12-2014

The use of variable-bagging and the cross-validation selector in the prediction of alzheimer's using the adni database.

Michael Wayne Godbey
University of Louisville

THE USE OF VARIABLE-BAGGING AND THE CROSS-VALIDATION SELECTOR IN THE PREDICTION OF ALZHEIMER'S USING THE ADNI DATABASE

By

Michael Wayne Godbey
B.A., West Virginia University, 1982
M.A., Marshall University, 1990

A Dissertation
Submitted to the Faculty of the
College of Arts and Sciences of the University of Louisville
In Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Mathematics
University of Louisville
Louisville, Kentucky

December 2014

THE USE OF VARIABLE-BAGGING AND THE CROSS-VALIDATION SELECTOR IN THE PREDICTION OF ALZHEIMER'S USING THE ADNI DATABASE

By

Michael Wayne Godbey
B.A., West Virginia University, 1982
M.A., Marshall University, 1990

Dissertation Approved on

October 17, 2014

by the following Dissertation Committee:

_____
Dr. Ryan Gill, Director

_____
Dr. Kiseop Lee

_____
Dr. Jiaxu Li

_____
Dr. Prasanna Sahoo

_____
Dr. Patrick Shafto

ACKNOWLEDGMENTS

I would like to thank my dissertation director, Dr. Ryan Gill, for his guidance and patients with novel ideas that might be construed by some as outlandish. I would also like to thank the other committee members, Dr. Prasanna Sahoo, Dr. Patrick Shafto, Dr. Jiaxu Li and Dr. Kiseop Lee, for their words of perseverance and encouragement.

Foremost, I would like to thank the Alzheimer's patients and volunteers who gave of their time and energy in the fight against this dreaded disease.

ABSTRACT

THE USE OF VARIABLE-BAGGING AND THE CROSS-VALIDATION SELECTOR IN THE PREDICTION OF ALZHEIMER'S USING THE ADNI DATABASE

Michael W. Godbey

October 17, 2014

Dimensionality plays a huge part in the modeling process. If there are more elements in a data set than variables in each element then there are very few restrictions in selection of an algorithm. Bagging, bootstrap aggregating (Breiman, 1994), may also be used to improve a model's prediction capability. On the other hand, if there more variables in each observation than the number of observations in the dataset, the number of usable algorithms is greatly reduced. The recently developed algorithm, support vector machines, was designed for such situations, in comparison to algorithms such as logistic regression which have instability issues caused by the dimensionality. Localizing or reducing the variables is an option if the loss of information is of little importance. This paper introduces a method called variable bagging (a term which was inspired by bagging) which lifts the barrier imposed by dimensionality. Instead of randomly selecting elements of the data set and using all the variables, variable bagging randomly selects variables and uses all the resultants of the data set to develop an appropriate model chosen by the cross-validation selector. The procedure is repeated several times until a committee is formed in order to "vote" on the final outcome. Theatrical results

justifying use of the cross-validation selector are also discussed. In particular, this paper obtains and proves an improved upper bound for the risk of the cross-validation selector compared with similar upper bounds in existing literature.

TABLE OF CONTENTS

ACKNOWLEDGMENTS……..……. iii

ABSTRACT..... iv

LIST OF TABLES…….... ix

LIST OF FIGURES…….…… x

1 THE DATA ..…….…… 1

1.1 Alzheimer's Disease……. 1

1.1.1 Degenerative changes …….. 2

1.1.2 Anatomy of the Brain …….. 3

1.1.3 Progression of Alzheimer's ……. 5

1.2 Description of the Data Archive…….…….. 6

1.3 Outline of the Remainder of the Paper …….…... 8

2 IMAGE PREPROCESSING ..…….... 10

2.1 Preliminary Decisions: Choosing the Pre-processing Algorithm ..……. 11

2.2 SPM's DARTEL…….. 14

3 METHODS…..…… 20

3.1 Local Logistic Regression…… 20

3.2 Support Vector Machines……. 25

3.2.1 Background Theorems ……..….. 25

3.2.2 The Formulation of the Problem ...…….. 28

3.3 Neural Networks……..... 33

3.4 Decision Trees…….. 37

3.5 Baggin …… 40

4 THE CROSS-VALIDATION SELECTOR AND VARIABLE BAGGING..… 44

4.1 Introduction…….. 45

4.2 Background Lemmas ……... 47

4.3 Cross Validation Selector……. 58

4.3.1 Notation……..…….. 59

4.3.2 The Convergence of Conditional and Optimal Risks…..……..…. 62

4.3.3 Discussion…… 74

4.4 Variable Bagging …… 79

5 EXAMPLES...……..81

5.1 Example One: “Hand Picking” the Variables…….. 82

5.1.1 Dimension Reduction……83

5.1.2 Further Reduction of Variables…… 84

5.1.3 Results…….. 87

5.2 Example Two: Variable Bagging……. 89

5.2.1 Case 1: Variables with t-values < -1.96 ……..… 89

5.2.2 Case 2: Variables with t-values > 1.96 …… 91

5.2.3 Case 3: Mixture of Data with t-values < -1.96 and > 1.96…….. 93

5.2.4 Discussion…… 94

5.3 Example Three: Using K-Fold Cross Validation Selector...……..... 95

5.3.1 Using the Cross-Validation Selector – Individual Case …….. 96

5.3.2 Using the Cross-Validation Selector inside Variable–Bagging ….……..98

6. DISCUSSION ……... 101

6.1 Overview…….…… 101

6.2 Advantages ……. 103

6.3 Disadvantages …..……. 104

6.4 Ideas for Improvement and for the Future ……..…… 105

REFERENCES …….. 107

CURRICULUM VITAE ……...……. 111

LIST OF TABLES

Table 1: The success rate using the five specific point clusters …….....…….. 88

Table 2: Applying variable-bagging with the variables having t-values < -1.96 ……. 90

Table 3: Applying variable-bagging with the variables having t-values > 1.96 ……. 92

Table 4: Applying variable-bagging with the variables having both t-values< -1.96 and t-values > 1.96 …….…….....……... 94

Table 5: Running the cross validation selector on 100 individual trials …….. 97

Table 6: Comparison of 100 variable-bagged trials for each algorithm ……..……... 99

Table 7: How the cross-validation within variable-bagging ranked……. 100

LIST OF FIGURES

Figure 1: Main parts of the brain…… 4

Figure 2: The changes of the brain …… 6

Figure 3: Hoeffding's inequality vs. Bernstein's inequality w.r.t. s ….....…..…... 79

Figure 4: Hoeffding's inequality vs. Bernstein's inequality w.r.t. n ……..……... 77

Figure 5: Hoeffding's inequality vs. Bernstein's inequality w.r.t. δ ……...…...…….. 78

Figure 6: The graph of success rates for t-values vs. local logistic regression .……... 85

1 THE DATA

In any study using statistical methods and data mining techniques, the first order of business is the gathering of data. The second is the transformation of this data into usable information.

The data in this case comes from The Laboratory of Neuro Imaging (LONI) Data Archive at UCLA, in particular the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu). The transformation of this collection of MRI brain scans into something which is both normalized and numerical requires some preprocessing. All of this is needed in order to examine and develop statistical models which will distinguish between two groups of individuals, those with Alzheimer's disease (AD) and those in a normal control group (NC).

1.1 Alzheimer's Disease

Dementia is a devastating disease affecting memory and intellectual functions of the brain. By definition, "patients with dementia must have memory disturbances as well as defects on other mental abilities such as abstract thinking, judgment, personality, language praxis and visuospatial skills. The defects must be of sufficient magnitude to

interfere significantly with work or social activities." (American Psychiatric Association, 1994) It is estimated that between 50 to 60% of all dementia cases in the United States and Europe is due to Alzheimer's disease (AD) (R. P. Friedland & Wilcock, 2000). At the age of 65, a person has a 5.1% chance of developing AD in their lifetime and half of all people age 95 and older have some form of AD according to reports presented by the Government Accounting Office (1998).

1.1.1 Degenerative changes

The description below describes a seven-stage system by Dr. Barry Reisberg of the New York University School of Medicine's Silberstein Aging and Dementia Research Center.

In the beginning, a person having AD shows no signs of impairment. A person may have AD for up to 20 years before any degenerative changes or clinical assessment has been made (Gomez-Isla et al., 1996). Eventually, a small instance of memory lapses, mislaying objects or forgetting the proper word starts to creep in typical life. Still these things cannot be clinically classified as AD and might be explained as part of the normal aging process. The decline continues to a classification of mild cognitive impairment (MCI) where family and friends may begin to notice the person having a problem coming up with a proper word to use, having trouble remembering a name or misplacing objects. The person may start to have trouble with concentration, organization and planning.

Clear cut symptoms can be detected in the early stage of AD. A person may start to become moody or withdrawn in social situations. There is difficulty in decision

making such as budgeting, paying bills and even planning what to have for dinner. There is also forgetfulness of most recent and reoccurring events.

In mid-stage AD, gaps in memory are quite noticeable. Significant details about their life and family are still remembered; however incidental aspects may be forgotten, such as their own address or phone number. The person may even become confused about where they are.

Symptoms continue to worsen to a moderately severe AD. Memory worsens to a point where they may forget their spouse's name. There may also be changes in personality. Changes in sleep patterns occur where the patient tends to sleep during the day and is up during the night. They also tend to wonder and bladder and bowel control may become an issue.

In the end, the individual loses the ability to respond to their environment. They need assistance with their daily care, personal hygiene and eating. Control of movement is lost; as they may not be able to hold up their head and may have trouble swallowing.

1.1.2 Anatomy of the Brain

The three main parts of the brain are, the cerebrum or also known as the cortex, which is the most prominent and noticeable, which is divided into two hemispheres, the cerebellum which is tucked under the cerebral hemispheres and the brain stem which connects the brain to the spinal cord, which is located in front of the cerebellum and below the cortex.

Each hemisphere of the cerebral cortex is divided into four lobes: the frontal, parietal, occipital and temporal lobes. The frontal lobes are at the front of the cerebrum. The functions include reasoning, problem solving, planning and personal expression. The parietal lobes are behind the frontal lobes and are responsible for information processing, recognition, the sense of touch, speech and cognition. The occipital lobes are at the back portion of the cerebral cortex. These lobes are the center for visual perception and color recognition. The temporal lobes are located on the sides of the cerebral cortex. The functions include: visual memories, short term memories, language recognition, emotion and processing sensory input (Dawbarn & Allen, 2007). The parts of the brain and lobes are illustrated in Figure 1.

