensive data collection impose to the problem. The low number of data samples and its effect on statistical models is a well studied problem. Particularly, in machine learning, an area focused in automatic data analysis, the lack of data causes model over-fitting, meaning that the learned model is more influenced by noise variations than the data itself.

Some attempts to overcome the effects of over-fitting are in the form of norm regularization (Hoerl and Kennard, 1970; Tibshirani, 1996; Zou and Hastie, 2005), dropout (Srivastava et al., 2014), and hypothesis-based analysis. The norm regularization methods seek to minimize the norm of model parameters, following the assumption that the parameter space is often sparse. Classic examples of this technique are Ridge regression (Hoerl and Kennard, 1970), Lasso regression (Tibshirani, 1996), and Elastic net (Zou and Hastie, 2005). A more recent method, called Dropout, was proposed in the setting of neural networks. The dropout technique showed promising results on several applications (Pham et al., 2014; Krizhevsky et al., 2012). It temporarily removes a set of parameters and chooses randomly a new set for each training step. As Srivastava et al. (2014) state, this prevents parameters from co-adapting too much. Training several smaller sub-models prevents the big model from not generalizing for unseen data. Finally, another popular approach is to hypothe-

## *Chapter 1. Introduction*

size areas of interest prior to the analysis by removing unwanted features based on previous knowledge.

In this Master's thesis, we focus on the analysis of images from magnetic resonance scanners called structural magnetic resonance images (SMRI) from a Schizophrenia dataset. We also propose a novel methodology that improves SMRI classification accuracy by acting as a new form of regularization.

This introductory chapter is organized as follows. In Section 1.1 we introduce SMRI data and the pre-processing steps. In Section 2, we briefly present machine learning and the concept of deep neural networks. Then, in the remaining sections we present the motivation for the current work, the thesis statement, our contribution, and the thesis overview.

### **1.1 Structural Magnetic Resonance Imaging**

SMRI is a technique for brain imaging collected using magnetic resonance scanners. It measures matter density through magnetic properties of brain tissue. Since this technique is non-invasive and present no harm to humans, SMRI is a popular technique for brain imaging.

In order obtain an image of a brain, a subject is located in the center of a magnetic resonance scanner, see Figure 1.1. The machine induces a strong magnetic field (from 1 to 10 Teslas) that causes the protons in the subject's head to align their spins. Then, by introducing a pulse of magnetic energy perpendicular to the main magnetic field the spins absorb energy and become excited. The excitatory pulse is a radio frequency pulse tuned at the specific frequency of the hydrogen atom from brain imaging, due to its high concentration as water in the human body. The time it takes the protons to return to their equilibrium magnetization is an exponential



## Chapter 1. Introduction



Figure 1.1: Siemens magnetic resonance scanner. Example of brain scanning for SMRI. Retrieved from <http://www.siemens.co.uk>

decay process with time constant parameters called transverse relaxation time,  $T_2$ , and longitudinal relaxation time,  $T_1$ . The relaxation time  $T_1$  acts as a measure of the material's density. A time sampling interval,  $T_R$ , is designed to be long enough so that gray matter tissue can fully recover in between pulses.

By mapping  $T_1$ , the magnetic resonance machine can report a contrast image, where the intensity of each voxel proportionally relates to the gray or white matter concentration. See an SMRI image example in Figure 1.2.

SMRI is often used by physicians to diagnose brain related diseases or abnormalities like tumors (Young and Knopp, 2006), traumatic brain injuries (Gale et al., 2005), or birth defects (Sowell et al., 2008). More recently, SMRI has also proven useful for the identification of more subtle differences found in patients diagnosed with mental disorders like schizophrenia (Narr et al., 2005; Gupta et al., 2014), bipolar (Fornito et al., 2009), depression (Kipli et al., 2013), attention deficit hyperactivity disorder (Dai et al., 2012), among others.

The raw image obtained from the scanner will vary across subjects depending on the position and size of the head, and scanner parameters. Ideally, a dataset would

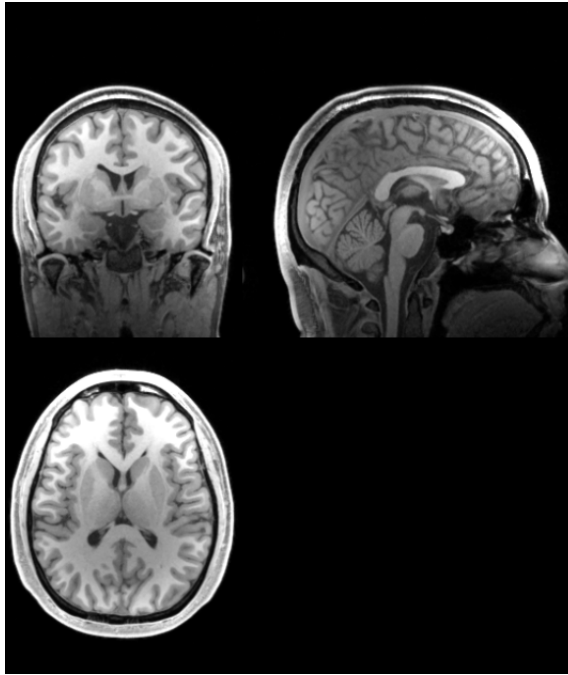


Figure 1.2: Raw SMRI example. This image was taken at the Mind Research Network Institute and belongs to the author of this thesis.

contain images perfectly aligned in size and position, however, this is physically impossible. Thus, several pre-processing steps are required for SMRI dataset analysis. We describe them in the following section.

### 1.1.1 SMRI Preprocessing

Following scanning, the SMRI data is pre-processed to enhance data quality and allow for improved statistical power. The pre-processing steps are performed using SPM 8 software as described in Ins (2012). The basic steps include:

#### **Step 1.** Slice Timing

The slices are collected at different times and require synchronization to avoid signal biases. Slice timing correction is performed using sinc function interpo-

## Chapter 1. Introduction

lation.

### **Step 2.** Realignment

Head motion during scanning, even in the order of millimeters, can still cause significant artifacts. Motion correction is achieved realigning a time series of images acquired from the same subject using a least squares approach and a 6 parameter (rigid body) spatial transformation. Also, the fat chemical that envelops the brain causes a shift in the resulting image and the susceptibility map is not homogeneous due to the air canals near the brain such as the auditory and nasal canals. This correction is performed using an estimate of the susceptibility map and reconstruction from the phase of the acquired image.

### **Step 3.** Spatial Normalization

Spatial normalization involves image registration to the brain atlas to allow for comparisons among different individuals.

### **Step 4.** Smoothing

Spatial smoothing is applied for improving the signal-to-noise ratio (SNR) to allow for better activation detection. Smoothing does not have a high impact on frequency estimation because it is only reducing amplitude and not distorting frequency content.

### **Step 5.** Vectorization and Masking

We vectorize each three-dimensional image. Then we search for voxels outside the brain and remove them from the analysis. This results in a sample by voxel matrix.

### **Step 6.** Covariates

The brain is in constant change through the development of every human being. The literature reveals that age and gender affects gray matter concentration, thus we treat them as covariates and regress them out by keeping the residuals

of a linear regression model which dependent variable is the voxel intensity with gender and age as covariates.

## 1.2 Machine Learning

Machine learning aims to design and implement machines that automate decision making. This field of study combines statistical modeling and computer science to create statistical models that can adapt from observed data in a computationally reasonable manner. A typical machine learning problem is pattern classification, where a statistical model is designed to estimate class labels from observed samples.

In general, classification models define a decision function  $y = f(\mathbf{x}, \mathbf{w})$ , where  $\mathbf{x} \in \mathbb{R}^p$  is a  $p$ -dimensional observed sample,  $\mathbf{w} \in \mathbb{R}^q$  is a vector of  $q$  parameters, and  $f(\cdot)$  is defined in  $\mathbb{R}^p \rightarrow \mathbb{R}$ . The decision function draws a boundary hyperplane in the  $p$ -dimensional space that divides the samples in classes, such that when a new unlabeled sample arrives to the system we can estimate its class by applying  $f(\cdot)$ .

The optimal parameters  $\mathbf{w}$  are learned using a training set of samples. The training set of labeled samples are predicted by the classification model, then the predicted label,  $\hat{y}$ , is compared with the true label,  $y$ , with some loss function  $L(y, \hat{y})$ . The methods seek to minimize such error, so the loss function is minimized by adjusting the parameters,  $\mathbf{w}$ , according to a designed learning rule. In close form we can define the problem as

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{n=1}^N L(y_n, \hat{y}_n), \quad (1.1)$$

where  $N$  is the size of the training set. Then, the optimal decision function is defined by  $f(x, \mathbf{w}^*)$ .

A perfect classification is not always desired, because it may not generalize

for data samples not yet seen. This problem is called over-fitting. Thus, as we described in the introductory section, regularization was introduced to help alleviate this problem. The regularization technique seeks to minimize the parameter norm based on the assumption of a sparse  $\mathbf{w}$ . The updated cost function is  $L'(y, \hat{y}, \mathbf{w}) = L(y, \hat{y}) + \lambda \|\mathbf{w}\|_r$ , where  $\|\cdot\|_r$  denotes the r-norm function, and  $\lambda$  is the emphasis of the regularization on the overall loss function.

### 1.2.1 Deep Neural Networks

Neural networks are machine learning models inspired in the biological function of the nervous system. As in the human brain, the basic unit of processing is called a neuron which acts as a continuous function defined in  $\mathbb{R}^n \rightarrow \mathbb{R}$ ,  $n$  inputs and one output. Several neurons interconnected to each other form a neural network. The connection between neurons are weights that modulate the network as it learns from observed data. See an example of a basic neural network architecture in Figure 1.3.

The first implementations of neural networks can be traced back to the 1950's. The first attempt was in the form of electrical circuits (McCulloch and Pitts, 1943), and later computer implementations (Rosenblatt, 1958) resulted in the oldest neural network still in use today, the Perceptron.

The popularity of neural networks peaked in the 1980's and early 1990's when artificial intelligence took over the media and over-promised in its early results. However, it quickly declined in the early 2000's after the machine learning community revealed many flaws on the early designs of neural networks. This challenged the theoretical foundations of neural networks, yet new advances in training techniques and computational power available made possible the re-discovery of neural networks following a re-branding as deep learning.

Deep learning is a multi-layer neural network that stacks neural networks, i.e.,

Chapter 1. Introduction

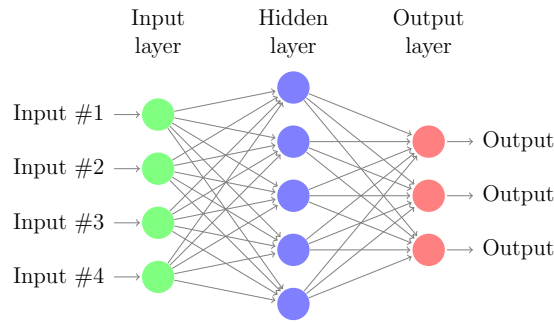


Figure 1.3: Example diagram of a single layer neural network with 3 input features, 4 hidden units, and 3 outputs.

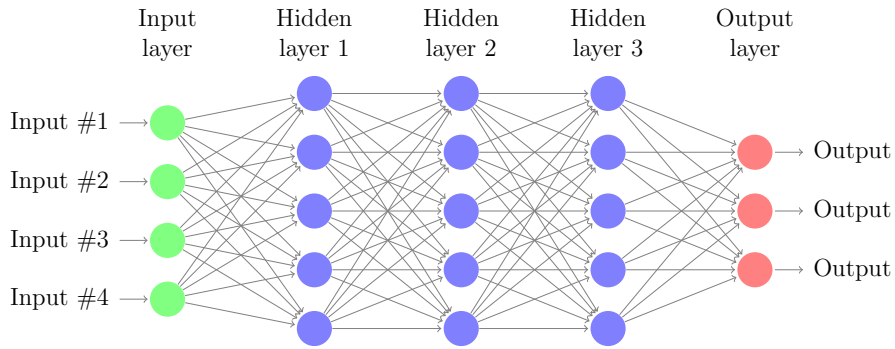


Figure 1.4: Example diagram of a multi-layer neural network with 4 input features, 3 hidden layers of 5 units each, and 3 output.

Perceptrons, to create a bigger and more complex learning model. Such complex neural network structure is most effective on large datasets such as natural images, video, audio, and text (LeCun et al., 2015). See Figure 1.4 for an example of a deep neural network, called Multilayer Perceptron (MLP), with 4 input features, 3 hidden layers of 5 units each, and 3 output.

## 1.3 Motivation for Current Work

Several studies make use of large SMRI datasets to provide evidence of gray matter concentration variations generated by lesions or neuro-degenerative diseases. Detection of tumors (Young and Knopp, 2006), brain lesions (Cuingnet et al., 2011; Gale et al., 2005), and more recently, mental illnesses like attention deficit hyperactivity disorder (ADHD) (Dai et al., 2012), Alzheimer’s disease (Sabuncu et al., 2014), and Schizophrenia (Schnack et al., 2014; Liu et al., 2012). Yet, it often relies on extra pre-processing steps that either reduce the dimensionality at the cost of interpretability (Van Giessen, 2012), or impose prior assumptions with the hope to inflate statistical power (Elsayed et al., 2010).

On the other hand, Deep learning has shown excellent results for the big data scenario, where the number of collected observations is several orders of magnitude larger than the number of variables. For example, crowd-sourced databases of natural images (Krizhevsky et al., 2012), video (Le, 2013), and text (Xue et al., 2008). However, the neuroimaging field poses the opposite scenario. The image of a brain can be composed of around 50,000 voxels (variables) and may only contain between 400 and 2,800 images (Meda et al., 2008; Sabuncu et al., 2014).

The high cost of SMRI data collection usually yields datasets with not enough samples to be applicable in a deep learning setting. Thus, there is a need for methods that help alleviate this problem.