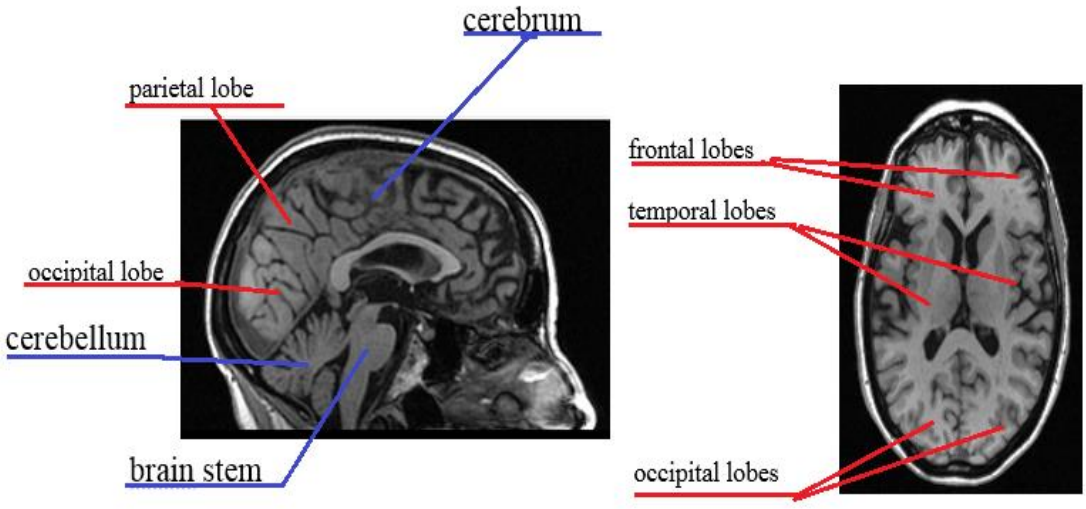

Figure 1:Main parts of the brain.

1.1.2 Progression of Alzheimer's

Despite all the statistics on the number of AD patients, diagnosis for AD is currently based on clinical and psychometric assessment. A definite diagnosis can only be made by having both amyloid plaques and neurofibrillary tangles (R. P. Friedland, 2010). A clear diagnosis of AD is usually made at autopsy.

Amyloid plaques consist of deposits of aluminum silicate and bits of beta-amyloid protein (wiseGEEK.com; alz.org). These deposits clumped together form an insoluble plaque which builds up on the outside of neurons. A more modern theory suggest that the smaller pieces of the soluble beta-amyloid deposits known as oligomers and not the large invaluable amyloid plaques are the culprit that physically disrupts signing at the synapse of the neurons (Schnabel, 2010). This disruption distresses the neuron causing the activation of the immune system leading to cell death (alz.org).

Tau proteins, which are abundant in the neurons of the brain, are proteins that stabilize microtubules (Dawbarn and Allen, 2007). Microtubules are long; hallow tubes that help maintain the structure of the cell and which act as a transport system for key materials and nutrients needed by the cell. In a hyperphosphorylated state, mutations occur and cause tau to dysfunction. Either tau losses the ability to interact with microtubules or there is an overproduction of tau (Goedert & Spillantini, 2000) causing tau to collapse into twisted bundles called tau tangles, which clog the microtubules (Alzheimer's Association). In the end the microtubules kink and eventually disintegrate.

In the earliest stages, amyloid plaque and neurofibrillary tangles start to form in and around the areas of the hippocampus and amygdala, both deep inside the temporal

lobes. These areas are associated with long term memories and emotions. As the disease progresses, the amyloid plaques and tau tangles build up in those areas associated with memory and spread through the temporal lobes which affects the speaking and understanding language, and into the occipital lobes which is associated with orientation of self to the surrounding environment.

In the final stages of Alzheimer's, most of the cerebral cortex is seriously damaged. The brain and especially the hippocampus have dramatically shrunk. The ventricles have grossly enlarged. The ability to recognize family and to care for themselves is lost. Figure 2 illustrates tissue atrophy and enlarged ventricles that can be seen for a sagittal slice of a brain image from an Alzheimer's patient when compared with that of a normal patient.

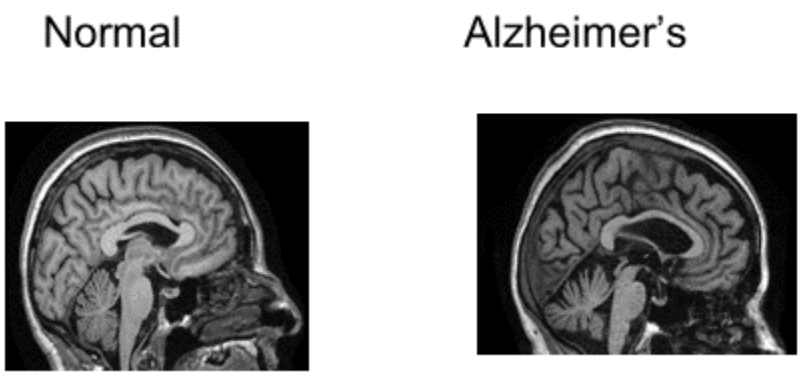

Figure 2: The changes of the brain. The cerebral cortex shrinks causing the gyri (ridges) and sulci (grooves) to become more pronounced and the ventricles become larger.

1.2 Description of the Data Archive

"Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu). The ADNI

was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a $60 million, 5-year public-private partnership. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials.

"The Principal Investigator of this initiative is Michael W. Weiner, MD, VA Medical Center and University of California – San Francisco. ADNI is the result of efforts of many coinvestigators from a broad range of academic institutions and private corporations, and subjects have been recruited from over 50 sites across the U.S. and Canada. The initial goal of ADNI was to recruit 800 adults, ages 55 to 90, to participate in the research, approximately 200 cognitively normal older individuals to be followed for 3 years, 400 people with MCI to be followed for 3 years and 200 people with early AD to be followed for 2 years." (Further information can be found at www.adni-info.org.)

The ADNI study has brain image data which includes MRI and PET images which have been validated by the study. The Laboratory of Neuro Imaging (LONI) Data Archive at UCLA is the environment through which outside investigators can obtain access to the data. The ADNI study's brain image data is not public, but is made available to the general scientific community through the website www.loni.ucla.edu/ADNI/ to investigators who successfully complete an on-line application. Investigators wishing to obtain access to the data must submit this on-line application which requires information about the researcher's purposed plans and requires the researcher to agree to the ADNI sharing and publication policies.

Due to the archival nature of the study, there is no direct risk to the human subjects involved stemming from their participation of the study. All data has already

been collected and has been de-identified so that an investigator will have no access to protected health information (PHI). As required, I have successfully filled out an application and been granted access to the data by ADNI's Data Publication and Sharing Committee as indicated at http://www.loni.ucla.edu/ADNI/Data/ADNI_DataAccounts.jsp. Furthermore, I have successfully completed the CITI training course for Biomedical Responsible Conduct of Research and the CITI training course for Human Research.

1.3 Outline of the Remainder of the Paper

After the selection of the data, the MRI images must be converted into a useable form. Chapter 2 describes the thought that went into finding a preprocessing program followed by a cookbook recipe on how to use the chosen program. There are many image preprocessing programs available ranging from open source programs from the internet to those programs which are sold for thousands of dollars. After reading several articles rating various programs, SPM's DARTEL open source image preprocessing program was selected.

Chapter 3 describes the statistical methods used in the development of this paper. No new methods are introduced in this section. Methods used include: logistic regression, support vector machines, neural networks, decision trees and bagging. The derivation and description of these methods can be easily be found in statistical and/or data mining texts.

Chapters 4 and 5 are the major sections of the paper. Chapter 4 starts out by describing and presenting the theoretical reasoning behind the cross-validation selector.

The section begins by introducing and proving background lemmas which include: Markov's inequality, Chebyshev's inequality, Chernoff's bounding methods, Bernstein's inequality, and Hoeffding's inequality. All these lemmas may be found in a source about nonparametric regression and are needed in order to prove Dudoit and van der Laan's (2003) theorem on the upper bound for the risk of the cross validation. I was able to obtain and improve this bound as shown in Theorem 2. The section ends with the introduction of variable bagging, a method I developed specifically to aid those algorithms which have issues with dimensionality.

Chapter 5 provides three examples illustrating the ideas presented. The first example shows what a classical statistician might do in order to apply the logistic regression algorithm to a dimensionally dense data source such as MRI brain images. For this case, dimension reduction is mandatory. Example two provides an alternative solution to the problem by using variable bagging. The example also shows off the boosting capabilities of variable bagging by dramatically improving success rates of predictability. The third example unlocks the restraints of using just one algorithm for modeling by incorporating the cross-validation selector into the variable bagging procedure. The cross-validation selector allows the investigator to use many algorithms in the modeling process. As a result, the cross–validation selector can produce success rates which are better than the success rates of any singular algorithm considered.

Finally, the paper ends with a discussion of the findings.

2 IMAGE PREPROCESSING

It is important to perform pre-processing on the T1 weighted structural MRIs before applying the statistical methods to the data. Voxel-based analysis relies heavily on the accuracy of the matching of anatomical regions from subject to subject, i.e. spatial normalization. Matching skull features does not necessarily provide a good match for the anatomical regions of brain tissue (Tosun-Turgut, 2012). Thus skull stripping is usually performed before any spatial normalization.

In pre-processing the images, it is first proposed that the N3 correction algorithm (Sled & Pike, 1998) should be used to iteratively estimate a smooth intensity mapping function and sharpen the peaks in the image histogram. Then, an intensity normalization step is used to remove outlier intensity values by eliminating intensity values below the percentile 0.1 and above the percentile 99.9. Then, spatial normalization is performed by transforming the coordinate system to a standard brain-based system (stereotaxic space) so that similar anatomical structures from different data sets are mapped to an equivalent system (Fox, Perlmutter, & Raichle, 1985; Mazziotta, Toga, Evans, Fox, & Lancaster, 1995). The spatial normalization step also includes a registration algorithm (Collins, Neelin, Peters, & Evans, 1994) using an average MRI image based on optimizing 9 parameters (3 translations, 3 rotations, and 3 scalings). Finally, a second pass of spatial normalization is performed to correct the intensities of each image with respect to the

patient-specific stereotaxic target after intensity normalization using least trimmed squares (Rousseeuw & Leroy, 1987).

Selecting a pre-processing algorithm proved to be difficult, especially for someone not in the field of preprocessing. There are many image preprocessing programs available ranging from the open source programs to those programs which are sold for thousands of dollars. In the fast pace world of pre-processing development, many projects have not kept pace with the recent ideas about spatial alignment and nonlinear deformation. Many algorithms available have, in fact, been abandoned by the developer and have become obsolete. Two well-known packages that are highly accepted in dementia research are LLDMM and SPM's DARTEL. These nonlinear high dimensional warping algorithms are well suited for accurate localized anatomical matching which is needed for voxel-based analysis. In the end, SPM's DARTEL was selected to do the pre-processing. Section 2.2 list the steps needed for the preprocess procedure.

2.1 Preliminary Decisions: Choosing the Pre-processing Algorithm

The paper "Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration" by Klein et al [2009] provided a good starting point and source for several of the freely available software. In short, the results of their study concluded that ART, SyN, IRTK and SPM's DARTEL were ranked highest in pre-processing of MRI brain scans. However, the value of the conclusions came into question when the

paper pointed out that the comparisons were made on "normal" brains. Alzheimer's brains cannot be considered "normal" because of the presence of lesions, enlarged ventricles and the presence of areas of atrophy.

Brett et al. [2001] advocated cost function masking and found that cost function masking significantly improves non-linear registration/normalization results. This method became the accepted method which overcame the difficulties related to the normalizing damaged brains. In cost function masking, the voxels representing abnormalities are masked or blotted out, and the remaining regions of the brain are registered to the target. After the registration, the masked areas are reinserted. These abnormalities are identified by calculating the cost function or the "distances" between the intensities of the image and the target. If this distance (cost function) is too large, the area is masked. Andersen [2010] continued to agree with Brett's results and concluded that the failure to use cost function masking results in less accurate results in terms of i) deformation field displacement, ii) voxelwise intensities of the lesion areas and iii) a significant underestimation of lesion volume.

Cost function masking, however, may not be a viable choice, especially when registering a population with lesions that are large and/or are bilateral. These traits are common to Alzheimer's patients (Kim, Avants, Patel, & Whyte, 2008).

Again, the idea of cost function masking is to first blot out defective areas, register the remaining brain tissue to a template and finally fit the masked portion to the resulting empty space, usually by affine transformations (Brett, Leff, Rorden, & Ashburner, 2001). However, these affine transformations would be undesirable because of the non-linear nature of damaged portion of the brain, which for example might

include: the region being non-symmetric within itself and to the rest of the brain, the random occurrence of atrophy and/or ventricular enlargement. A non-linear transformation for the injured portions of the brain would seem to be more desirable.

Common non-linear transformations are based on linear combinations of polynomials or functions of cosine basis. But again, these approaches have their limitations due to the assumption that the damaged area is small. On the other hand, the DARTEL toolbox in SPM, the FLIRT tool in FSL, and the SyN in ANTS algorithms were designed using a diffeomorphism which can implement spatial normalization applications with large areas of deformation (Kim et al., 2008).

Two well-known packages, LLDMM (C++ based) and DARTEL/SPM (Matlab based), were recommended by Duygu Tosun-Turgut, an University of California San Francisco (UCSF) School of Medicine Assistant Professor (Through personal correspondence, June 29, 2012). Professor Tosun-Turgut explained that these two nonlinear high dimensional warping algorithms are well suited for accurate localized anatomical matching which is needed for voxel-based analysis. The programs have also been around for many years and are highly accepted in dementia research.

Thus, based upon the discussions by Klein et al., Kim et al., and the recommendation by Dr. Tosun-Turgut, the Anatomical Registration Through Exponentiated Lie algebra (DARTEL) toolbox in SPM8 was chosen to perform the preprocessing and registration of the sample images.

2.2 SPM's DARTEL

As the name indicates, DARTEL (Diffeomorphic Anatomical Registration Through Exponential Lie algebra) is a diffeomorphic algorithm. A diffeomorphism is a one-to-one continuously differentiable mapping $f : M \rightarrow N$ of a differentiable manifold M into a differentiable manifold N in which the inverse mapping is also continuously differentiable (www.Encyclopediaofmath.org). As a result, this mapping will preserve topology (Ashburner, 2007) which is important in the circular nature of DARTEL.

DARTEL uses two approaches in the segmentation of brain images: tissue classification and registration to a template (Ashburner & Friston, 2005).

Tissue classification uses the intensities of each voxel. The intensity distribution of any individual image can be represented by a mixture of three normal distributions representing the cerebrospinal fluid (CSF), gray matter (GM) and white matter (WM) (Magnin et al., 2009). Each voxel is automatically classified, according to its intensity, to the particular tissue class which has the highest probability (Ashburner & Friston, 2005).

Registering to a template involves warping a template image to the subject's image. The regions of the template are pre-defined allowing for an automatic identification of the brain's structure (Ashburner & Friston, 2005).

Classification in DARTEL requires the images to first be registered with tissue probability maps which will represent the prior probabilities. These prior probabilities can then be used with tissue types classified from the image intensities to provide the posterior probabilities using Bayes rule. Thus a circular procedure revolves requiring an

initial tissue classification for registration and an initial registration for tissue classification (Ashburner & Friston, 2005).

The main disadvantage of intertwining the two procedures is that it produces a complex algorithm requiring more time to run than the two procedures running separately. The code itself also is hard to access. However, the results are more accurate.

Performing pre-processing and registration to the T1 weighted MRI's is important to insure that the various regions of the brain are lined up correctly. For the beginner the SPM8 manual, which is a 451 page pdf file, might prove to be a bit intimidating. A great alternative is the "cookbook" type discussion entitled "VBM Tutorial" by John Ashburner found at: www.fil.ion.ucl.ac.uk/~john/misc/VBMclass10.pdf . The following notes follow this tutorial closely.

The first thing to notice is that the SPM8 package is actually a folder containing algorithms and files written in the MATLAB environment. However, the knowledge of MATLAB is not necessary. (Note: As of this writing, there exists a beta version of a stand-alone SPM8 for those who do not have the MATLAB program.) The SPM8 software, along with the "patches /fixes" may be downloaded from: http://www.fil.ion.ucl.ac.uk/spm/ .

Step 0: Start up MATLAB and bring up the SPM environment.

Start up MATLAB. In the "Command Window" type: *spm pet* . (alternate: In the "Command Window" of MATLAB type *spm,* followed by the clicking on the "PET & VBM" button.)

SPM requires the T1-weighted images to be: 1) in the NIfTI format. SPM does provide a way of converting many types of images that are in the DICOM format to the required format. The images from the ADNI archive were already downloaded to the proper NIfTI format. 2) The images should be aligned within 5cm and 20 degrees of each other. The "Check Reg" button will allow you to view several images at once to check to see if the images are in the proper orientation. The "Display" button will allow you to readjust the tilt of the head or the origin of the axes. Historically the center is usually located near the anterior cingulate (AC).

Step 1: Segmenting the image, i.e. skull stripping.

Batch → SPM → Tools → New Segment

There are three main headings within the "Current Module: New Segment" window: *Data*, *Tissues*, and *Warping & MRF*. Under the *Data* heading, the path of the images which need to be segmented are defined. By highlighting the *Volume* subheading and clicking on the *Select Files* button, the path of the file which contains the images can be established. The available images will appear in the right hand window. By right clicking on the first image and clicking on *"select all"*, the path of all the images in the file will be identified for segmentation. If select files are preferred, then click on only the individual files of interest. Complete the selection by clicking the *Done* button.

Under the *Tissues* heading, there are six other *Tissue* subheadings which will identify the six tissue categories of the head. They are, in order: gray matter (GM), white matter (WM), cerebral spinal fluid (CSF), skull, soft tissue outside the brain (throat,

muscles, eyes etc), and the material outside the head. Accept all defaults under the *Tissues* heading except for the "*Native Tissue*" entries for the first two *Tissue* subheadings which were changed from "*Native Space"* to *"Native +DARTEL import".*

Accept the defaults under the *Warping & MRF* heading.

When ready, click the green triangle at the top to run the batch. It takes approximately 11 minutes per image to run. The batch will produce "Native Space" files, with prefixes c1, c2, c3, c4 and c5 representing, respectively, GM, WM, CSF etc. The batch will also produce two DARTEL import files, prefixes rc1 and rc2, which will be used in the next step of the registration process.

Step 2: Create Templates using DARTEL

In the current *Batch Editor*:

SPM → Tools → DARTEL Tools → Run DARTEL (create Templates)

In this step DARTEL will simultaneously align the GM and WM of each image to create an inter-subject average template. This is an iterative process, done at each voxel of the brain, which matches each image to an average template formed from all the images.

To define the parameters in the "Current Module: Run DATREL (create Templates)" window, first create two *Images* subheadings under the main *Images* heading. This is done by highlighting and replication the initial *Images* subheading. Then define the paths of DARTEL's imported GM images (rc1's) in the first *Images*

subheading and the imported WM images (rc2's) in the second. Make sure that the selection of the WM and GM images are made in the same order. For the rest of the settings, use the defaults.

The process takes approximately 40 minutes per image to run. The result is a series of templates (zero through six) and a "u-rc1" image for each image. The templates are averages of all the images which are registered to the MNI space, the last being the best representation of the registration. The "u_rc1" images are the estimated deformations of the brain which will be used to encode the shapes of the brains to the MNI space.

Step 3: Normalising the images to the MNI space.

In the current *Batch Editor*:

SPM → Tools → DARTEL Tools → Normalise to MNI Space

In this step, DARTEL generates images that are smooth, spatially normalized and Jacobian scaled gray matter which is in the MNI space. The final image will have the prefix of "smwc1".