## 1.4 Thesis Statement

The primary thesis of this dissertation is that the generation of synthetic data by our proposed method can lead to improved classification accuracy rates.

## *Chapter 1. Introduction*

First, we provide a comparison framework for the most popular classification methods on raw data. Second, we generate synthetic data, following the proposed method, to replace the raw data where we explore its results with statistical tools.

We hypothesize that feeding a large number of synthetic MRI images to classical machine learning methods may improve the classification accuracy of schizophrenia patients, thus its high potential to be useful in context beyond classification.

### **1.5 Contributions**

In this thesis, we propose a new classification architecture that uses a data-driven sample generation technique to mitigate the effects of a limited sample size. Our approach preserves statistical properties of observed data in the synthetic samples that are continuously generated for training large neural networks.

While the idea of synthetic data generation has previously been used for the recognition of natural images (Netzer et al., 2011; Goodfellow et al., 2013) and in its primitive form (additive noise) as an inherent part of the de-noising autoencoder (Vincent et al., 2010), we are unaware of studies that used synthetic neuroimaging data in an online learning framework.

### **1.6 Thesis Overview**

The thesis is organized in six chapters. The first chapter presented a brief concept review of SMRI data acquisition and machine learning. The second chapter summarizes the classical classification methods used in classification. The third chapter introduces the proposed method and the comparison framework. The fourth chapter describes the case study and the results of applying the proposed method. Chapter



*Chapter 1. Introduction*

five provides a discussion of the results and conclusion. Finally, in the last chapter we suggest future work.

# Chapter 2

## Machine learning

In this chapter, we briefly review some popular classification methods and present the cross-validation technique used for performance evaluation. Finally, we introduce the case study, dataset, and details of the data collection.

### 2.1 Classification methods

Among the vast number of classification methods, we chose the most representative of non-parametric methods such as: nearest neighbors, decision trees, random forests; linear methods such as: logistic regression, linear support vector machines (SVM); non-linear methods such as: multilayer perceptron, polynomial SVM, radial SVM; and the Voting Classifier as an ensemble method.

#### 2.1.1 Nearest neighbors

Nearest neighbors is a simple classification method that bases its decision rule on the class that is mostly represented in the closest labeled samples. If the method

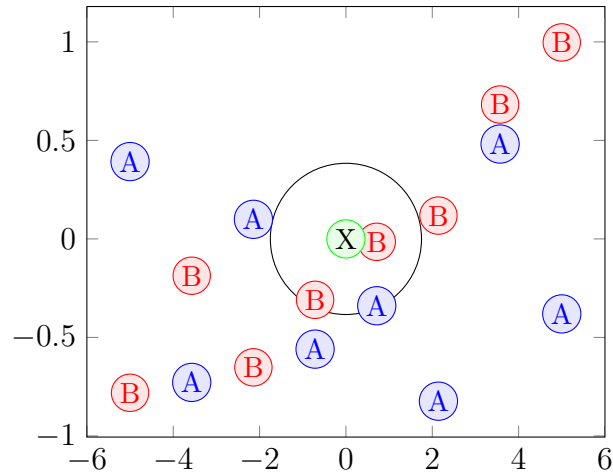


Figure 2.1: Nearest neighbors example for  $k = 3$ . The green sample will be assigned class B because there are 2 samples on class B and 1 sample in class A

chooses to poll all the labeled samples, every new unlabeled sample would be labeled as the class with the more samples on the dataset. Thus, we have to set a determined number of samples to poll from. This is the hyper-parameter  $k$ .

Given a new sample and  $k$  neighbors to search, the method determines the  $k$  nearest points based on Euclidean distance from the new observation and counts the number of elements that belong to each class. Then, it assigns a new point to the class that is most common among the  $k$  neighbors.

As an example of a nearest neighbor classifier with  $k = 3$ , see Figure 2.1. The blue circles represent samples that belong to Class A, and the red circles represent samples that belong to class B. We plot these samples and identify a new sample, marked in green with unknown class X. The black circle around the green sample shows the distance at which the third nearest neighbor is located, so all the samples inside the circle will vote on the class of the green sample. In the figure, we observe two B samples and one A sample, thus we assign the class B to the green sample.

The parameter  $k$  has to be adjusted. A low number of neighbors to use as voters

may lead to overfitting, whereas a high number of neighbors may lead to not fit the data well enough (underfitting).

The simplest implementation is to compute the distance of every point to the query point, then sort by distance and compute the class mode for the  $k$  nearest samples. This simple algorithm is of complexity  $O(nd)$ , where  $n$  is the number of samples and  $d$  is the data dimensionality (number of features). A more recent implementation reduces the complexity to  $O(\log(n))$  (Yianilos, 1993).

The main advantage of this method is that it does not require an assumption on the statistical properties of the data. Moreover, it is very fast to train. However, it can create very complicated and hard to interpret decision boundaries.

### 2.1.2 Decision Tree

Decision Tree is an algorithm that automatically determines how to construct decision rules by constructing a tree that ends in a single class label. One can think of the resulting tree as a recipe. For any new sample, first look at the property in the root node of the tree, if the value for that sample is greater than the value specified in the root, check the property in the node left of the tree otherwise to the right of it, and proceed until a leaf with decisive label is reached.

As an example of a decision tree classifier see Figure 2.2. In this case, we are interested in classifying new coffee samples by its flavor, either good or bad. The tree was constructed after training on various coffee samples from which we know the age, if it is organic or not, and the climate of its precedence. Then when a new sample arrives, we first check the age. If it is more than one year old, then classify it as bad, otherwise keep investigating. The tree not only provides the set of decision rules but by the order of it we can determine its importance. In this example, the age is the most important feature, and the climate the least important feature, because,

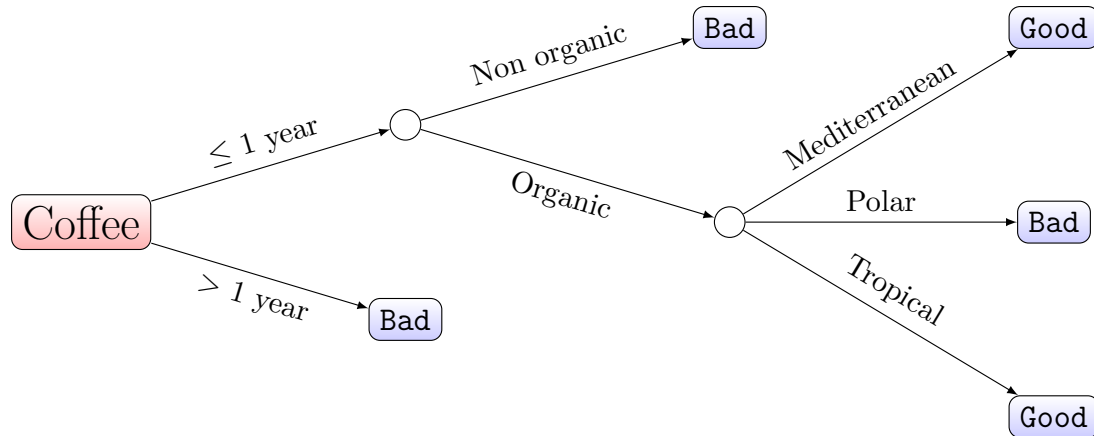


Figure 2.2: Decision tree example for coffee classification. Note that the order of the decisions determines the importance of a feature.

it does not matter if the new coffee is from a tropical region, as long as it is old it will not be good.

The method constructs decision rules that determine the class of a sample by direct numerical comparisons (Breiman et al., 1984). It learns the rules from training data, then it applies the chain of rules to unlabeled data until it reaches the end of the tree to predict its label. In terms of interpretation, the chain of decision rules allows us to directly identify what is the decision mechanism the classifier is following to arrive at any given conclusion.

In theory, there are exponentially many decision trees that can be constructed from a given set of features. Finding the optimal tree is computationally unfeasible because the search space is of exponential size as the number of features increases. Nevertheless, there are several efficient algorithms to train a decision tree, the fastest one claims a complexity of  $O(nT)$  (Kearns and Mansour, 1998), where  $T$  is the depth of the tree. The necessary depth is directly related to the number of features, thus a large number of features will result in a big depth.

A strength of decision trees is that the decision boundaries are arbitrary and can

accommodate both numerical and categorical data. Moreover, the decision rules are also easy to understand and interpret. However, it is highly prone to overfitting as the tree grows in complexity. We can control the maximum depth of the tree to avoid overfitting, yet a low depth will lead to underfitting.

### 2.1.3 Random Forest

Random forest is a classification method that constructs several (thousand) decision trees. Each tree is grown using a bootstrap sample of training data (Breiman, 2001), then all the trees vote on the final class decision. Contrary to the decision tree classifiers, random forests are robust to overfitting at the cost of computation time. Since random forest methods are dependent on decision trees, its computational complexity goes in hand with decision trees.

### 2.1.4 Naive Bayes

The naive Bayes method relies on the Bayes' theorem for the construction of a decision rule:

$$p(y|\mathbf{x}) = \frac{p(y)p(\mathbf{x}|y)}{p(\mathbf{x})}$$

where,  $\mathbf{x}$  is a sample vector, and  $y$  denotes the sample label. Then, when a new sample arrives we ask for the  $p(y|\mathbf{x}_{\text{new}})$  and assign  $\mathbf{x}_{\text{new}}$  to the class with the highest probability.

Taking the Fisher's iris plant dataset as an example, Figure 2.3 depicts the level contours of Gaussian distributions fitted to each class where features are assumed to be independent. This distributions are then used in the decision rule to classify new samples.

From the Bayes' rule, we can infer that  $p(\mathbf{x})$ , evidence, refers to the probability

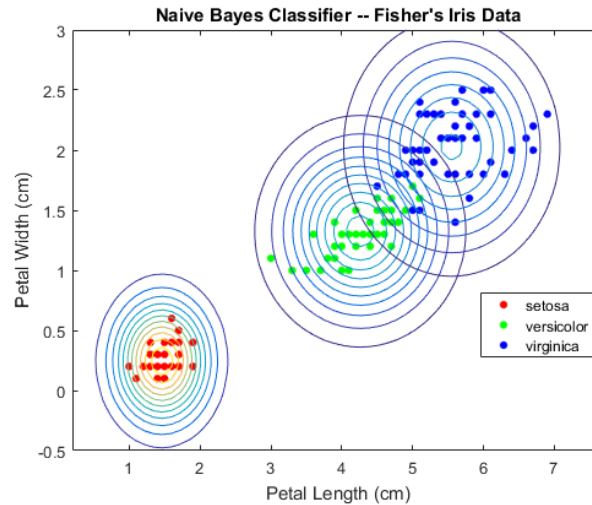


Figure 2.3: Bayes classifier example on Fisher’s iris dataset. Obtained from <http://www.mathworks.com/help/stats/fitnaivebayes.html>

distribution of the population;  $p(y)$ , prior, indicates the population frequency of each label; and  $p(\mathbf{x}|y)$ , likelihood, can be interpreted as the probability density function of samples from each class.

Naive Bayes gets its name because of its strong assumptions. First, it requires a known distribution of samples. Second, it assumes our sample is representative of the population. And finally, it naively assumes independence of features to simplify its decision rule.

The algorithm for naive Bayes is simple and fast to train. The prior is estimated from the frequency of each class in the data, the likelihood is computed from the assumed distribution in close form, and the evidence is drawn from the probability density function of the assumed distribution, where we estimate the parameters from sample statistics. Then, a new sample is assigned to the class with the highest posterior probability for that sample. The training algorithm for Naive Bayes is of complexity  $O(nd)$  (Metsis et al., 2006).

As we stated, the main downside of the method is its strong assumptions. It builds a closed-form likelihood formula, typically assuming a Gaussian or multinomial distribution (Metsis et al., 2006). Moreover, it assumes independence of features which, overall, makes it difficult to apply to any generic dataset.

## 2.1.5 Logistic Regression

Logistic regression, like naive Bayes, is a probabilistic classification method with the goal of estimating  $p(y|\mathbf{x})$ . In particular, logistic regression is designed in a linear regression framework, where it constructs a linear function of  $p(y|\mathbf{x})$  dependent on  $\mathbf{x}$ , where a variation of  $x$  will result in a variation of  $p(y|\mathbf{x})$ . However,  $p(y|\mathbf{x})$  must be between 0 and 1, and linear functions are unbounded. Thus, the logistic function is used to bound the response variable in this regression framework:

$$\log \left( \frac{p(y|\mathbf{x})}{1 - p(y|\mathbf{x})} \right) = \theta_0 + \mathbf{x}\theta.$$

Solving for  $p(y|\mathbf{x})$ , we obtain

$$p(y|\mathbf{x}) = \frac{1}{1 + e^{-\theta_0 + \mathbf{x}\theta}}. \tag{2.1}$$

Logistic regression then uses eq. (2.1) for classification. It creates a hyperplane decision boundary,  $p(y_1|\mathbf{x}) = p(y_2|\mathbf{x})$ , that best divides samples from different classes. In other words, the method estimates the probability  $p(y|\mathbf{x})$ . Contrary to Naive Bayes, Logistic regression is robust on the distribution of the classes in the feature space.

Several implementations of logistic regression have been developed over the years. In principle, the likelihood function for the parameters  $\theta$  is optimized with one of several gradient-based methods (gradient descent, Newton-Raphson, or LBFGS) which can be selected between because some are better suited for different situations dependent on the number of samples and features.