The "Current Model: Normalise to MNI Space" window has six headings to be considered: i) "DARTEL Template", defines the path to the final template created in the last step. ii) In the "Select according to" heading, select "Many Subjects". This will produce the sub-heading "Flow fields" in which the paths of the "u_rc1" images will be defined. Under the sub-heading "Images" there is a double sub-heading "Images" which is used to define the paths of the gray matter images "c1". iii) The default in "Voxel"

will be used to indicate a voxel size of 1.5mm. iv) The default is also used for the "Bounding box" heading. v) Under the "Preserve" heading, choose "Preserve Amount" in order to have the tissue volumes compared used in VBM studies. The "Preserve Concentrations" choice is suggested for fMRI studies which have no modulation. vi) Finally, the "Gaussian FWHM" is the size of the standard deviation of the Gaussian used for smoothing; the lower the smoothing constant, the more accurate the alignment will be. A value of 8mm was used, i.e. [8 8 8], instead of the default of 10mm was used as suggested by Ashburner in his tutorial.

Normalising the images to the MNI space took about 50 minutes per image.

3 METHODS

The methods described in this section can be found in any good text about statistical learning theory (Hastie, Tibshirani, & Friedman 2009; Vapnik 1998). The discussion about bagging came from the original paper (Breiman, 1994). For the purpose of this paper, only four methods were considered even though many more algorithms could have been included. The four methods are: logistic regression, support vector machines, neural networks, and decision trees.

3.1 Local Logistic Regression

Let the response variable, $y_i \in \{0,1\}$, be binary for the i^{th} subject and let $y = [y_1, \cdots, y_n]^T$ be the vector of responses of all n subjects of the data set. Let x_{ij} be the intensity of the j^{th} voxel for the i^{th} subject and let $x_i = [1, x_{i1}, \cdots, x_{im}]^T$ be the vector of intensities for the i^{th} subject where m is the number of voxels in each image. Furthermore, let v_j be the location of the j^{th} voxel. Define

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

to be the $n \times (m+1)$ design matrix for all voxel intensities for the data set. Thus, at a single voxel located at $\boldsymbol{v} \in \Re^3$, logistic regression models the data as follows.

Assume $y_1, \cdots, y_n$ are independent random variables such that y_i follows a Bernoulli(p_i) distribution where

$$p_i = p_i(\beta) \equiv \frac{e^{\beta_0 + w_1\beta_1 x_{i1} + \cdots + w_m\beta_m x_{im}}}{1 + e^{\beta_0 + w_1\beta_1 x_{i1} + \cdots + w_m\beta_m x_{im}}} = \frac{\exp\left(\beta^T W x_i\right)}{1 + \exp\left(\beta^T W x_i\right)}$$

for $i = 1, \ldots, n$. Here $\beta = [\beta_0, \beta_1, \cdots, \beta_m]^T$ is the vector of regression coefficients where $\beta_0 = \beta_0(v)$ is the intercept and $\beta_i = \beta_i(v)$ is a regression coefficient for the i^{th} voxel for the model. The weights w_j are nonincreasing functions of the distances from $\boldsymbol{v}_j$ to $\boldsymbol{v}$ defined on [0, ∞), and $W = W(v)$ is a $(m+1)\times(m+1)$ diagonal matrix having the elements $w_0 = 1, w_1, \cdots, w_m$ on the diagonal. For example, one possible choice of weights is

$$w_j = w_j(v) \equiv f\left(\left\|v - v_j\right\|\right)$$

where

$$f(z) = \begin{cases} \frac{15}{16}\left(1 - \left(\frac{z}{2}\right)^2\right)^2 & if \quad z < 2 \\ 0 & if \quad z \geq 2 \end{cases}.$$

The log-likelihood function for β is

$$l(\beta) = \ln\left(\prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}\right)$$

$$= \sum_{i=1}^{n} \ln\left(p_i^{y_i}(1 - p_i)^{1-y_i}\right)$$

$$= \sum_{i=1}^{n} \left(y_i \ln\ p_i + (1 - y_i)\ln(1 - p_i)\right).$$

Using p_i from the equation above, we have

$$\ln\, p_i = \beta^T W x_i - \ln\left(1 + e^{\beta^T W x_i}\right)$$

and

$$\ln(1 - p_i) = \ln\left(\frac{1}{1 + e^{\beta^T W x_i}}\right)$$

$$= -\ln\left(1 + e^{\beta^T W x_i}\right).$$

Thus the likelihood can be written as

$$l(\beta) = \sum_{i=1}^{n}\left(y_i \beta^T W x_i - y_i \ln\left(1 + e^{\beta^T W x_i}\right) - (1 - y_i)\ln(1 + e^{\beta^T W x_i})\right)$$

$$= \sum_{i=1}^{n}\left(y_i \beta^T W x_i - \ln(1 + e^{\beta^T W x_i})\right).$$

To maximize l, find the value of β such that

$$\nabla l(\beta) = 0.$$

Thus, it follows that

$$\nabla l(\beta) = \sum_{i=1}^{n}\left(y_i W x_i - \frac{e^{\beta^T W x_i}}{1 + e^{\beta^T W x_i}} W x_i\right)$$

$$= \sum_{i=1}^{n}\left(y_i W x_i - p_i W x_i\right)$$

$$= \sum_{i=1}^{n}\left(y_i - p_i\right) W x_i$$

$$= X^{\mathrm{T}}\, W\, (y - p)$$

where $p = [p_1, \ldots, p_n]^{\mathrm{T}}$.

Note that the solution to $\nabla l(\beta) = 0$ is a global maximizer since

$$\nabla^2 l(\beta) = \frac{\partial}{\partial \beta}\left[\frac{\partial l(\beta)}{\partial \beta^T}\right]$$

$$= \sum_{i=1}^{n}\left(x_i^T W^T \frac{\partial(y_i - p_i)}{\partial \beta}\right)$$

$$= -\sum_{i=1}^{n}\left(x_i^T W^T W x_i \frac{e^{\beta^T W x_i}}{(1+e^{\beta^T W x_i})^2}\right)$$

$$= -X^T \tilde{W} X$$

which is a nonpositive definite matrix (if X is of full rank then it is negative definite) where $\tilde{W} = \tilde{W}(\beta)$ is a diagonal matrix with diagonal elements $\tilde{w}_i = w_i^2 \frac{e^{\beta^T W x_i}}{\left(1-e^{\beta^T W x_i}\right)^2}$, $i = 1, \ldots, n$.

There is no closed form to the solution of $\nabla l(\beta) = 0$, thus numerical methods are needed to find the root of the equation. The Newton-Raphson algorithm is one of the most widely used methods to find the root of equations of this form. To find the root of a non-linear *p*-dimensional equation *F(x)* = 0, the multivariate version of the Newton-Raphson algorithm is based on the first order Taylor approximation

$$F(x) = F(x_0) + \frac{\partial F}{\partial x^T}(x_0)(x - x_0) \ .$$

If x_0 is the current estimate of the root then the updated estimate of the root is as follows:

$$0 = F(x_0) + \frac{\partial F}{\partial x^T}(x_0)(x - x_0)$$

$$x = x_0 - \left(\frac{\partial F}{\partial x^T}(x_0) \right)^{-1} F(x_0).$$

In our case of local logistic regression, F is the same as ∇l, thus the iterative step of the Newton-Raphson algorithm is

$$\begin{aligned} \hat{\beta}_{new} &= \hat{\beta}_{old} - \left(\nabla^2 l(\hat{\beta}_{old}) \right)^{-1} \nabla l(\hat{\beta}_{old}) \\ &= \hat{\beta}_{old} + \left(X^T \tilde{W}(\hat{\beta}_{old}) X \right)^{-1} X^T \left(y - p(\beta_{old}) \right) . \end{aligned}$$

Hence the Newton-Raphson algorithm will proceed as follows:

1. Start with an initial estimate $\hat{\beta}_0$. A reasonable choice is the weighted least squares estimate

$$\hat{\beta}_0 = \left(X^T W^2 X \right)^{-1} X^T W y .$$

2. Update the estimate of β until convergence.

$$\hat{\beta}_{i+1} = \hat{\beta}_i + \left(X^T \tilde{W}(\hat{\beta}_i) X \right)^{-1} X^T \left(y - p(\beta_i) \right)$$

This algorithm is also called iterative reweighted least squares (IWLS) when used with logistic regression.

Note that the algorithm produces only a local maximum. The algorithm should be run several times by taking various initial value of $\hat{\beta}_0$ in the domain. A local maximum will be produced from each running of the algorithm. Taking the maximum of all the local maximums is more likely to produce the desired global maximum for $\hat{\beta}$.

3.2 Support Vector Machine

3.2.1 Background Theorems

For a more in depth discussion, please refer to Vapnik (1998). In particular refer to section 9.5: "Three Theorems of Optimization Theory".

A support vector machine (SVM) is an example of a supervised classification method. Intuitively, the idea is to not only be able to define a hyperplane separating two distinct groups but to define a hyper-boundary having a margin of 2M which separates groups. The goal of the problem is to find this hyper-boundary which has the maximum margin.

Three theorems play an important role in developing the theory associated with SVM. The first is familiar to any Calculus student wanting to optimize a function with *n* variables.

Theorem *(Fermat's theorem for functions of n variables)*

Let f be a function of n variables, $x_1, \ldots, x_n$*, and differentiable at the point* $x^* = (x_1^*, \ldots, x_n^*)$*. If* x^* *is a point of local extrema of f(x) then*

$$\frac{\partial f}{\partial x_i} = 0 \quad \text{for } i = 1, \ldots, n.$$

In the case where $f(x)$ is to be optimized given restrictive conditions, the Lagrange method can be used. By introducing variables called Lagrange multipliers to help include the constraints into the original function, the Lagrange equation helps solve many conditional optimization problems.

Theorem *(Lagrange's Theorem)*

Let the functions $f_k(x), k = 0, 1, \ldots, m$ *be continuous and differentiable in a neighborhood about the point* x^**. If* x^* *is a point of local extrema then there exist Lagrange multipliers* $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$ *and* λ_0*, where* λ_i^**'s,* $i = 1, \ldots, m$ *and* λ_0 *are not all equal to zero, such that the following conditions hold true:*

$$\frac{\partial L(x^*, \lambda^*, \lambda_0)}{\partial x_i} = 0 \quad \text{for } i = 1, \ldots, n$$

where

$$L(x^*, \lambda^*, \lambda_0) = \sum \lambda_k^* f_k(x^*) + \lambda_0 f_0(x^*)$$

is called the Lagrange function.

For example, in order to set up the Lagrange equation to find the maximizer (or minimizer) of a given function

$$f(x_1, \ldots, x_n)$$

given the constraints

$$g_j(x_1, \ldots, x_n) = b_j \qquad j = 1, \ldots, m \qquad m < n,$$

first define the Lagrange multipliers λ_j, $j = 1, \ldots, m$, to construct the Lagrange equation

$$L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j \left(b_j - g_j(x)\right).$$

In order to find the stationary points, the partial derivatives of L with respect to all the variables $x_i's$ and $\lambda_j's$ are set to zeros, i.e.

$$\frac{\partial L}{\partial x_i} \overset{set}{=} 0 \quad for\ i = 1, \ldots, n \quad \text{and} \quad \frac{\partial L}{\partial \lambda_j} \overset{set}{=} 0 \quad for\ j = 1, \ldots, m.$$

Thus in order to optimize the equation one has to solve the system of $m + n$ equations.

Lagrange was the one who introduced a method for solving the conditional optimization problem using equality type constraints. It was Kuhn and Tucker who suggested a solution to the convex optimization problem which uses constraints of inequality type.

Theorem *(Kuhn-Tucker Theorem)*

Let X be a linear space and let A be a convex subset of X. Also let $f_i(x), i = 0, 1, \ldots, m$ *be convex functions. If the point* x^* *minimizes the function* $f_0(x)$ *subject to the constraints*

$$f_k(x) \le 0, \qquad k = 1, 2 \ldots, m$$

$x \in A$,

then there exist Lagrange multipliers λ_0^* *and* $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$, *not all* $\lambda_i^{*}'s = 0$,

$i = 0, 1, \ldots, m$ *such that the following hold true:*

a) $\min_{x \in A} L(x, \lambda_0^*, \lambda^*) = L(x^*, \lambda_0^*, \lambda)$

b) $\lambda_i^* \geq 0, \qquad i = 0, 1, \ldots, m$

c) $\lambda_k^* f_k = 0 \qquad k = 1, \ldots, m.$

3.2.2 The Formulation of the Problem

For a more in depth discussion concerning the subject of this section refer to sections 10.1-10.3 of Vapnik (1998) and sections 4.5 and 12.1-12.2 of Hastie, Tibshirani, and Friedman (2009).

In the separable case, the idea is to construct a hyperplane, i.e. a linear decision boundary in hyperspace, which will distinctly separate two sets of points. Further, it is ideal to define a region symmetrically about this hyperplane having a margin, *M*, of maximum distance. The width of this region is therefore 2*M*. The support vectors in the separable case will be those points which lie *M* units from the boundary.

However, in the nonseparable case there will be points which lie on the wrong side of the margin by the amount of $\xi_j^* = M\xi_j$. Those points on the correct side of the

margin have $\xi_j^* = 0$. Thus, in the nonseparable case, the margin is maximized subject to the total distance of points on the wrong side of their margin, i.e. $\sum M\xi_j$.

Using the training data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in \Re^p$ and $y_i \in \{-1, 1\}$, we will define the hyperplane as

$$\{x : f(x) = x^T \beta + \beta_0 = 0\}$$

with $\|\beta\| = 1$. The classification rule is the sign of $f(x)$. Again the problem is to define a region symmetrically about this hyperplane having a maximum marginal distance of *M*. In other words, for all *i*, the problem for the separable case can be stated as

$$\max_{\beta, \beta_0, \|\beta\|=1} M$$

subject to

$$y_i (x_i^T \beta + \beta_0) \geq M.$$

We can eliminate the $\|\beta\| = 1$ condition by writing

$$y_i (x_i^T \frac{\beta}{\|\beta\|} + \beta_0) \geq M$$

or

$$\frac{1}{\|\beta\|} y_i (x_i^T \beta + \beta_0) \geq M.$$

This, by the way, will redefine β_0 from the previous equation.

By arbitrarily setting $\|\beta\| = \frac{1}{M}$ the problem is more conveniently written as

$$\min_{\beta, \beta_0} \|\beta\|$$

subject to

$$y_i(x_i^T\beta+\beta_0)\geq 1.$$

Now for the nonseparable case, there are points which are on the wrong side of the margin by the distance of $\xi_j^* = M\xi_j$. The problem in this case still is to maximize *M* but it is allowable to have some points on the wrong side of the boundary. Define the slack variables $\xi=(\xi_1,\ldots,\xi_n)$ such that $M\xi_i$ is the distance for which the i^{th} point is on the wrong side of the boundary. It is easy to see that for all i, $\xi_i \geq 0$ ($\xi_i = 0$ for those points that are on the correct side of the boundary). Now set $\sum_{i=1}^{n}\xi_i \leq K$, for some constant *K*. Note that misclassification occurs when $\xi_i > 1$. By setting a bound on $\sum_{i=1}^{n}\xi_i$, this sets a bound on the number of misclassifications for the training data.

Thus, the problem becomes:

$$\min_{\beta,\beta_0}\|\beta\|$$

subject to

$$y_i\left(x_i^T\beta+\beta_0\right)\geq 1-\xi_i$$

$$\xi_i \geq 0$$

$$\sum_{i=1}^{n}\xi_i \leq K.$$

The problem is now quadratic with linear constraints. In other words it is a convex optimization problem subject to Lagrange's and to Kuhn-Tucker's Theorems. It is convenient to rewrite the problem in the following equivalent form which includes the number of misclassifications in the main optimization portion of the equation.

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i$$

subject to

$$y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \xi_i$$

$$\xi_i \geq 0.$$

Hence, the primal Lagrange function becomes:

$$L_p = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left[y_i\left(x_i^T\beta + \beta_0\right) - \left(1-\xi_i\right)\right] - \sum_{i=1}^{n}\mu_i\xi_i$$

where $\alpha_i, \mu_i \geq 0$ for all *i*.

Finding the saddle point by minimizing with respect to β, β_0 and ξ_i, and maximizing with respect to α_i and μ_i, we obtain

$$\frac{\partial L_p}{\partial \beta} = \beta - \sum_{i=1}^{n}\alpha_i y_i x_i \overset{set}{=} 0 \qquad i.e. \qquad \beta = \sum_{i=1}^{n}\alpha_i y_i x_i,$$

$$\frac{\partial L_p}{\partial \beta_0} = -\sum_{i=1}^{n}\alpha_i y_i \overset{set}{=} 0 \qquad i.e. \qquad \sum_{i=1}^{n}\alpha_i y_i = 0, \text{ and}$$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - \mu_i \overset{set}{=} 0 \qquad i.e. \qquad \alpha_i = C - \mu_i.$$

Substituting back into L_p to produce the Lagrange dual problem L_D gives

$$L_p = \frac{1}{2}(\beta \cdot \beta) + \sum_{i=1}^{n}(C - \mu_i)\xi_i - \sum_{i=1}^{n}\alpha_i y_i x_i^T \beta - \beta_0 \sum_{i=1}^{n}\alpha_i y_i + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i \xi_i,$$

$$= \frac{1}{2}(\beta \cdot \beta) + \sum_{i=1}^{n}\alpha_i \xi_i - (\beta \cdot \beta) + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i \xi_i$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}(\beta \cdot \beta)$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i-1}^{n}\sum_{j=1}^{n}\alpha_i \alpha_j y_i y_j x_i^T x_j$$

Then maximize L_D subject to:

$$0 \le \alpha_i \le C$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\beta = \sum_{i=1}^{n}\alpha_i y_i x_i$$

and also the Kuhn-Tucker conditions

$$\alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \right] = 0$$

$$\mu_i \xi_i = 0$$

$$y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \ge 0.$$

Thus, β has the solution in the form

$$\hat{\beta} = \sum_{i=1}^{n}\hat{\alpha}_i y_i x_i$$

Those support vectors on the boundary of the margin have $\hat{\xi}_i = \mathbf{0}$. However, for those support vectors that are on the wrong side of the boundary, $\hat{\xi}_i \geq \mathbf{0}$, which implies that $\mu_i = \mathbf{0}$ and $\hat{\alpha}_i = C$ (since $\alpha_i = C - \mu_i$). Finally for those support vectors on the margin, ($\hat{\xi}_i = \mathbf{0}$) along with $0 < \alpha_i$, so that

$$\hat{\beta}_0 = y_i - x_i^T \hat{\beta}.$$

3.3 Neural Networks

Artificial neural networks had its motivation by the desire to model the human brain by a computer (Györfi, Kohler, Krzyżak, & Walk, 2002). This was one of the first ideas of how to construct a "learning machine". A simple model was introduced by McCulloch and Pitts (1943), where they modeled the neuron by a real valued function, g(*x*), in $\Re^d$ which would apply a threshold function to a linear weighted combination of inputs. The range of the threshold function being but the binomial set, {0, 1}. The artificial neuron can thus be defined as:

$$g(x) = \sigma(\alpha_0 + \alpha_1{}^T x),$$

where the input vectors, $x \in \Re^d$, are weighted by $\alpha_1{}^T \in \Re^d$ and $\alpha_0 \in \Re$. The construction of a network of neurons (i.e. a neural network) begins with the initial n_0 inputs. The final outcome is but one output. In between the initial inputs and final output

are several hidden layers. The graph may be represented by a forward feeding network graph where the output of one layer will become the input of the next layer.

To estimate the unknown coefficients for all neurons, the threshold function is first replaced with a sigmoid function, $\sigma(x):\Re \to [0,1]$ which is defined as a nondecreasing function with $\lim_{x\to-\infty}\sigma(x)=0$ and $\lim_{x\to\infty}\sigma(x)=1$. This is also referred to as a squashing function (Györfi et al., 2002). Some of the more familiar squashing functions are:

logistic squasher: $\sigma(x)=\dfrac{1}{1+e^{-x}}$,

Gaussian squasher: $\sigma(x)=\dfrac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{\left(\frac{-y^2}{2}\right)}dy$ and

arctan squasher: $\sigma(x)=\dfrac{1}{2}+\dfrac{1}{\pi}\arctan(x)$.