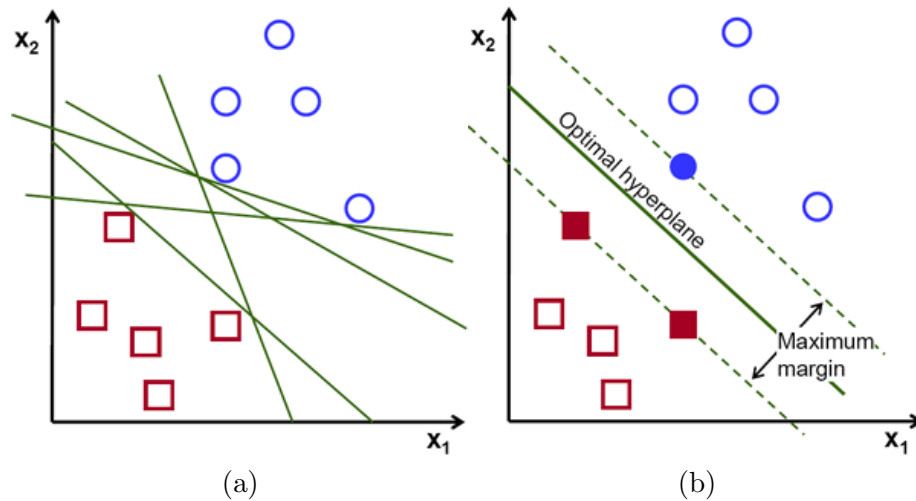


Figure 2.4: Example of a linear decision boundary for a support vector machine. (a) The green lines denote some of the infinity of planes that can divide blue samples from red samples in the feature space of  $\mathbf{x}$ . (b) The green line shows the optimal plane which has the maximum margin. Obtained from <http://docs.opencv.org>

### 2.1.6 Support Vector Machines

As in previously described methods, support vector machines (SVM) seek to define an effective decision boundary to discriminate unlabeled samples. In most scenarios, the decision boundary is not unique and it is up to the method to pick a solution. SVMs search for an optimal solution by maximizing the margin around the separating boundary. See a graphical example of a linear decision boundary in Figure 2.4.

A unique property of SVMs compared to other methods is that not all the points influence equally in the estimation of the decision boundary, only points close to the boundary do. As a result, SVM is robust to outliers. Moreover, it projects the observed samples to a higher dimensional space in order to find separation. The method doesn't allow any points on the wrong side of the boundary.

The main advantage of SVM is its flexibility. The decision boundary for training

data can be designed as either linear or non-linear. Different kernels can produce different boundaries such as quadratic or polynomial. Thus, we could cross validate the best kernel for the training set, and benefit from both linear and non-linear classification.

SVM is based in the optimization of a particular cost function, thus it is simple to incorporate regularization in the form of  $L_1$  or  $L_2$  weight norm, under the assumption of sparse weights.

### 2.1.7 Multilayer Perceptron

A multilayer perceptron (MLP) is a neural network that stacks multiple perceptrons, as explained in section 1.2.1. Each perceptron projects a set of inputs to a multi-dimensional representation of the previous layer. Thus, as we increase the number of layers, we give the network the ability to model increasingly complex structures.

The output projection of the input data is compared to the desired label of training data using a binary cross-entropy cost function defined as

$$e(y, \hat{y}) = - \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)),$$

where  $y$  is the training set labels,  $\hat{y}$  is the classification from the neural network, and  $n$  is the number of samples.

The method then uses a learning algorithm to minimize the binary cross entropy. Most learning algorithms for MLPs use a gradient descent variant method to find an optimal solution. In this thesis, we chose one of the most popular and successful learning algorithm called Adagrad (Zeiler, 2012) due to its convergence speed compared to traditional methods.

The learning algorithm is batch based, i.e., it grabs a batch of samples at a time to optimize the parameters. The batch learning allows us to feed the MLP small sets

of data at a time. The direct benefit of this approach is that it does not need to load all samples into computer memory.

The training of these neural networks grows in complexity with the number of stacked layers, however recent developments improved the design of the starting point and convergence rate for the optimization procedure (Zeiler, 2012).

### 2.1.8 Majority Voting Classifier

The voting classifier relies on all other previously described classifiers. It polls the results of predicting from every other classifier and assigns the new sample to the class with the most votes.

### 2.1.9 Implementation

Several considerations have to be taken into account when fitting and testing classification methods. First, we can not report results on training data, because the model was optimized on this dataset and good results on training data does not necessarily extrapolate to unseen data. Therefore, we have to split the data into training and testing sets.

A popular approach is called  $k$ -fold cross validation, where the samples are divided into  $k$  sets, using  $k - 1$  sets for training and one for testing. The cross validation reports classification scores on the testing set only. Thus, we calculate the average of  $k$  scores for a classifier.

In terms of scoring, simple accuracy is not a fair result to compare, because it does not account for imbalanced datasets and it's only fair when we have same number of samples for each class, which is rarely the case. Instead, we report the area under the

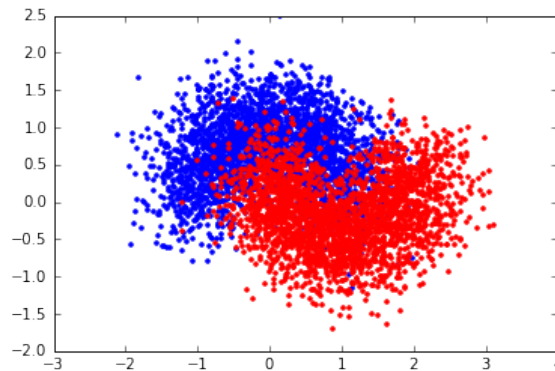


Figure 2.5: Moon dataset for classifier evaluation.

curve (AUC) of the receiver operating characteristic (ROC) curve. The ROC curve is a plot of the true positive rate against the false positive rate. The AUC is then a performance metric for binary classification that is insensitive to sample imbalance.

We also use cross validation for hyperparameter optimization. From the training set, we split in three folds and measure the average score for each combination of parameters. The parameter combination that provides the highest score on the training set is used on the testing set. This allows us to automatically choose a combination of parameters that are not obvious to infer from the data.

As an example, we present results of our implementation (Ulloa et al., 2016), which can be retrieved from <https://github.com/MRN-Code/polyssifier>, in an artificial “moon” dataset with 5000 samples, noise level of 0.4, and two classes, see Figure 2.5. Refer to Pedregosa et al. (2011) for more detail on the moon dataset. We then compute the resulting scores and plot it in order of the testing score in Figure 2.6. Finally, we show the predicted labels across all 10 folds for each classifier in Figure 2.7.

From Figure 2.5, we can observe that the moon dataset is best separated by a non-linear classifier. The results of our implementation confirms that the non-linear

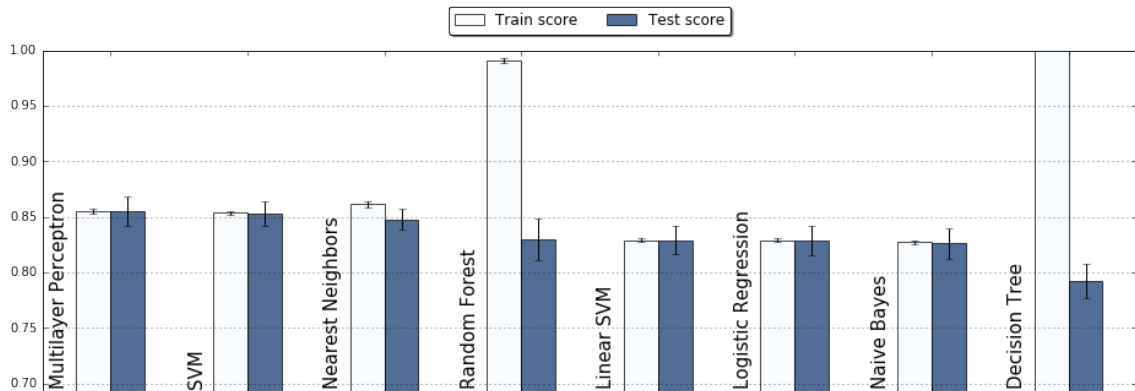


Figure 2.6: Moon dataset results on various classifiers.

classifiers yield the best results. Also, it is possible to observe the decision boundaries from Figure 2.6, where it is clear to observe that decision tree and random forest show differences with more granularity than others, which may have led it to overfit.

Overall, this example supports the reliability of our software showing that it yields results that can be trusted.

## 2.2 Dataset, SMRI

### 2.2.1 Participants

The SMRI data was collected from four sites: Johns Hopkins University, the Maryland Psychiatric Research Center, the Institute of Psychiatry (IOP), London, UK, and the Western Psychiatric Research Institute and Clinic at the University of Pittsburgh, as described in Meda et al. (2008). It contains 198 schizophrenia patients (121 M/ 77 F; age =  $39.68 \pm 12.12$ , range 17-81) and 191 controls (97 M/ 94F; age =  $40.26 \pm 15.02$ , range 16-79).

### **MRI settings**

MRI was obtained on 1.5 T Signa GE scanners with the following parameters (repeat time (TR) = 35 ms, echo time (TE) = 5 ms, flip angle = 45 degrees, 1 excitation, slice thickness = 1.5 mm, field of view = 24 cm, and matrix size =  $256 \times 256$ ), except for IOP data, which was obtained using a 35 degree flip angle and a 20 cm field of view. Patients and controls were collected at all sites. For more information please refer to Meda et al. (2008).

### **Pre-processing**

The T1-weighted images were normalized to Montreal Neurologic Institute (MNI) standard space, interpolated to voxel dimensions of  $1.5 \text{ mm}^3$  and segmented into gray matter, white matter, and cerebro spinal fluid maps. The resulting gray matter images were then smoothed with an isotropic 8 mm full width at half maximum Gaussian filter.

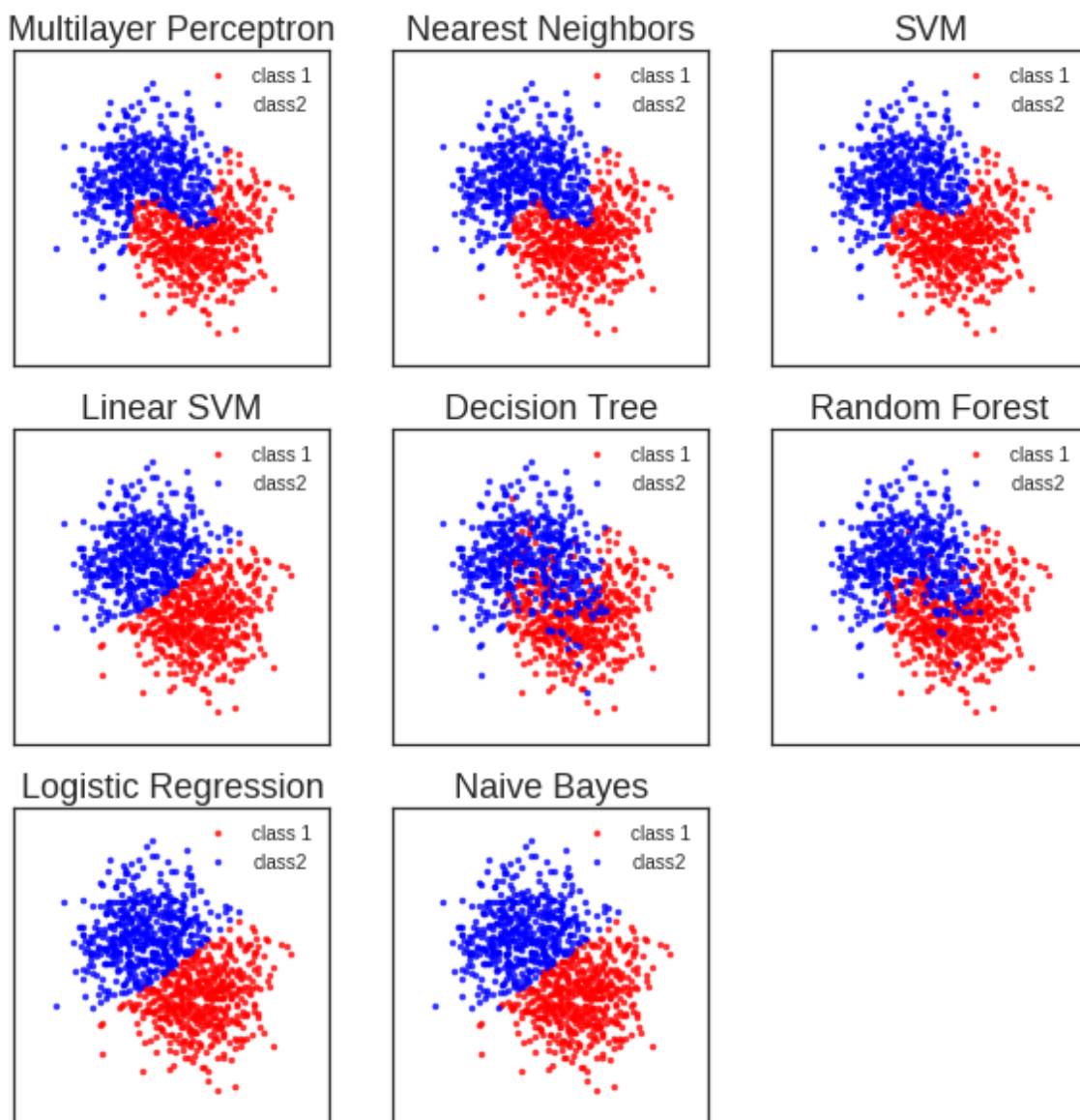


Figure 2.7: Predicted labels across all 10 folds for each classifier

# Chapter 3

## Sample generator model

We propose a sample generator model for improving classification scores on reducible datasets. We define a reducible dataset as a matrix of samples and features that can be reconstructed with small error after being factored by a matrix factorization method such as PCA or ICA. We hypothesize that by augmenting a reducible dataset in its projected space and then reconstructing back to the original space, machine learning should improve the representation of raw data, hence produce better scores.

The proposed method is composed of two main steps, a matrix factorization and a random variable (RV) sampler. We first proceed to introduce both concepts and later describe the proposed method.