To formulate the model, let (X,Y) be the set of training data where $X=(X_1,X_2,\ldots,X_{n_0})$ is the set of input vectors, each having equal length, and Y the corresponding outputs. Further, suppose that there are *l* different classifications for the outputs.

Now consider a neural network with $m+1$ layers. Each layer will be connected with the previous layer. For clarity, define the initial input data as the initial layer, i.e, $X=(X_1,X_2,\ldots,X_{n_0})=\left(x_1(0),\ldots,x_{n_0}(0)\right)$, and the image of *X* at the k^{th} level as: $X(k)=\left(x_1(k),\ldots,x_{n_{k0}}(k)\right)$, for $k=1,\ldots,m$. Thus, for the k^{th} level of the neural network and the i^{th} decision, $i=1,\ldots,l$, the neuron may be written as:

$$X_i(k)=\sigma_k(\alpha_{0k}+\alpha_k{}^T X_i(k-1)).$$

A backpropagation algorithm was introduced by Rumelhart and McClellan (Rumelhart & McClelland, 1986). The performance measure of the L_2 error function: $\sum_{i=1}^{l}\left(y_i - x_i(m)\right)^2$ is minimized by the technique of Lagrange multipliers (Vapnik, 1998). The Lagrange function becomes:

$$L(\alpha, X, \beta) = \sum_{i=1}^{l}\left(y_i - x_i(m)\right)^2 + \sum_{i=1}^{l}\sum_{k=1}^{m}\left(\beta_i(k)\cdot(X_i(k) - \sigma_k(\alpha_{0k} + \alpha_k{}^T X_i(k-1))\right),$$

where $\beta_i(k)$ are the Lagrange multipliers. The process of finding the minimum follows the usual procedure of setting the gradient of *L*, with respect to all parameters, equal to zero. First taking the partial of *L* with respect to the Lagrange multipliers, $\beta_i(k)$, gives an iterative procedure of defining the output vectors within the hidden layers which gives, what is called, a forward dynamic to the problem.

$$\frac{\partial L}{\partial \beta_i(k)} = X_i(k) - \sigma(\alpha_{0k} + \alpha_k{}^T X_i(k-1)) \overset{set}{=} 0 \qquad \text{or}$$

$$X_i(k) = \sigma(\alpha_{0k} + \alpha_k{}^T X_i(k-1)) \qquad \text{for } i = 1, \ldots, l \quad k = 1, \ldots, m,$$

with the initial condition $X_i(0) = X_i$.

Secondly, by taking the partial derivative of *L* with respect to the inputs X_i, the resulting equations give a backward iterative definition to the Lagrange multipliers β_i. For the last layer:

$$\frac{\partial L}{\partial X_i(m)} = -2(y_i - x_i(m)) + \beta_i(m) \overset{set}{=} 0 \quad \text{which implies}$$

$$\beta_i(m) = 2(y_i - x_i(m)) \qquad \text{for} \quad i = 1, \ldots, l$$

and for the hidden layers:

$$\frac{\partial L}{\partial X_i(k)} = \beta_i(k) - {\alpha_{k+1}}^T \nabla \sigma_{k+1}(\alpha_{0(k+1)} + {\alpha_{k+1}}^T X_i(k))\beta_i(k+1) \overset{set}{=} 0 \quad \text{which}$$

implies,

$$\beta_i(k) = {\alpha_{k+1}}^T \nabla \sigma_{k+1}(\alpha_{0(k+1)} + {\alpha_{k+1}}^T X_i(k))\beta_i(k+1)$$

for $i = 1, \ldots, l \quad k = 1, \ldots, m-1$.

Finally, the partial of L with respect to the weights α_i is taken. For clarity, define the input vectors $X_i(k)$ as augmented matrices. Then we can write:

$(\alpha_{0(k+1)} + {\alpha_{k+1}}^T X_i(k)) = \omega_k X_i(k)$, where $X_i(k)$ on the right is an augmented matrix. Thus

$$\frac{\partial L}{\partial \omega_k} = -\sum_{i=1}^{l} \beta_i(k) \nabla \sigma_{k+1}({\omega_k}^T X_i(k-1)) \cdot {X_i}^T(k-1) \overset{set}{=} 0$$

This form, however, does not give a direct way for computing the weights ω_k. The algorithm called the steepest gradient descent, fortunately, may be used to estimate ω_k.

$$\omega_k + \gamma_t \sum_{i=1}^{l} \beta_i(k) \nabla \sigma_{k+1}({\omega_k}^T X_i(k-1)) \cdot {X_i}^T(k-1) \rightarrow \omega_k \ ,$$

where γ_t is defined as a small value used in each iteration *t*.

It should be pointed out that the L_2 error function, $\sum_{i=1}^{l} \left(y_i - x_i(m)\right)^2$, is nonconvex and will have several local minima. Thus, the final solution depends on the choice of starting weights ω_k (Hastie et al., 2009). Typically a number of random starting weights should be selected whereby the lowest result will be chosen as the minimum error. In

addition in each trial, the weights should initially be chosen to be close to zero. Choosing the weights to be close to zero causes the sigmoid function to be roughly linear in the early stages of the algorithm. The model will become increasingly nonlinear as the weights increase as needed in the algorithm.

Another issue is overfitting due to the many weights associated with neural networks. An early stopping point is usually implemented in which a validation set is used to determine the stopping point. Another procedure used to avoid overfitting, called weight decay, places a penalty on to the error function similar to that of ridge regression (Hastie et al., 2009).

By controlling the growth of the number of hidden layers *m* and bounding the Lagrange constant $\sum_{i=1}^{m} |\beta_i| \leq c$, where *c* is a finite constant, the empirical L_2 error function minimization provides universally consistent neural network estimates (Györfi et al., 2002). (The theorem and proof can be found on page 301 in "A Distribution-Free Theory of Nonparametric Regression" by Gyorfi et. al.)

3.4 Decision Trees

One needs to realize that there is not a unique decision tree algorithm. The major differences occur in the decision formula and in the pruning of branches. Some of the more common classification tree algorithms are: CART (Leo Breiman, 1984), PART (Chambers & Hastie, 1993), C4.5 (Quinlan, 1993) which is an extended version of ID3

and the CRAN r packages: “tree”(Ripley, 2013) and “rpart”(Therneau, Atkinson, & Foundation, 2013).

As the name indicates, decision trees consist of nodes and branches. Each node branches out indicating a decision has been made on how to group a new node, the data which best represents the dependent variable. This process continues until a final node, called a leaf, and a conclusion is reached.

In developing a decision tree, three decisions need to be made (Leo Breiman, 1984):

i) how to select the splits,

ii) how to determine if a new node is a terminal node (leaf) or not,

iii) and how to make the assignment of each leaf to a particular class.

For splitting rules, define A to be a node which contains a set of data points. Each data point, has J attributes and a resultant term which corresponds to a particular class C. Define $p_{i,A}$ as the probability of being in class i from set A. Define $I(A)$ as the impurity of set A where $I(A) = \sum_{i \in C} f(p_{i,A})$ for some impurity function $f(\cdot)$.

By choosing an impurity function that has the desirable properties of being concave and having the endpoints $f(0) = f(1) = 0$ the results would, by definition, guarantee that: 1) a “pure set” (a set that is entirely of a single class) would have impurity of zero, i.e. $I(A) = 0$, and 2) by Jensen’s inequality (Royden, 1988; Billingsley, 1995), the impurity reduction would be nonnegative, i.e.

$$\Delta I = p(A)I(A) - p(A_L)I(A_L) - p(A_R)I(A_R) \geq 0$$

where A_L and A_R are sets obtained from a partition of A.

Traditionally, there are two possibilities considered for the impurity function: the information index $f(p) = -p \log p$ and the Gini index $f(p) = p(1-p)$. Graphically, the two functions are similar and it has been observed that "final tree selection are surprisingly insensitive to the choice of splitting rule" due to the impurity function (Leo Breiman, 1984).

The general idea of the split is simple; we want to split the data in such a way as to be able to give an accurate prediction in terms of output (the class) of an independent sample by following the branches of the tree. Thus, at each node a splitting decision is made by using a greedy search method. By considering every possible split of every attribute, the split which produces the greatest "impurity reduction" will be accepted. In other words:

$$\max \Delta I \quad \text{where}$$

$$\Delta I = p(A)I(A) - p(A_L)I(A_L) - p(A_R)I(A_R)$$

Typically, splits divide a group in two branches to produce a binary tree. Numerical attributes are ordered and splits $\leq$ vs. > each of the elements are considered. For categorical attributes, every division resulting in two subgroups are considered. Multivariable splits are avoided since they could fragment the data too quickly with the possibility of having several categories with only one response (Shalizi, 2009). (Note: CRAN R's package rpart uses several different measures of impurity in an attempt to avoid problem such as "ties".)

A completed decision tree can be quite large and complex. Decisions must now be made on how to prune the tree. Stopping the splitting process at a node will make the model more predictable and it is expected to lower the predicted error rate (Kantardzic,

2003). Early stopping rules included stopping if the size of the node was less than a pre-assigned value and/or if the maximum decrease in the impurity was less than a given value, i.e. $\max \Delta I \leq \beta$ (Breiman, 1984). However, both of these criteria tend to give unsatisfactory results. Other types of prepruning stopping criterion are based on some statistical test such as F or χ^2 test. If there is no significant difference in accuracy before the split verses after the split, then the node is a terminal node.

Postpruning first runs a decision tree to completion followed by the removal of the tree structure. The package rpart chooses the minimum cost of every sub tree. The cost for any sub tree T_i of tree *T* is defined as:

$$R_\alpha(T_i) = R(T_i) + \alpha \cdot \left|T_i\right| ,$$

where

$R(T_i) = \sum_{\forall subtree of\ T_i} p(T_j) R(T_j)$ is the risk of T_i,

$\left|T_i\right|$ is the number of nodes of T_i, and

$\alpha \in [0, \infty)$ is the "cost" of adding another number.

Cross validation is used to choose the best value of α. A more complete explanation may be found in the documentation for the rpart package (Therneau et al., 2013).

3.5 Bagging

Bagging or bootstrap aggregating "is a method for generation of multiple versions of a predictor and using these to get an aggregated predictor" as described by Leo Breiman, the developer of the method (Breiman, 1994). It is a method which improves the

accuracy of an estimate or prediction by allowing several estimates to "vote" on the prediction.