The matrix factorization step seeks to decompose a matrix as follows

$$X_{n \times m} = A_{n \times c} S_{c \times m} + \varepsilon, \quad (3.1)$$

where  $X$  is the observed dataset with  $n$  samples (rows) and  $m$  variables (columns),  $A$  is the loading coefficient matrix,  $S$  is the component matrix,  $c$  is the number of components, and  $\varepsilon$  is the error.

Since the application of this thesis is classification, we propose the use of two very



popular matrix factorization techniques, principal component analysis (PCA) and independent component analysis (ICA). Then, we let the machine learning method choose what is the best decomposition method for the data at hand using the classification score in a nested cross-validation framework.

RV samplers are methods for generating RVs from various probability density functions (PDFs). In computers, it is easy to generate uniform distributed RVs and often we rely on the cumulative distribution function (CDF) inverse to transform uniformly distributed numbers to RVs with a particular PDF of interest. However, the inverse CDF is not always possible to derive and we have to rely on iterative sampling methods such rejection sampling or Gibbs sampling, among others.

In the proposed method, the RV sampler function will generate RVs with the same PDF as those in the  $A$  matrix from eq. (3.1). Again, since the final goal in this thesis is maximizing classification score, we implemented two RV samplers and let the method choose the best suited for the data at hand. The first method is a modification of a rejection sampling for multivariate samples, and the second assumes multivariate normality.

## 3.1 Matrix Factorization

As shown in eq. (3.1), we seek to decompose a matrix of data where each row represents an observation and each column represents a variable. The data matrix  $X$  decomposes into loadings  $A$  and components  $S$  given certain constraints. In this thesis, we focus on the two most popular methods PCA and ICA.

### 3.1.1 Principal Component Analysis

PCA was introduced by Hotelling in 1933 (Hotelling, 1933), and it is still widely used for data analysis, matrix factorization, and data reduction. In the context of this thesis, we will use PCA as a matrix factorization with reduction, sample variation summary.

Following eq. (3.1), PCA transforms  $X$  into principal components such that they are uncorrelated. Algebraically, the principal components are linear combinations of the RVs in each column of  $X$ . Geometrically, the components are the result of a coordinate system rotation with each column of  $X$  as the coordinate axes.

The principal components depend only on the covariance matrix of  $X$ . First, the method estimates the sample covariance and computes its spectral decomposition  $E[XX^T] = U\Lambda U^T$ , where  $\Lambda$  is a diagonal matrix containing  $c$  eigen values, and  $U$  contains the eigen-vectors in its columns. Then, the matrix  $S$  is estimated as  $S = \Lambda^{-\frac{1}{2}}U^T X$ , and  $A = U\Lambda^{\frac{1}{2}}$ . Thus, the reconstruction is  $AS = U\Lambda^{\frac{1}{2}}\Lambda^{-\frac{1}{2}}U^T X = UU^T X = X$ .

The first principal component of PCA represents the direction of largest variance in the data, then it searches for the second largest variance direction such that is orthogonal to the first, and so on. Since each component retains a certain amount of variance, it is possible to rank them. PCA is often used to reduce the dimension of the data by retaining the top subset of components that retain a prespecified proportion of total variance.

To illustrate the effect of PCA, we show an example in Figure 3.1. In the example, we set  $A = \begin{bmatrix} \cos(\pi/6) & -\sin(\pi/6) \\ \sin(\pi/6) & \cos(\pi/6) \end{bmatrix}$ , a  $30^\circ$  rotation matrix. Then we sample independent RVs and set them to the rows of  $S$  which represent the true sources. The matrix multiplication of  $A$  and  $S$  results in the observed data  $X$ . We then estimate

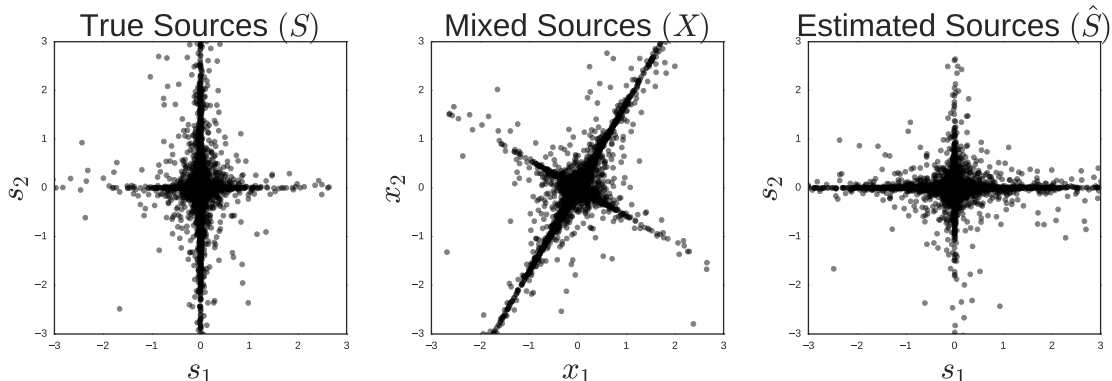


Figure 3.1: Visual example of principal component analysis decomposition.

the sources from the rotated data, which shows the estimated sources rotated back to its orthogonal position. Note that the sign of the first component is flipped, this is accounted in the estimated  $A$  that contains the negative of the original but same in magnitude.

### 3.1.2 Independent Component Analysis

ICA was introduced by Herault et al. in 1983 (Hérault and Ans, 1984) as an extension of PCA. ICA has also been applied to various areas of signal processing including speech separation, communications, and functional magnetic resonance imaging (fMRI) (Hyvärinen et al., 2004; McKeown et al., 1997).

While PCA aims for uncorrelated sources, ICA seeks for independence, often with a function related to the fourth moment. ICA also relaxes the orthogonality required for PCA.

ICA is based on two main assumptions: statistically independent sources, and no more than one Gaussian distributed source. In real world problems these assumptions are reasonable because observed signals are usually composed of weighted sums of several other random signals. Thus, the observed signal tends to be of Gaussian

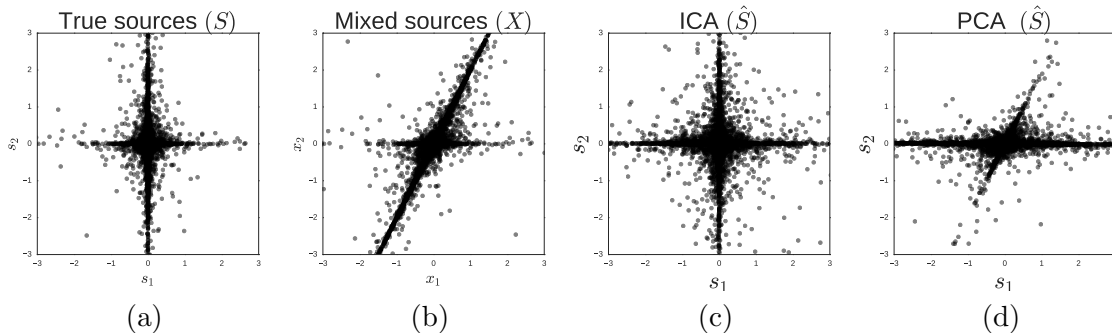


Figure 3.2: Visual example of an independent component analysis decomposition. The plot shows the (a) true sources, (b) mixed sources, (c) estimated sources computed with ICA, and (d) estimated sources computed with PCA

nature (central limit theorem), yet the sources that generates them tend to show high kurtosis. A typical example of this observation is shown sound signals (Bell and Sejnowski, 1996).

We can visualize the decomposition effect of ICA by simulating data in two dimensions, see Figure 3.2. We let  $A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$ , and sample 10,000 independent samples from the logistic distribution for two components of  $S$ . We then multiply  $A \times S$  to generate  $X$ . Note that in contrast to PCA, the matrix  $A$  doesn't have to be an orthogonal matrix.

For this example  $S$  contains independent sources and  $A$  mixes the sources to generate the simulated observed samples. In Figure 3.2 we plot the original independent sources in the left plot, we plot the mixed sources  $X$  in the the middle plot, and we show the estimated sources  $\hat{S}$  in the right plot. Note that PCA would have just rotated the axis in the direction of the component with the highest variance, which is not enough to recover the original sources, see Figure 3.2d.

The literature reveals a various algorithms to estimate independent sources, including infomax, fast ICA, and joint approximate diagonalization of eigenmatrices.

### Chapter 3. Sample generator model

In this thesis, we focus on infomax ICA, which shown the best results for brain imaging data (Correa et al., 2007).

#### Infomax

The infomax algorithm was proposed in Bell and Sejnowski (1995). From the information theory perspective, infomax seeks to estimate sources to minimize the mutual information at zero. In other words, it uses the definition of independence using the joint entropy which is defined as

$$H(\mathbf{x}) = - \int f(\mathbf{x}) \log f(\mathbf{x}) d\mathbf{x}. \quad (3.2)$$

Then, the mutual information is defined as

$$I(x) = -H(g(x)) + E \left[ \sum_i \log \frac{|g_i(x_i)|}{f_i(x_i)} \right], \quad (3.3)$$

where  $g(x)$  depends on the distribution  $f(x)$ . Typically,  $g(x)$  is designed as the sigmoid function

$$g(x) = \frac{1}{1 + \exp^{-x}}. \quad (3.4)$$

Since the expectation term in eq. (3.3) is constant, minimizing the mutual information is equivalent to maximizing the joint entropy. Thus, the infomax algorithm then designs the following optimization problem for maximization of the joint entropy:

$$\widehat{W} = \underset{W}{\operatorname{argmax}} H(g(WX)), \quad (3.5)$$

where  $W$  is the mixing matrix  $W = A^{-1}$ .

Because of the lack of implementation of infomax ICA in python, we ported infomax ICA from the matlab version (Delorme and Makeig, 2004) to python and published it under the GPL license at <https://github.com/alvarouc/ica>.

## 3.2 Random variable samplers

The most simple approach for generating RV samples from a determined PDF is to obtain the inverse CDF in closed form,  $F^{-1}(x)$ , and apply it to transform samples from the uniform distribution. When  $F^{-1}(x)$  is not accessible, we rely on other sampling methods. Moreover, in real data applications, we do not have access to  $f(x)$ , thus we have to either estimate it or assume it to use a RV sampler.

For the purpose of this thesis, we present two methods. The first is a modified version of rejection sampling that does not impose any assumptions on the probability density function of the data but estimates samples assuming variable independence, and the second one assumes a multivariate normal joint density.

### 3.2.1 Rejection sampling

Given a PDF,  $f(x)$ , where  $F^{-1}(x)$  does not exist, we can use rejection sampling to draw samples from  $f(x)$  using an iterative procedure.

First, the method requires the definition of an envelop function  $e(x)$  such that  $e(x) \geq f(x)$ ,  $\forall x \in \mathbb{R}$ . Let  $e(x) = \alpha h(x)$ , where  $h(x)$  is a PDF that is available to sample from, such as the uniform or gaussian PDF, and  $\alpha > 0$  is a scale factor that ensures  $e(x) = \alpha h(x) \geq f(x)$ . Then, the method obtains a sample  $y \sim h(y)$ , and  $u \sim \text{Uniform}(0, e(y))$ , where it accepts  $y$  as a sample from  $f(x)$  if  $u > f(y)$ . This procedure is repeated until the desired number of samples is accepted. The iterative algorithm for rejection sampling is described in Algorithm 1.

---

**Algorithm 1** Rejection sampling algorithm for univariate random variables.

---

```
repeat
  Sample  $y \sim h(y)$ 
  Sample  $u \sim \text{Uniform}(0, e(y))$ 
  if  $u > f(y)$  then
    Reject  $y$ 
  else
    Accept  $y$  as a sample from  $f(x)$ 
  end if
until the desired number of samples is accepted
```

---

### Efficiency

The efficiency of the rejection sampling algorithm depends on the design of  $e(x)$ . The ratio of rejected samples should be minimal to acquire the required number samples in as few iterations as possible. Thus, a poorly designed  $e(x)$  will lead to a large number of rejected samples. Ideally,  $e(x)$  should be tangent to  $f(x)$  or as close as possible.

For example, let  $f(x) = \exp(-\frac{(x-1)^2}{2x})\frac{x+1}{12}$  as in Figure 3.3a, and let  $e(x)$  be defined with  $\alpha = 4.5$  and  $h(x) = \text{Uniform}(0, 15)$ , which we use to produce RVs from  $f(x)$  using the rejection sampling method. To corroborate that the obtained samples are drawn from  $f(x)$ , we plot the normalized histogram of the obtained samples in Figure 3.3b.

In the previous example, the arbitrary  $e(x)$  is not optimal. To illustrate the effect of choosing a better designed  $e(x)$ , we show the ratio of rejected samples by drawing each sample in the  $(y, u)$  coordinate space, where the rejected samples are drawn in blue and the accepted samples in green. The rejection area (blue) is 77.7% of the total area of  $e(x)$ , see Figure 3.3c. Therefore, a better design could be  $e(x)$

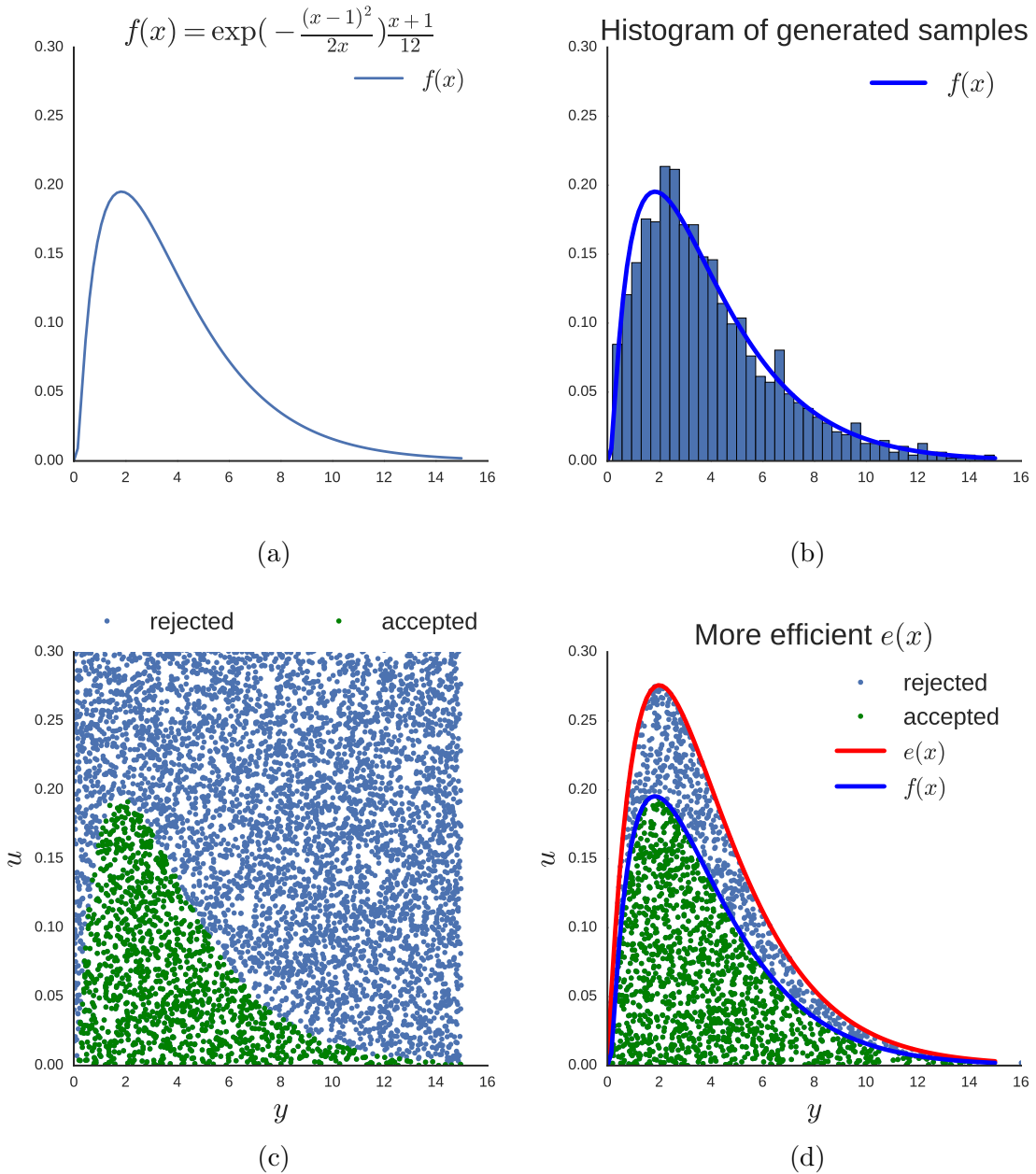


Figure 3.3: Rejection sampling efficiency visualized. (a) Plot of density function to sample from, (b) histogram of accepted samples, (c) plot of rejected samples in blue, and accepted samples in green for  $e(x) = 0.3 \times \text{Uniform}(0, 1)$ , and (d) same plot as in (c) with  $e(x) = 1.5 \times \chi^2(4)$ .



Chapter 3. Sample generator model

with  $\alpha = 1.5$  and  $h(x) = \chi^2(4)$ , where the new rejection area is 33.3% of the total, see 3.3d. Therefore, the second design is more efficient than the first one, yet not optimal.

As we observed, the optimal envelop function  $e(x)$  depends on the area in between  $e(x)$  and  $f(x)$ . As the area reduces, the efficiency of rejection sampling increases. Therefore, we propose that the optimal  $e(x)$  can be found by solving the following optimization problem:

$$\hat{\theta}, \hat{\alpha} = \underset{\theta, \alpha}{\operatorname{argmin}} \int (\alpha h(x|\theta) - f(x)) dx, \text{ s.t. } e(x) - f(x) \geq 0, \forall x \in \mathbb{R}, \quad (3.6)$$

which, using the fact that  $h(\cdot)$  and  $f(\cdot)$  are PDFs, reduces to

$$\hat{\theta}, \hat{\alpha} = \underset{\theta, \alpha}{\operatorname{argmin}} \alpha, \quad \text{s.t. } \alpha h(x|\theta) - f(x) \geq 0, \forall x \in \operatorname{Domain}\{f\} \quad (3.7)$$

See an example of an analytical solution for  $\alpha$  and  $\theta$  in the case were we require samples from Beta(2, 2) using  $h(x|\theta) = \operatorname{Uniform}(0, \theta)$  in Appendix A.

### Multivariate RV extension

Let the joint probability density function of  $A$  be  $f_A(\mathbf{x})$ , and the marginal densities be  $f_A(x_i)$  for  $i = 1, 2, \dots, c$ , where  $c$  is the number of components in  $A$ . Then, assuming the marginal random variables are independent, we can obtain the joint distribution by  $f_A(\mathbf{x}) = \prod_{i=1}^c f_A(x_i)$ . In other words, we assume  $x_i$  are independent and apply rejection sampling to each  $x_i$  to generate  $\mathbf{x}$ .

### 3.2.2 Multivariate Normal

We use the sample mean and sample covariance matrix from  $A$  as input to this generator. Then, we use the spectral decomposition approach for generating multivariate

random normal samples. Contrary to the rejection sampling generator, this approach accounts for the correlation structure among the RVs, but it loses the generality of the marginal distributions.

### 3.3 Data driven sample generator

The proposed method is designed with the goal to provide machine learning models an augmented dataset that is as close as possible to real data. Machine learning methods then could take advantage of the extra sample variability to build more robust decision boundaries and avoid overfitting. In particular, we focus on datasets that are rich in features but short of samples, which is the scenario where machine learning models tend to overfit and fail.

Our proposed method for a data driven sample generator builds from two assumptions:

- The input dataset is reducible, as in error from matrix factorization reconstruction is minimal.
- A group of samples with a common diagnosis (class) share statistical properties that are reflected in their loading coefficients ( $A$ ).

Based on these assumptions, the proposed method proceeds as follows. First, it factors the observed dataset  $X$  into  $A$  and  $S$  (e.g., using PCA or ICA). Next, it splits  $A$  into sub-matrices,  $A = [A_1^T, A_2^T, \dots, A_C^T]^T$ , where each  $A_i$  represents a class of samples.

After that it feeds each  $A_i$  matrix to a RV generator of choice. In case of the rejection sampling method, we first estimate the PDF of each column of  $A_i$  and generate new samples with that distribution. Otherwise, we compute the sample

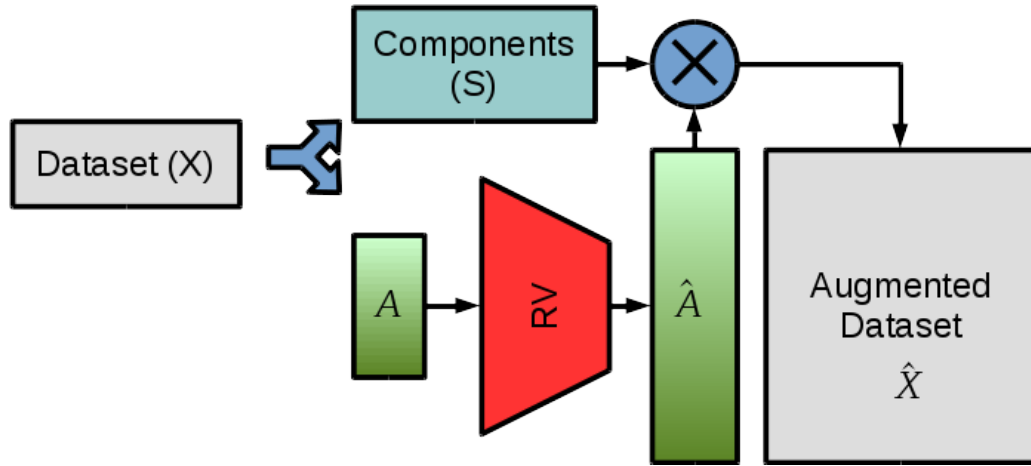


Figure 3.4: Data driven generator block diagram. The dataset is factorized into components and loading matrix  $A$ . The RV block denotes the RV generator that fits from  $A$  and generates new samples to reconstruct the synthetic dataset.

mean and covariance to generate new samples with the same parameters. The method then reconstructs a new dataset using  $\hat{X} = \hat{A}S$ . As we stated before, the matrix factorization method is left to the classification score to decide, as well as the RV sampler of choice.

To use the rejection sampling method, we estimate the joint PDF of each  $A_i$  using a normalized histogram with  $M$  bins of each column and a smoothing kernel, which we denote by the function  $\text{pdf}_M\{\cdot\}$ . Based on data observations, we set  $M = 20$ . We use the rejection sampling method to sample the marginals from the joint PDF,  $f_i = \prod_{j=1}^c \text{pdf}_N\{[A_i]_j\}$ ,  $\forall i \in \{0, 1, \dots, N\}$ , where  $j$  indicates the  $j^{\text{th}}$  marginal PDF. In the case of the multivariate normal RVs, we simply use the maximum likelihood estimators (MLE) to estimate the mean and covariance matrix of  $A_i$  and generate  $M$  samples using the estimated parameters,  $\hat{A}_i \sim \text{MVN}(\bar{A}_i, \Sigma_i)$ .

The method RV generator method is depicted in Figure 3.4.

# Chapter 4

## Application to SMRI classification

The number of SMRI images that can be collected per study is limited by the high collection cost and patient availability. It requires expensive facilities, qualified technicians, and significant patient time. In Chapter 3 of this thesis, we proposed a data driven generator model to mitigate the negative effects of a limited sample size by artificially augmenting the dataset. As a case study, we use SMRI images to classify people into two diagnostic groups: patients with schizophrenia and healthy controls.

The dataset was presented in section 2.2. We now elaborate on the process of applying the proposed method in a classification problem. We will apply the traditional methods that were described in Chapter 2 to SMRI data for baseline results and to the augmented dataset to measure any significant improvement.

The present chapter follows with a linear statistical analysis of the SMRI images related to patient demographics, then classification results using both raw and augmented datasets.

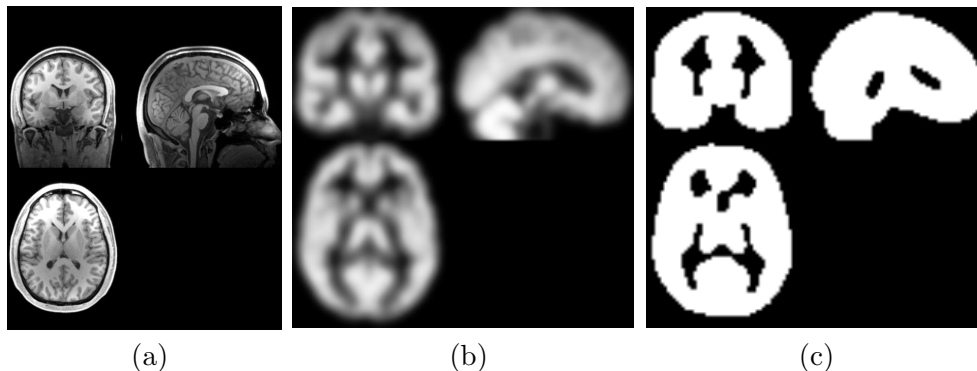


Figure 4.1: (a) Image taken from the author of this thesis as an example of a raw SMRI image. (b) Image after pre-processing steps described in Section 2.2.1. (c) Mask used to keep intracranial voxels.

## 4.1 Data analysis

### 4.1.1 Data set

The dataset consists of 389 subjects, roughly age and gender balanced (schizophrenia is 40% more prevalent in males), as described in Table 4.1. Each subject was scanned to obtain a three dimensional image of 52 by 63 by 45 voxels, where each voxel represents gray matter concentration (GMC) (from 0 to 1) in a 3mm by 3mm by 3mm cube, e.g., see Figure 4.1a. After the preprocessing steps described in 2.2.1, all the images are aligned and look as we show in Fig. 4.1b. We then mask out voxels outside of the brain, see Figure 4.1c, and vectorize the image. This results in a vector of 60,465 voxels per sample.

### 4.1.2 Linear model

Since age and gender have been shown to be highly correlated with gray matter concentration (Takahashi et al., 2011), we include them in the analysis as factors.

Table 4.1: Participants demographics distribution.

	Patient	Control	Total
Male	121	97	218
Female	77	94	171
Age	39.68±12.12	40.26±15.02	
Total	198	191	389

Table 4.2: Participants demographics distribution for three factors: age, gender, and diagnosis.

Age	Healthy		Patient		Total
	Male	Female	Male	Female	
Young (16-33)	39	35	37	19	130
Adult (34-43)	27	25	51	25	128
Senior (44-81)	31	34	33	33	131
Total	97	94	121	77	389

We partition age into three groups: young (16-33), adult (34-43), senior (44-81). The demographics of the participants passing quality control, see Section 2.2.1, are summarized in the three-way table in Table 4.2.

Now, we conduct a three-way ANOVA on GMC as the response variable, diagnosis, age, and gender as factors, and its interactions. For every voxel, we fit the full model including the three-way interaction. Then we perform backward selection in the standard way. We first check whether the three-way interaction (age-gender-diagnosis) is significant at the 0.01 level; if it is not significant, we remove that term and repeat testing until all model terms are either significant or included in a higher-order significant term.