Define L to be a training set which contains a sample of independent elements (*x*, *y*) drawn from the distribution *P*. Define $f(x, L)$ to be the predictor function of *x* based on the sample set *L*. Define $f_A(x, P) = E_L f(x, L)$ to be an aggregate of predictors.

Let *X, Y* be random variables from the distribution *P* and independent of *L*. The average prediction error *e* in $f(x, L)$ is:

$$e = E_L E_{X,Y}\left(Y - f(X, L)\right)^2 .$$

Similarly the prediction error for the aggregate is:

$$e_A = E_{X,Y}\left(Y - f_A(X, P)\right)^2$$

Using the identity $E(z^2) \geq (Ez)^2$ for any random variable *z*:

$$\begin{aligned} e &= E_L E_{X,Y}\left(Y - f(X, L)\right)^2 \\ &= E_L E_{X,Y}\left(Y^2 - 2Yf(X, L) + \left(f(X, L)\right)^2\right) \\ &\geq \left(E_{X,Y} Y\right)^2 - 2E_{X,Y} Y f_A(X, P) + \left(E_{X,Y} f_A(X, L)\right)^2 \\ &= E_{X,Y}\left(Y - f_A(X, P)\right)^2 \\ &= e_A \end{aligned}$$

Hence, the aggregate predictor produces a lower error than an individual predictor. This improvement depends on the difference in the identity, $E(z)^2 \geq (Ez)^2$ i.e. and thus how unequal $E_L f(X, L)^2 \geq \left(E_L f(X, L)\right)^2$ are. The higher the variability is to the replicate of *L*, the more improvement the aggregate will produce. Further notice that

the bagging aggregate is not $f_A(X,P)$ which is based on the entire distribution *P*, but rather a bootstrap approximation of P, P_L. If the procedure is stable, then the aggregate is close to a predictive value, $f_A(X,P_L) \cong f(X,L)$, and bagging will have little to no use.

A more intuitive "proof" is achieved by assuming that the data $(X_1,Y_1),(X_2,Y_2),\ldots,(X_n,Y_n)$ is iid, and that the aggregate is the simple mean of the predicted values, $Z = \frac{1}{m}\sum_{i=1}^{m}\hat{Y}_i$, where $f(X_i) = \hat{Y}_i$ is the estimator for X_i based on the training data *L*. *Z* is as unbiased estimator of *Y*,

$$EZ = E\left(\frac{1}{m}\sum_{i=1}^{m}\hat{Y}_i\right) = \frac{1}{m}\sum_{i=1}^{m}E\hat{Y}_i = \frac{1}{m}\sum_{i=1}^{m}Y = Y\ .$$

If $\sigma^2 = E\left((2-EZ)^2\right)$ exist, the expected loss function for *Z* is:

$$\begin{aligned} E(Z-y)^2 &= E\left((Z-EZ)^2\right) \\ &= \sigma^2(Z) \\ &= \frac{1}{m^2}\sum_{i=1}^{m}\sigma^2(Y_i) \\ &= \frac{1}{m}\sigma^2(Y) \end{aligned}$$

Thus, as $m \to \infty, \quad E(Z-y)^2 \to 0$, i.e. by increasing the number of predictions, the mean of these predictions comes closer to the true value.

Again this second derivation is intuitive, informal and provides the main idea of bagging. However the previous derivation which was based on Leo Breiman's proof,

provide the understanding of how much of an improvement bagging can produce for unstable situations.

4 THE CROSS-VALIDATION SELECTOR AND VARIABLE BAGGING

The central idea of this section is Dudoit and van der Laan's theorem (2003). The theorem presented in this paper is more specific to the needs of the K-fold cross-validation selector as opposed to the general form presented in the original paper. In any event, a lot of background information is needed for the theorem's understanding. Therefore, the section starts out by stating and proving background lemmas. The lemmas include: Markov's inequality, Chebyshev's inequality, Chernoff's bounding methods, and Bernstein's inequality. All these lemmas may be found in a complete text about nonparametric regression. The proof of the theorem is quite involved, however, it provides an upper bound for the risk of the cross-validation selector. With the use of Hoeffding's inequality, the bound in Theorem 1 can be improved. The statement and proof of this statement can be found in Theorem 2.

The section ends with the introduction of a new procedure called variable bagging. This method was specifically developed to aid those algorithms which have issues with dimensionality. In later sections, both the cross-validation selector and the variable bagging process will be combined to build models with impressive results.

4.1 Introduction

Consider the following scenario. Let P_0 is a specific, but unknown, true probability distribution set. Let $\hat{P}_0$ be the empirical distribution of the random sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from the data generating distribution P_0 where $X_i \in \Re^j$ and $Y_i \in \Re$ is the univariate outcome. The goal is to model P_0 by using the data set $\hat{P}_0$. This is one of the main problems in statistics which can be filled with many pitfalls. In a traditional approach, the practitioner would assume the structure of the underlying distribution of P_0, i.e., the practitioner would assume a parametric model. There are some advantages to this approach. The model usually depends only on a relatively small number of parameters and this model would usually be easy to interpret (Györfi et al., 2002). However, by using the assumption of a parametric model from the beginning, the practitioner is admitting that the model is wrong. Thus, no matter how representative the data set may be, the resulting model is limited to the best model based on the predetermined parametric structure (Györfi et al., 2002). Therefore the model is biased. This bias cannot be improved upon, no matter what the sample size (M. J. van der Laan & Rose, 2011).

One way to correct this problem is to let the model learn from the data which would discover the underlying trends represented by the data $\hat{P}_0$ (M. J. van der Laan & Rose, 2011). A nonparametric statistical model assumes only that the empirical data contains *n* iid (independent, identically distributed) observations from the data generating distribution P_0 which is unknown. The goal of this nonparametric model, also called

machine-learning, is to find an acceptable generalization which represents the underlying distribution by reviewing the data set (Kantardzic, 2003).

Another drawback for both the parametric and the non-parametric model lies in the zeal of the practitioner in finding the "best" model. This idea might seem confusing at first since it is the goal to produce the most accurate model in the prediction of future outcomes. The problem lies in overfitting (Cawley & Talbot, 2010). In parametric terms, overfitting occurs when the random error of the true model is incorporated in the model along with the underlying trend (Everitt, 2002). In machine learning (non-parametric modeling) overfitting occurs when the model memorizes the training data (Kantardzic, 2003) instead of representing the trend. Methods in avoiding overfitting involving a finite data set include: k-fold cross-validation, pruning, early stopping, Bayesian priors on parameters and optimization of performance bounds (Cawley & Talbot, 2010).

The following section develops the nonparametric and semi-parametric theory which will be used in section 4.3 to prove the main theorem of the chapter which in turn is used in developing a model for classifying whether or not a MRI scan shows an Alzheimer's case. The data which will be used will be the LONI data set. The theory not only gives validity for the use of k-fold cross-validation methodology but also extends to the practice of combining several models (both parametric and nonparametric) to come up with one "super learner" model (M. J. van der Laan, Polley, & Hubbard, 2007).

4.2 Background Lemmas

In establishing the theory, it is helpful to start with the basic theorems pertaining to the theory of concentration of measure. Though Markov's inequality, Chebyshev's inequality, Chernoff's exponential bounding method and Bernstein's inequality are easily found in many different sources, the first three were taken from Vincent, T. et al. (Vincent, Tenorio, & Walkin) and the latter from page 594 of Györfi et al.(2002).

Markov's inequality is the basis in the theory of convergence. For any non-negative distribution and $t>0$, it provides an upper bound of the percentage of the tail of the distribution which is above t. Of course if the distribution were known, better estimates are usually available. Also, the inequality relates probability of a distribution to its expectation.

Lemma 1 *(Markov's inequality)*

For any nonnegative random variable X with finite mean and t>0,

$$\Pr(X \geq t) \leq \frac{\mathrm{E}[X]}{t} \quad .$$

Proof: For the continuous case, for the nonnegative random variable X and $t>0$

$$\mathrm{E}[X] = \int_0^\infty xf(x)dx$$

$$= \int_0^t xf(x)dx + \int_t^\infty xf(x)dx$$

$$\geq \int_t^\infty xf(x)dx$$

$$\geq \int_t^\infty tf(x)dx$$

$$= t\int_t^\infty f(x)dx$$

$$= t\ \Pr(X \geq t)$$

■

Chebyshev's inequality extends the ideas of Markov's inequality for variances. Historical note: Markov was the student of Chebyshev and proved the inequality in his dissertation which Chebyshev stated 10 years earlier without stating a proof. (Taylor)

Lemma 2 *(Chebyshev's inequality)*

For random variable X with finite variance σ^2 ,

$$\Pr\left(\left|X - \mathrm{E}[X]\right| \geq t\right) \leq \frac{\sigma^2}{t^2} \qquad \textit{for every } t > 0.$$

Proof:

$$\Pr\left(\left|X - \mathrm{E}[X]\right| \geq t\right) = \Pr\left(\left|X - \mathrm{E}[X]\right|^2 \geq t^2\right)$$

Using Markov's inequality,

$$\Pr\left(\left|X - \mathrm{E}[X]\right|^2 \geq t^2\right) \leq \frac{\mathrm{E}\left(\left|X - \mathrm{E}(X)\right|^2\right)}{t^2}$$

$$= \frac{\sigma^2}{t^2}$$

Notice in Chebysev's inequality, the second moment is used before applying Markov's inequality. Extend the idea by using the moment generating function and relying on the monotonic property of the exponential function before using Markov's inequality.

Lemma 3 *(Chernoff's bounding method)*

For any random variable X and t>0,

$$\Pr(X \geq t) \leq \min_{s>0} \frac{\mathrm{E}[e^{sX}]}{e^{st}} \qquad \textit{if the RHS exist.}$$

Proof:

For any $s > 0$,

$$\Pr(X \geq t) = \Pr(e^{sX} \geq e^{st})$$

using Markov's inequality,

$$\Pr(e^{sX} \geq e^{st}) \leq \frac{\mathrm{E}(e^{sX})}{e^{st}}$$

Since this is true for any s, it follows that

$$\Pr(X \geq t) \leq \min_{s>0} \frac{\mathrm{E}(e^{sX})}{e^{st}} \qquad \text{if the RHS exist.}$$

■

Chernoff's bounding method is crucial in the proof of Bernstein's inequality which in turn is used in van der Laan's theorem showing the convergence of conditional

and optimal risk. The proof follows the proof provided by Gyorfi et al. (2002) on page 594.