### 4.1.3 Results

We compute the p-value for every voxel and we transform it with  $-\log_{10}(-p)$  to improve the graphical representation overlaid on the MNI structural template. For the main effects, we correct the resulting p-values for multiple comparisons using the false discovery rate (FDR) with a 0.01 level. In Figure 4.2 we show the FDR-corrected transformed p-value where the effects of each factor are significant.

The main factor of interest in this study is the diagnosis. The brain regions where the diagnosis showed a significant effect are at the left and right superior temporal gyrus, and the superior frontal gyrus. The schizophrenia patients show an average reduction of GMC of 0.043 at the right superior temporal gyrus, 0.048 at the left superior temporal gyrus, and 0.036 at the superior frontal gyrus. No other factor, or interaction showed significant effects on these brain regions.

We also extract brain regions where age showed a significant effect. These are the left and right thalamus, and parahippocampal gyrus. Overall, it shows an increasing GMC trend as age progresses, however in the parahippocampal gyrus, adults and seniors show no significant difference. Again, we observed no interaction or effects of other factors in these regions.

Age also showed interaction effect with diagnosis on the left and right parietal lobule, where young patients showed the largest reduction in GMC compared to controls (0.073 on right and 0.065 on left).

Gender showed no significant effect on any brain region, however, it showed significant interaction with diagnosis on the right fusiform gyrus, where the largest difference is found between male patients and male controls (0.02 average difference).

The three way interaction only had a significant effect on a small region of the left precuneus, where senior female patients showed the largest GMC reduction compared

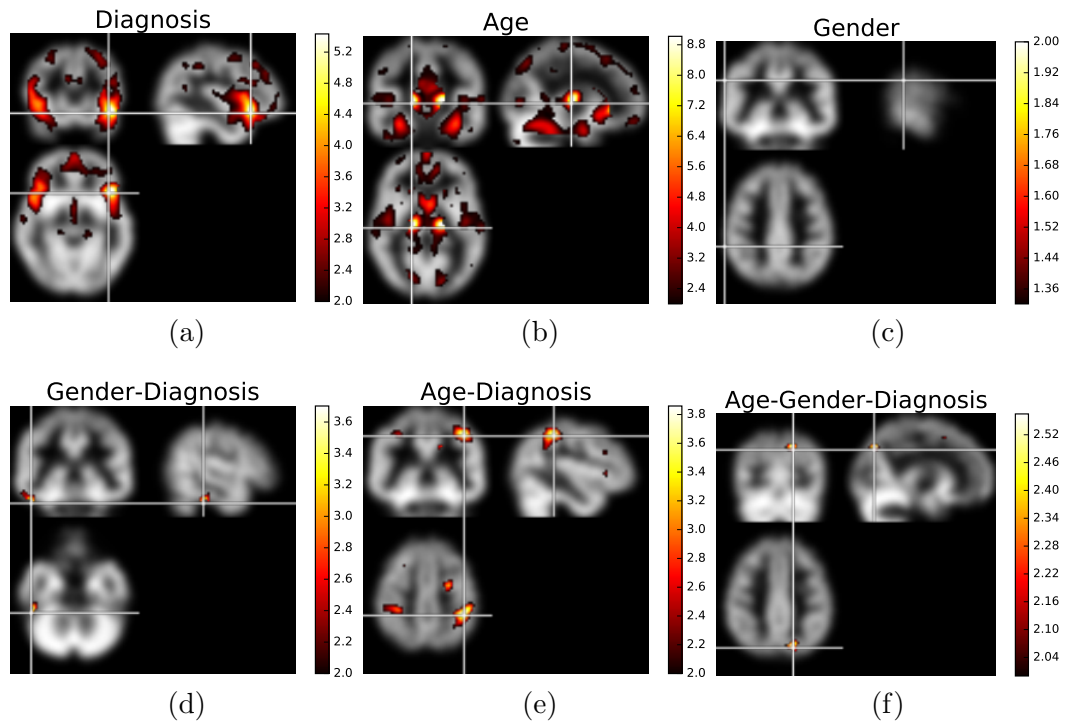


Figure 4.2: Three-way ANOVA results. Voxels passing fdr correction for multiple comparison at the 0.01 level for the (a) diagnosis, (b) age, and (c) gender effects. Significant interactions between (d) gender and diagnosis; (e) age and diagnosis; and (f) age, gender, and diagnosis. We show the significance as  $-\log_{10}(p)$ .

to all others (0.041).

We plot this results by factor in Figure 4.2, and all factors at one in Figure 4.3. We also summarize the effects and group means in Table 4.3 and Table 4.4.

## 4.2 Classification results

We now present classification scores for each of the classifiers discussed in Section 2.1. We compute scores for raw data, ICA projected data, PCA projected data, and the augmented dataset generated with the proposed method in this thesis.



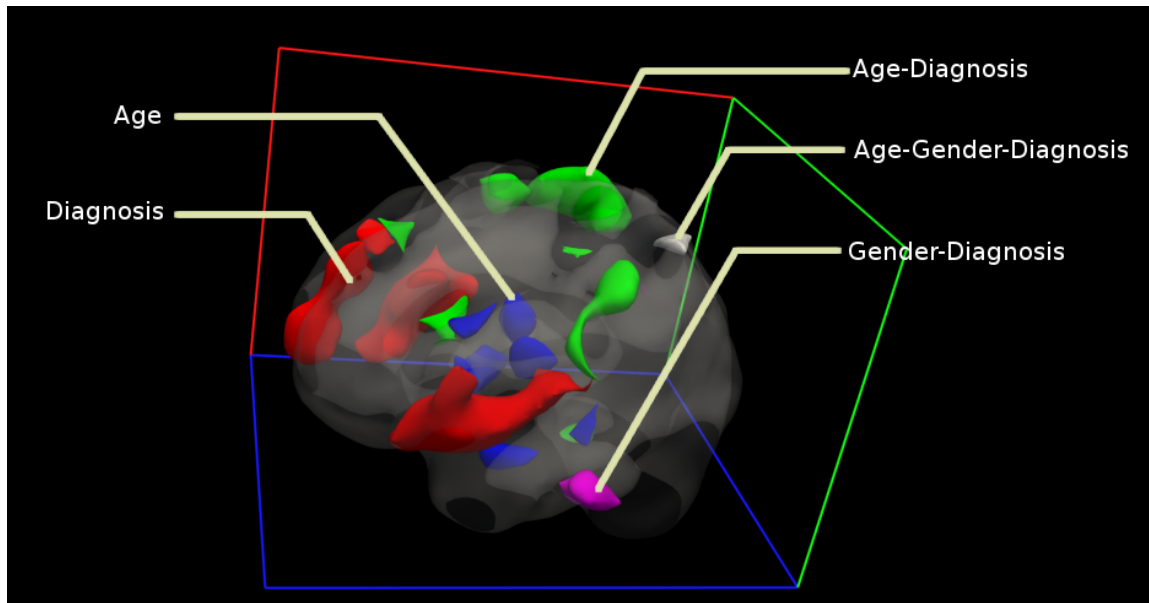


Figure 4.3: Three dimensional view of all effects on the GMC mean. Red marks the diagnosis effect after FDR correction at 0.01 level. Blue shows the age effect after FDR correction at 0.01 level. Green shows the age-diagnosis effect at 0.01 level. Pink shows the gender-diagnosis effect at 0.01 level. White shows the three-way interaction effect at 0.01 level.

Table 4.3: Three way ANOVA group means for the main effects of schizophrenia dataset.

Effect	Brain Region	Group Means ( $\times 10^{-2}$ )		
		Control	Patient	
Diagnosis	Right Superior Temporal Gyrus	57.7	52.9	
	Left Superior Temporal Gyrus	60.2	55.4	
	Superior Frontal Gyrus	39.3	35.7	
Age	Left Thalamus	30.8	33.7	35.9
	Right Thalamus	34.1	37.5	39.8
	Right Parahippocampal Gyrus	68.3	73.7*	73.3*
	Left Parahippocampal Gyrus	66.8	71.3*	71.7*
Gender	None			

\* Not statistically different.

Chapter 4. Application to SMRI classification

Table 4.4: Three way ANOVA group means for the effects of interactions on schizophrenia dataset.

Effect	Brain Region	Group Means ( $\times 10^{-2}$ )			
			Control	Patient	
Gender-Diagnosis	Right Fusiform Gyrus	Male	27.7 <sup>(a)</sup>	29.7 <sup>(b)</sup>	
		Female	29.4 <sup>(a,b)</sup>	28.1 <sup>(a,b)</sup>	
Age-Diagnosis	Right Inferior Parietal Lobule	Control	Young 54.5 <sup>(c)</sup>	Adult 53.0 <sup>(b,c)</sup>	Senior 47.1 <sup>(a)</sup>
		Patient	47.2 <sup>(a)</sup>	50.7 <sup>(a,b)</sup>	48.8 <sup>(a)</sup>
	Left Inferior Parietal lobule	Control	Young 51.9 <sup>(b,c)</sup>	Adult 52.2 <sup>(c)</sup>	Senior 47.1 <sup>(a)</sup>
		Patient	45.4 <sup>(a)</sup>	50.2 <sup>(a,b)</sup>	48.3 <sup>(a,b)</sup>
Age-Gender-Diagnosis	Left Precuneus	Senior Female Patient	43.0	Others 47.1	

We set the parameters for each classification method shown in Table 4.5. The methods that show a list of values for a parameter were trained using a grid search approach, where the best scoring combination of parameters in a subset of the training data was used to predict on the test data. In the case of the proposed methodology, we appended two additional hyper-parameters, the decomposition method (PCA or ICA) and the R.V. generator (rejection or MVN).

Not all classifiers are scale invariant, so we normalize the data by removing the mean and scaling each voxel to unit variance over the subjects. Then, we split the data in 5 folds, where 4 are used for training and the remaining for testing. Each fold is used for testing once. We fit each classifier on the training set, on the projection of the training set, or on the augmented dataset created from the training set using the proposed method. Finally, we report the score and standard deviation from the testing folds and summarize it on Table 4.6 and Figure 4.4.

Overall, the MLP method showed the best performance when the proposed methodology was used for training. Among the type of classifiers, the linear methods showed the best average scores, followed by the non-linear classifiers, and the non-

Chapter 4. Application to SMRI classification

Method	Parameter	Values
Nearest Neighbors	Number of neighbors	[1, 5, 10, 20]
Decision Tree	Maximum number of features	'auto'
Random Forest	Number of estimators	[5...20]
Naive Bayes	Kernel	Gaussian
Logistic Regression	C	[0.001, 0.1, 1]
Support Vector Machines	Kernel	[radial, polynomial]
	C	[0.01, 0.1, 1]
Linear SVM	C	[0.01, 0.1, 1]
	Penalty	['L1', 'L2']
Multilayer Perceptron	Depth	[3, 4, 5]
	Number of hidden units	[50, 100, 200]

Table 4.5: Classification methods and parameters for grid search

parametric classifiers. The decision tree method showed no average improvement but reduced its standard deviation when using the generator. Linear SVM along with logistic regression reported no difference on the average score for raw vs augmented dataset, however when using data projections they report decreased scores. Naive bayes along with random forest performed the best when the dataset was projected using PCA but generally not performing well.

Method	Raw	ICA	PCA	Augmented
Logistic Regression	72.1 ± 3.5	66.4 ± 7.6	67.5 ± 3.9	71.0 ± 3.0
Multilayer Perceptron	60.2 ± 12.5	67.9 ± 5.2	66.6 ± 3.7	75.0 ± 4.5
SVM (radial, poly)	70.5 ± 5.9	57.0 ± 4.7	64.0 ± 5.5	70.1 ± 4.0
Linear SVM	69.1 ± 6.7	68.2 ± 7.5	67.4 ± 4.3	71.3 ± 3.9
Naive Bayes	60.3 ± 6.0	59.8 ± 8.6	65.2 ± 5.8	58.3 ± 3.7
Decision Tree	55.5 ± 4.9	54.3 ± 5.1	56.0 ± 5.6	55.2 ± 3.3
Random Forest	60.1 ± 3.4	62.3 ± 5.7	65.6 ± 3.9	63.3 ± 2.3
Nearest Neighbors	62.7 ± 3.5	58.6 ± 6.2	65.1 ± 3.8	60.3 ± 3.5

Table 4.6: Classification results on raw data, ICA reduced, PCA reduced, and augmented dataset.

## Chapter 4. Application to SMRI classification

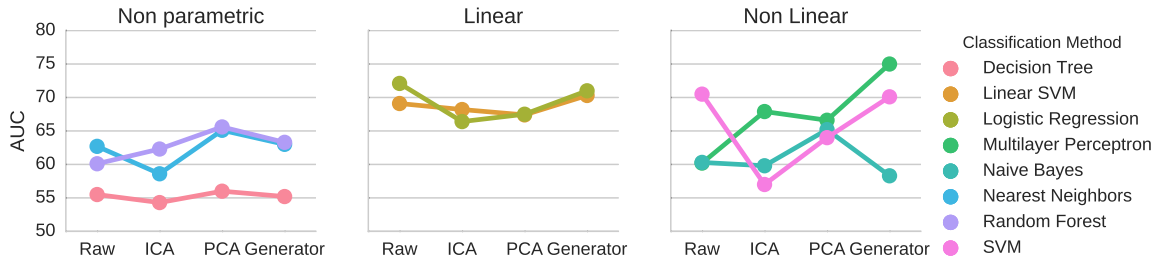


Figure 4.4: Average classification score results for raw, ICA reduced, PCA reduced, and augmented data grouped by type of classifier.

### 4.3 Size effect for data generator

We tested the size effect for the proposed methodology. The experiment consists on varying the number of synthetic samples as 10, 100, 1000, and 5000 samples per group, and measuring the average and standard deviation of the scores across folds for each classifier at each size level. We did not make use of the grid search approach for this experiment for computation and time constrains. This should not affect the overall trend but may affect the absolute values when compared to the results obtained in Table 4.6.

We plot the average score of each classification method and the spread of the scores measured by the standard deviation for the training and testing sets. For the training scores, we trained on synthetic data and report the score on the training set from which the synthetic data was generated. The training average score increases with the size and saturates at smaller numbers of generated samples for naive Bayes, logistic regression and linear SVM. The standard deviation of training scores decrease with the data size. For the test score we also observe an increasing trend, however the variability is higher. The results are summarized in Figure 4.5.

Chapter 4. Application to SMRI classification

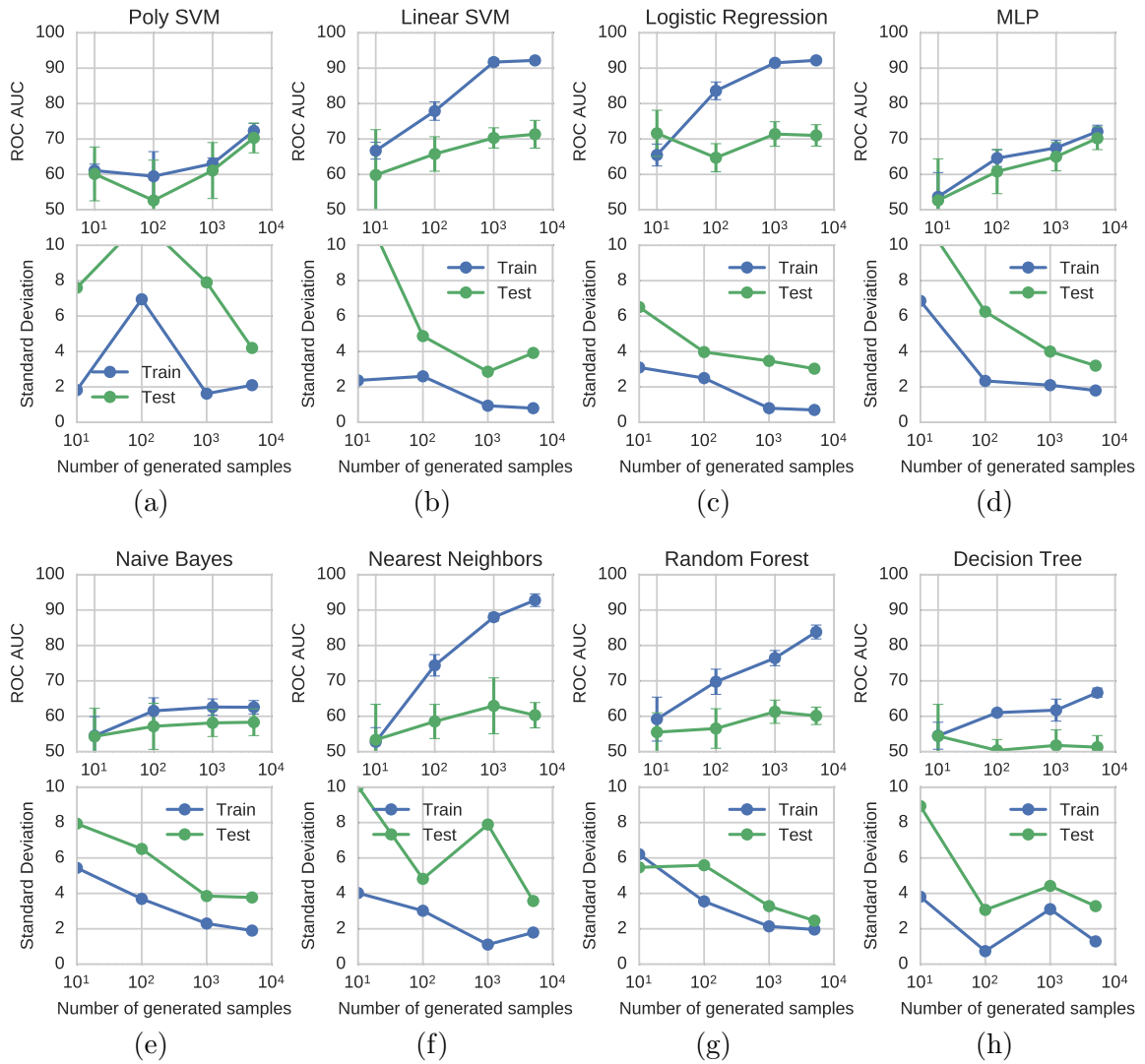


Figure 4.5: Size effect for various classification methods trained on synthetic data.

# Chapter 5

## Discussion

We initially hypothesized that training on a large number of synthetic but realistic samples may improve the accuracy of classifying schizophrenia patients versus healthy controls. The results show evidence in favor of our hypothesis because when using synthetic data, the classification scores matched or improved when compared to other approaches, see Figure 4.4. Results are encouraging and provide evidence of the value of the proposed synthetic data generator for classification.

We conducted an exploratory data analysis to investigate the differences between healthy controls and schizophrenia patients with an ANOVA model. The model consists of the voxel intensity as the response variable, diagnosis as the main factor, and age and gender as covariates. After conducting the analysis with model reduction, the results suggested a significant influence of the superior temporal and frontal gyrus on the schizophrenia diagnosis, where patients exhibited a significant decrease in GMC. These brain regions have been reported on several other publications concerning schizophrenia and GMC (Kasai et al., 2003; Rajarethinam et al., 2000; Gupta et al., 2015). Thus, our dataset replicated past findings, which builds confidence in its validity.

## Chapter 5. Discussion

Secondary results of the exploratory analysis suggested that age, gender and interaction effects are significant on other non-overlapping brain regions. As the age progresses, all the subjects showed an increased GMC on the thalamus and the parahippocampal Gyrus, which matches findings in (Rzezak et al., 2015; Fama and Sullivan, 2015). The gender showed no significant effects on GMC when controlling for diagnosis and age, yet it suggested significant interaction effects with the diagnosis, where males showed the highest increase of GMC at the right fusiform gyrus. A recent study in (Koenders et al., 2015) reports the opposite of our findings, where male patients exhibited decreased GMC, thus we suggest caution and further replication should be studied. The age factor also showed a significant effect on the inferior parietal lobules, where only young subjects hold a meaningful reduction of GMC. A study focused in the inferior parietal lobules (Torrey, 2007) reveals a lack of consistency on the literature, where 6 studies reported decreased GMC on males only and 3 reported a significant increase in GMC. We suggest replication of this experiment to increase confidence in this last results. Finally, the analysis also revealed significant reduction of GMC for senior female patients when compared to all others at a very small section of the left precuneus. The literature does not show a clear finding related to schizophrenia in this brain region.

In summary, the results from the ANOVA analysis reveals that there is some separability between healthy controls and schizophrenia and linear classification methods should be able to find and exploit those regions.

Out of the learning methods used in our experiments, the MLP stands out in its classification performance when using synthetic data. This suggests the utility of deep learning in the area of neuroimaging. Deep neural networks have been gaining popularity in areas where data are abundant, so called “big data”, however its utility on the other side of the data size spectrum, scarce data, is yet to be defined. Our MLP with synthetic data generation brings deep neural networks a step closer to

applications to neuroimaging data.

A scenario where overfitting is not a problem is when the number of collected observations is much larger than the number of variables. Deep learning methods have proven most effective for big data problems, such as natural images (Krizhevsky et al., 2012), video (Le, 2013), and text (Xue et al., 2008) processing. However, in the medical imaging field we find the opposite scenario. For example, an image of the brain taken from structural magnetic resonance imaging (SMRI) can be composed of around 50,000 voxels (variables) and even a large dataset may only contain between 400 and 2,800 images (Meda et al., 2008; Sabuncu et al., 2014).

The literature shows many efforts on overcoming the effect of overfitting. For example, in classical regression and classification problems it is a common practice to add constrains to cost functions in the form of  $L_1$  and  $L_2$  norms of the model parameters (Schmidt et al., 2007; Tibshirani, 1996), or a combination of the two as elastic net regularization (Zou and Hastie, 2005). More specifically, deep learning methods also use other regularization techniques in combination to the latter ones, such as dropout (Dahl et al., 2013; Srivastava et al., 2014), and additive noise as an inherent part of the de-noising autoencoder (Vincent et al., 2010).

While the idea of synthetic image generation has previously been used for the recognition of natural images (Netzer et al., 2011; Goodfellow et al., 2013) and in its primitive form (additive noise) is an inherent part of the de-noising autoencoder (Vincent et al., 2010), we are unaware of studies that used synthetic neuroimaging, genetic, or combination of data modalities in a learning framework as presented in this thesis.

The proposed data generator method exploits the fact that SMRI data is spatially redundant (smooth) and, with insignificant loss of information, effectively reduces dimensionality using ICA. Several studies on gray matter concentration favor the use



of ICA (Smith et al., 2004; Liu et al., 2012) because of the easy interpretation of its results and its compatibility with known regions of the brain.

The reduced data is then passed to a data-driven R.V. generator. The R.V. generator takes the reduced data, mixing matrix, and emulates its statistical properties with two different approaches. The first approach is a method based on rejection sampling that, from the sampled PDF, simulates R.V.s column by column of the mixing matrix. This approach provides flexibility for modeling arbitrary PDFs from the sampled mixing matrix, however it ignores interactions among columns of the reduced data. On the other hand, the second approach, a simple multivariate normal RV generator, captures the correlations between columns but fixes the joint PDF of the variables.

In concordance with Li et al. (2007), we observed that as we increase the number of estimated sources, it is more likely to encounter correlation among columns of  $A$ . It often occurs that over-estimation of the number of sources results in spatial splits, which then show a similar pattern at the mixing matrix level. Thus, the use of a multivariate normal R.V. may be advantageous. On the other hand, if we set a lower number of sources the loading coefficients are less likely to be correlated, so the use of a rejection sampling R.V. generator should be preferred. Overall, it is a good practice to observe the level of correlation on the mixing matrix and pick a method that better fits the correlation structure.

To the best of our knowledge, the proposed methodology is the first attempt to classify neuroimaging data in an online fashion using purely synthetic data. Results, showing the proposed method in synergy with MLP had the highest average classification scores, are encouraging and provide positive evidence of a promising methodology. Moreover, the proposed application can be used for sharing data and let researchers use it to train their models searching for better algorithms to classify schizophrenia. This is especially true for datasets that cannot be shared in raw

## *Chapter 5. Discussion*

format for ethical or legal reasons.

Finally, it is important to mention that information from the model learned by the classifiers can be extracted to identify brain regions that are of importance for the classification. The identified brain region may be then used for targeting patient treatment or further research.

# Chapter 6

## Future Work

Despite the demonstrated utility of our approach, there are several open questions for the future work. For example, when does the simulator become useful as a function of the size of the data? That is, when is there enough information in the data to help generalization? Similarly, at what data size is there diminishing returns so that the benefits of the synthetic data generator levels off? Additionally, even the MVN simulator only captures second order statistics of the distribution, it remains unknown if distributions that model more complex interactions are of further utility. However, we hypothesize that this indeed may be the case according to the trend that we already observed going from univariate to multivariate PDF.

# Appendix

# Appendix A

## Example of analytic solution for optimal envelop function

Let  $f(x) = \text{Beta}(2, 2) = 6x(1 - x)$ ,  $x \in [0, 1]$ , see Figure A.1, and

$$h(x|\theta) = \text{Uniform}(0, \theta) = \begin{cases} 1/\theta, & \text{if } 0 \leq x \leq \theta \\ 0, & \text{otherwise} \end{cases}$$

From eq. (3.7), we solve the following

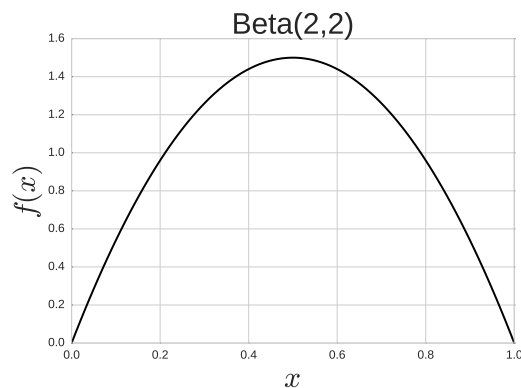


Figure A.1: Beta(2,2) probability density function.

Appendix A. Example of analytic solution for optimal envelop function

$$\hat{\theta}, \hat{\alpha} = \underset{\theta, \alpha}{\operatorname{argmin}} \int_0^1 (\alpha h(x|\theta) - f(x)) dx, \text{ s.t. } e(x) - f(x) \geq 0, \forall x \in [0, 1]$$

or

$$\hat{\theta}, \hat{\alpha} = \underset{\theta, \alpha}{\operatorname{argmin}} \alpha, \text{ s.t. } \alpha h(x|\theta) - 6x(1-x) \geq 0, \forall x \in [0, 1]$$

For  $\theta < 1$ ,  $h(x|\theta) = 0$  for  $x \in [\theta, 1]$ , then the constrain reduces to

$$0 - 6x(1-x) \geq 0,$$

$$x(1-x) \leq 0$$

where, a solution is  $x > 0$  and  $1-x < 0$ , which means  $x > 1$ ; and the other solution is  $x < 0$  and  $1-x > 0$  which means  $x < 0$ . This is a contradiction, thus,  $\theta \geq 1$  for the constrain to hold.

Now, for  $\theta \geq 1$ , the constraint is  $\alpha \frac{1}{\theta} - 6x(1-x) \geq 0, \forall x \in [0, 1]$

$$\frac{\alpha}{\theta} \geq 6x(1-x)$$

$$\frac{\alpha}{6\theta} \geq x - x^2$$

$$x^2 - x + \frac{\alpha}{6\theta} \geq 0$$

after some algebra,  $\frac{\alpha}{\theta} \geq 1.5$ .

Then, the optimization problem is reduced to

$$\hat{\theta}, \hat{\alpha} = \underset{\theta, \alpha}{\operatorname{argmin}} \alpha, \text{ s.t. } \frac{\alpha}{\theta} \geq 1.5, \text{ and } \theta \geq 1$$

Using Lagrangian multipliers,  $L(\alpha, \theta, \lambda, \gamma) = \alpha - \lambda(\frac{\alpha}{\theta} - 1.5) - \gamma(\theta - 1)$ , we solve

*Appendix A. Example of analytic solution for optimal envelop function*

the following system of equations

$$\begin{aligned}\frac{\partial L}{\partial \alpha} &= 1 + \frac{\lambda}{\theta} = 0 \\ \frac{\partial L}{\partial \theta} &= -\lambda \frac{\alpha}{\theta^2} + \gamma = 0 \\ \frac{\partial L}{\partial \lambda} &= \frac{\alpha}{\theta} - 1.5 = 0 \\ \frac{\partial L}{\partial \gamma} &= \theta - 1 = 0\end{aligned}$$

Then, the solution is  $\theta = 1$ , and  $\alpha = 1.5$ , which results in the optimal

$$e(x) = 1.5 \text{ Uniform}(0, 1),$$

which is intuitively correct since the maximum value for Beta(2, 2) is 1.5.

# Appendix B

## Code samples

### B.1 Baseline results

The code for running multiple classifiers is publicly available on github <http://github.com/alvarouc/polyssifier> with GPL license. It builds on top of two popular python libraries: Scikit-learn <http://scikit-learn.org> and Keras <http://keras.io>.

We named our library "polyssifer" and it can be used either from the terminal or in python code as follows:

#### Bash terminal sample

```
poly data.npy label.npy --name Moons --concurrency 8
```

#### Python sample



## Appendix B. Code samples

```
from polyssifier import poly, plot
scores, confusions, predictions = poly(data, label,
    n_folds=8, concurrency=4)
plot(scores)
```

## B.2 Multilayer Perceptron

The Keras library provides tools for the design of deep-learning classifiers, however, it was not compatible with scikit-learn. Thus, we wrote a wrapper code for making it compatible with scikit-learn and easy to use. The code is publicly available on <http://github.com/alvarouc/mlp>

### Python sample

```
from mlp import MLP
from sklearn.cross_validation import cross_val_score
clf = MLP(n_hidden=10, n_deep=3, l1_norm=0, drop=0.1,
    verbose=0)
scores = cross_val_score(clf, data, label, cv=5,
    n_jobs=1, scoring='roc_auc')
```

## B.3 Brain graphics

We created a library for plotting brain views from weight patterns of voxels. The code is publicly available on [http://github.com/alvarouc/brain\\_utils](http://github.com/alvarouc/brain_utils).

*Appendix B. Code samples*

**Python sample**

```
from brain_utils import plot_source
plot_source(source, template, np.where(mask), th=th,
            vmin=th, vmax=np.max(t), cmap='hot', xyz=xyz)
```

# References

- Huda Akil, Maryann E Martone, and David C Van Essen. Challenges and opportunities in mining neuroscience data. *Science (New York, NY)*, 331(6018):708, 2011.
- Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.
- Anthony J Bell and Terrence J Sejnowski. Learning the higher-order structure of a natural sound. *Network: Computation in Neural Systems*, 7(2):261–266, 1996.
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.
- Nicolle Correa, Tülay Adalı, and Vince D Calhoun. Performance of blind source separation algorithms for fmri analysis using a group ica method. *Magnetic resonance imaging*, 25(5):684–694, 2007.
- Rémi Cuingnet, Emilie Gerardin, Jérôme Tessieras, Guillaume Auzias, Stéphane Lehericy, Marie-Odile Habert, Marie Chupin, Habib Benali, Olivier Colliot, Alzheimer’s Disease Neuroimaging Initiative, et al. Automatic classification of

## REFERENCES

- patients with alzheimer’s disease from structural mri: a comparison of ten methods using the adni database. *neuroimage*, 56(2):766–781, 2011.
- George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE, 2013.
- Dai Dai, Jieqiong Wang, Jing Hua, and Huiguang He. Classification of adhd children through multimodal magnetic resonance imaging. *Front Syst Neurosci*, 6(63):1–8, 2012.
- Arnaud Delorme and Scott Makeig. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1):9–21, 2004.
- Ashraf Elsayed, Frans Coenen, Marta García-Fiñana, and Vanessa Sluming. Region of interest based image classification using time series analysis. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–6. IEEE, 2010.
- Rosemary Fama and Edith V Sullivan. Thalamic structures and associated cognitive functions: Relations with age and aging. *Neuroscience & Biobehavioral Reviews*, 54:29–37, 2015.
- Center for Behavioral Health Statistics and Quality. Behavioral health trends in the united states: Results from the 2014 national survey on drug use and health (hhs publication no. sma 15-4927, nsduh series h-50), 2015. URL <http://www.samhsa.gov/data/>. Retrieved from <http://www.samhsa.gov/data/>.
- Alex Fornito, Murat Yücel, and Christos Pantelis. Reconciling neuroimaging and neuropathological findings in schizophrenia and bipolar disorder. *Current opinion in psychiatry*, 22(3):312–319, 2009.

## REFERENCES

- T Frodl and N Skokauskas. Meta-analysis of structural mri studies in children and adults with attention deficit hyperactivity disorder indicates treatment effects. *Acta Psychiatrica Scandinavica*, 125(2):114–126, 2012.
- Shawn D Gale, L Baxter, N Roundy, and SC Johnson. Traumatic brain injury and grey matter concentration: a preliminary voxel based morphometry study. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(7):984–988, 2005.
- Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.
- Cota Navin Gupta, Vince D Calhoun, Srinivas Rachakonda, Jiayu Chen, Veena Patel, Jingyu Liu, Judith Segall, Barbara Franke, Marcel P Zwiers, Alejandro Arias-Vasquez, et al. Patterns of gray matter abnormalities in schizophrenia based on an international mega-analysis. *Schizophrenia bulletin*, page sbu177, 2014.
- Cota Navin Gupta, Vince D Calhoun, Srinivas Rachakonda, Jiayu Chen, Veena Patel, Jingyu Liu, Judith Segall, Barbara Franke, Marcel P Zwiers, Alejandro Arias-Vasquez, et al. Patterns of gray matter abnormalities in schizophrenia based on an international mega-analysis. *Schizophrenia bulletin*, 41(5):1133–1142, 2015.
- Jeanny Hérault and Bernard Ans. Réseau de neurones à synapses modifiables: Décodage de messages sensoriels composites par apprentissage non supervisé et permanent. *Comptes rendus des séances de l'Académie des sciences. Série 3, Sciences de la vie*, 299(13):525–528, 1984.
- Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

## REFERENCES

- Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.
- SPM8 Manual*. Institute of Neurology, UCL, 12 Queen Square, London WC1N 3BG, UK, February 2012. URL <http://www.fil.ion.ucl.ac.uk/spm/>.
- Kiyoto Kasai, Martha E Shenton, Dean F Salisbury, Yoshio Hirayasu, Chang-Uk Lee, Aleksandra A Ciszewski, Deborah Yurgelun-Todd, Ron Kikinis, Ferenc A Jolesz, and Robert W McCarley. Progressive decrease of left superior temporal gyrus gray matter volume in patients with first-episode schizophrenia. *American Journal of Psychiatry*, 2003.
- Michael J Kearns and Yishay Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *ICML*, volume 98, pages 269–277. Citeseer, 1998.
- Kuryati Kipli, Abbas Z Kouzani, and Matthew Joordens. Evaluation of feature selection algorithms for detection of depression from brain smri scans. In *Complex medical engineering (CME), 2013 ICME international conference on*, pages 64–69. IEEE, 2013.
- Laura Koenders, Marise WJ Machielsen, Floor J van der Meer, Angelique CM van Gasselt, Carin J Meijer, Wim van den Brink, Maarten WJ Koeter, Matthan WA Caan, Janna Cousijn, Anouk den Braber, et al. Brain volume in male patients with recent onset schizophrenia with and without cannabis use disorders. *Journal of psychiatry & neuroscience: JPN*, 40(3):197, 2015.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

## REFERENCES

- Quoc V Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- Y. Li, T. Adahi, and V. Calhoun. Estimating the number of independent components for functional magnetic resonance imaging data. *Hum Brain Mapp*, 28(11):1251–1266, 2007.
- Jingyu Liu, Alvaro Ulloa, Nora Perrone-Bizzozero, Ronald Yeo, Jiayu Chen, and Vince D Calhoun. A pilot study on collective effects of 22q13.31 deletions on gray matter concentration in schizophrenia. *PloS one*, 7(12):e52865, 2012.
- Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- Martin J McKeown, Scott Makeig, Greg G Brown, Tzyy-Ping Jung, Sandra S Kindermann, Anthony J Bell, and Terrence J Sejnowski. Analysis of fmri data by blind separation into independent spatial components. Technical report, DTIC Document, 1997.
- Shashwath A Meda, Nicole R Giuliani, Vince D Calhoun, Kanchana Jagannathan, David J Schretlen, Anne Pulver, Nicola Cascella, Matcheri Keshavan, Wendy Kates, Robert Buchanan, et al. A large scale (n= 400) investigation of gray matter differences in schizophrenia using optimized voxel-based morphometry. *Schizophrenia research*, 101(1):95–105, 2008.
- Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes? In *CEAS*, pages 27–28, 2006.

## REFERENCES

- Ralph-Axel Müller, Patricia Shih, Brandon Keehn, Janae R Deyoe, Kelly M Leyden, and Dinesh K Shukla. Underconnected, but how? a survey of functional connectivity mri studies in autism spectrum disorders. *Cerebral Cortex*, 21(10):2233–2243, 2011.
- Katherine L Narr, Robert M Bilder, Arthur W Toga, Roger P Woods, David E Rex, Philip R Szeszko, Delbert Robinson, Serge Sevy, Handan Gunduz-Bruce, Yung-Ping Wang, et al. Mapping cortical thickness and gray matter concentration in first episode schizophrenia. *Cerebral cortex*, 15(6):708–719, 2005.
- Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5. Granada, Spain, 2011.
- Emily J Ozer, Suzanne R Best, Tami L Lipsey, and Daniel S Weiss. Predictors of posttraumatic stress disorder and symptoms in adults: a meta-analysis. In *Annual Meeting of the International Society for Traumatic Stress Studies, 14th, Nov, 1998, Washington, DC, US; This article is based on a paper presented at the aforementioned meeting.*, page 3. Educational Publishing Foundation, 2008.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE, 2014.



## REFERENCES

- RP Rajarethinam, JR DeQuardo, R Nalepa, and R Tandon. Superior temporal gyrus in schizophrenia: a volumetric magnetic resonance imaging study. *Schizophrenia research*, 41(2):303–312, 2000.
- Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- Patricia Rzezak, Paula Squarzoni, Fabio L Duran, Tania de Toledo Ferraz Alves, Jaqueline Tamashiro-Duran, Cassio M Bottino, Salma Ribeiz, Paulo A Lotufo, Paulo R Menezes, Marcia Scazufca, et al. Relationship between brain age-related reduction in gray matter and educational attainment. *PloS one*, 10(10):e0140945, 2015.
- Mert R Sabuncu, Ender Konukoglu, Alzheimers Disease Neuroimaging Initiative, et al. Clinical prediction from structural brain mri scans: A large-scale empirical study. *Neuroinformatics*, pages 1–16, 2014.
- Mark Schmidt, Alexandru Niculescu-Mizil, Kevin Murphy, et al. Learning graphical model structure using l1-regularization paths. In *AAAI*, volume 7, pages 1278–1283, 2007.
- Hugo G Schnack, Mireille Nieuwenhuis, Neeltje EM van Haren, Lucija Abramovic, Thomas W Scheewe, Rachel M Brouwer, Hilleke E Hulshoff Pol, and René S Kahn. Can structural mri aid in clinical classification? a machine learning study in two independent samples of patients with schizophrenia, bipolar disorder and healthy subjects. *Neuroimage*, 84:299–306, 2014.
- Stephen M Smith, Mark Jenkinson, Mark W Woolrich, Christian F Beckmann, Timothy EJ Behrens, Heidi Johansen-Berg, Peter R Bannister, Marilena De Luca, Ivana Drobnjak, David E Flitney, et al. Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage*, 23:S208–S219, 2004.

## REFERENCES

- Elizabeth R Sowell, Sarah N Mattson, Eric Kan, Paul M Thompson, Edward P Riley, and Arthur W Toga. Abnormal cortical thickness and brain–behavior correlation patterns in individuals with heavy prenatal alcohol exposure. *Cerebral Cortex*, 18(1):136–144, 2008.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- Ryuichi Takahashi, Kazunari Ishii, Tatsuya Kakigi, and Kazumasa Yokoyama. Gender and age differences in normal adult human brain: Voxel-based morphometric study. *Human brain mapping*, 32(7):1050–1058, 2011.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- E Fuller Torrey. Schizophrenia and the inferior parietal lobule. *Schizophrenia research*, 97(1):215–225, 2007.
- Alvaro Ulloa, Eric Verner, and Sergey Plis. Application of a universal multi-model classification tool to structural magnetic resonance imaging. *BioIT World conference and expo*, 2016.
- Anoukh Van Giessen. *Dimension reduction methods for classification; MRI-based automatic classification of Alzheimer’s disease*. PhD thesis, TU Delft, Delft University of Technology, 2012.
- Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

## REFERENCES

- Gui-Rong Xue, Dikan Xing, Qiang Yang, and Yong Yu. Deep classification in large-scale text hierarchies. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 619–626. ACM, 2008.
- Peter N Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, volume 93, pages 311–321, 1993.
- Robert J Young and Edmond A Knopp. Brain mri: tumor evaluation. *Journal of Magnetic Resonance Imaging*, 24(4):709–724, 2006.
- Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.
- 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.