Lemma 4 *(Bernstein's inequality)*

Let $X_1, \ldots, X_n$ *be independent real valued random variables. Assume for each* $i = 1, \ldots, n$, $X_i \in [a,b]$ *with probability one, where* $a, b \in \Re$ *with a<b. Define* $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} \text{var}(X_i) > 0$. Then for all $\varepsilon > 0$,

$$P\left\{\left|\frac{1}{n}\sum\left(X_i - E(X_i)\right)\right| > \varepsilon\right\} \leq 2\exp\left(-\frac{n\varepsilon^2}{2\left(\sigma^2 + 2\varepsilon\frac{(b-a)}{3}\right)}\right) .$$

Proof:

Define $Y_i = X_i - E(X_i)$ for i = 1…, n. Then with probability one $|Y_i| \leq b - a$ and $\text{E}(Y_i^2) = \text{var}(X_i)$. Working with the positive portion in the absolute value,

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}\left(X_i - E(X_i)\right) > \varepsilon\right\} = P\left\{\frac{1}{n}\sum_{i=1}^{n} Y_i > \varepsilon\right\} .$$

Then for an arbitrary s > 0,

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}Y_i > \varepsilon\right\} = P\left\{s\sum_{i=1}^{n}Y_i - sn\varepsilon > 0\right\}.$$

Using Chernoff's exponential bounding method,

$$P\left\{s\sum_{i=1}^{n}Y_i - sn\varepsilon > 0\right\} \le E\left\{\exp\left(s\sum_{i=1}^{n}Y_i - sn\varepsilon\right)\right\},$$

and because of the independence of the Y_i's, we can write

$$\begin{aligned}
E\left\{\exp\left(s\sum_{i=1}^{n}Y_i - sn\varepsilon\right)\right\} &= e^{-sn\varepsilon}\prod_{i=1}^{n}E\left(e^{sY_i}\right) \\
&= e^{-sn\varepsilon}\prod_{i=1}^{n}E\left(1+sY_i+\sum_{j=2}^{\infty}\frac{(sY_i)^j}{j!}\right) \\
&\le e^{-sn\varepsilon}\prod_{i=1}^{n}E\left(1+sY_i+\sum_{j=2}^{\infty}\frac{s^jY_i^{\,2}(b-a)^{j-2}}{2\cdot 3^{j-2}}\right) \\
&= e^{-sn\varepsilon}\prod_{i=1}^{n}E\left(1+sY_i+\frac{s^2Y_i^{\,2}}{2}\sum_{j=2}^{\infty}\left(\frac{s(b-a)}{3}\right)^{j-2}\right) \\
&\le e^{-sn\varepsilon}\prod_{i=1}^{n}E\left(1+sY_i+\frac{s^2Y_i^{\,2}}{2}\cdot\frac{1}{1-\frac{s(b-a)}{3}}\right)
\end{aligned}$$

provided that $\left|\frac{s(b-a)}{3}\right| < 1$. Remembering that $E(Y_i) = 0$ and $E(Y_i^{\,2}) = \text{var}(X_i)$, then

$$\begin{aligned}
e^{-sn\varepsilon}\prod_{i=1}^{n}E\left(1+sY_i+\frac{s^2Y_i^{\,2}}{2}\cdot\frac{1}{1-\frac{s(b-a)}{3}}\right) &= e^{-sn\varepsilon}\prod_{i=1}^{n}\left(1+\frac{s^2\,\text{var}(X_i)}{2}\cdot\frac{1}{1-\frac{s(b-a)}{3}}\right) \\
&\le e^{-sn\varepsilon}\prod_{i=1}^{n}\exp\left(\frac{s^2\,\text{var}(X_i)}{2}\cdot\frac{1}{1-\frac{s(b-a)}{3}}\right) \\
&= \exp\left(-sn\varepsilon+\frac{s^2n\sigma^2}{2\left(1-\frac{s(b-a)}{3}\right)}\right)
\end{aligned}$$

From the first two terms of the Taylor expansion of e^x, i.e. $1+x < e^x$ for any x.

Now, set $s = \frac{\varepsilon}{\frac{\varepsilon(b-a)}{3} + \sigma^2}$, which satisfies the condition $\left|\frac{s(b-a)}{3}\right| < 1$, then

$$\exp\left(-sn\varepsilon + \frac{s^2 n\sigma^2}{2\left(1-\frac{s(b-a)}{3}\right)}\right) = \exp\left(\frac{-n\varepsilon^2}{\varepsilon\frac{(b-a)}{3}+\sigma^2} + \frac{\varepsilon^2}{(\varepsilon\frac{(b-a)}{3}+\sigma^2)^2}\cdot\frac{n\sigma^2}{2\left(1-\frac{\varepsilon(b-a)}{3(\varepsilon\frac{(b-a)}{3}+\sigma^2)}\right)}\right)$$

$$= \exp\left(\frac{-n\varepsilon^2}{\varepsilon\frac{(b-a)}{3}+\sigma^2} + \frac{\varepsilon^2}{(\varepsilon\frac{(b-a)}{3}+\sigma^2)}\cdot\frac{n\sigma^2}{2\left(\varepsilon\frac{(b-a)}{3}+\sigma^2-\varepsilon\frac{(b-a)}{3}\right)}\right)$$

$$= \exp\left(\frac{-n\varepsilon^2}{2\varepsilon\frac{(b-a)}{3}+2\sigma^2}\right)$$

Working with the negative portion in the absolute value yields the same result which proves the inequality.

■

The following lemma and proof follow closely Lemma 2 on page 15 of Dudoit and van der Laan paper 126.

Lemma 5 *(Convergence in probability)*

Let $X_1, X_2 \ldots$ *be a sequence of random variables with finite expected value* $\mathrm{E}(|X_n|) = O(g(n))$ *where g(n) is a positive function. Then* $X_n = O_P(g(n))$.

Proof:

Pick any number $\varepsilon > 0$ and let $\mathrm{E}(|X_n|) = O(g(n))$. Then there exist $N>0$ and $C > 0$, such that $\frac{\mathrm{E}(|X_n|)}{g(n)} < C$ for every $n>N$. Define $B = \frac{C}{\varepsilon}$ and consider $P\left(\frac{|X_n|}{g(n)} > B\right)$. Using Markov's inequality, then for every $n \geq N$, $P\left(\frac{|X_n|}{g(n)} > B\right) \leq \frac{\mathrm{E}(X_n)}{B \cdot g(n)} \leq \frac{C}{B} = \varepsilon$. Hence, $X_n = O_P(g(n))$.

■

In 1963 Wassily Hoeffding provided his own upper bound on the probability of the sum of the difference between random variables and their respected expected values. Hoeffding's inequality is a more general case of Bernstein's inequality and improves of the bound for values in the tails of the distribution as we will see later. The proof below of Hoeffding's inequality is based on the following papers: Hoeffding (1963), Györfi (2002), and Nowak (2007).

Lemma 6 *(Hoeffding's inequality)*

Let $X_1, \ldots, X_n$ be independent real valued random variables. Assume for each $i = 1, \ldots, n$, $X_i \in [a_i, b_i]$ with probability one, where $a_i, b_i \in \Re$ with $a_i < b_i$. Then for all $\varepsilon > 0$,

$$P\left\{\left|\frac{1}{n}\sum\left(X_i - E(X_i)\right)\right| > \varepsilon\right\} \leq 2\exp\left(-\frac{2n\varepsilon^2}{\frac{1}{n}\sum_{i=1}^{n}(b_i - a_i)^2}\right) .$$

Proof:

Define $Y_i = X_i - E(X_i)$ for $i = 1\ldots, n$. Then $E(Y_i) = 0$ and with probability one, $Y_i \in [a_i - E(X_i), b_i - E(X_i)]$. Working with the positive portion in the absolute value,

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - E(X_i)) > \varepsilon\right\} = P\left\{\frac{1}{n}\sum_{i=1}^{n} Y_i > \varepsilon\right\} \quad .$$

The proof begins by following the proof of the Bernstein's inequality. By using an arbitrary s > 0 and then using Chernoff's exponential bounding method we have,

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - E(X_i)) > \varepsilon\right\} \le e^{-sn\varepsilon}\prod_{i=1}^{n}\mathrm{E}\left(e^{sY_i}\right) \quad .$$

Fix an $i \in \{1,\ldots,n\}$, we then need to find an upper bound for $E(e^{sY_i})$. First, note that e^{sx} is a convex function which implies $e^{sx} \le \frac{x-a}{b-a}e^{sb} + \frac{b-x}{b-a}e^{sa}$. Therefore we can write:

$$E(e^{sY_i}) \le \frac{E(Y_i) - a_i}{b_i - a_i}e^{sb_i} + \frac{b_i - E(Y_i)}{b_i - a_i}e^{sa_i}$$

$$= \frac{b_i}{b_i - a_i}e^{sa_i} - \frac{a_i}{b_i - a_i}e^{sb_i} \qquad \text{, since } E(Y_i) = 0$$

$$= e^{sa_i}\left(1 + \frac{a_i}{b_i - a_i} - \frac{a_i}{b_i - a_i}e^{s(b_i - a_i)}\right)$$

$$= e^{-ps(b_i - a_i)}\left(1 - p + pe^{s(b_i - a_i)}\right) \qquad \text{, where } p = \frac{-a_i}{b_i - a_i} \quad .$$

Define $u = s(b_i - a_i)$, then

$$= e^{-pu}\left(1 - p + pe^{u}\right)$$

$$= e^{-pu}e^{\ln\left(1-p+pe^{u}\right)}$$

$$= e^{\phi(u)} \qquad \text{, where } \phi(u) = -pu + \ln\left(1 - p + pe^{u}\right) .$$

Using the first three terms of the Taylor series, approximate $\phi(u)$. We have

$$\phi(u) = \phi(0) + u \cdot \phi'(u) + \frac{u^2}{2}\phi''(\nu) \qquad \text{for some } \nu \in [0, u]$$

where:

$$\phi(0) = -p \cdot 0 + \ln\left(1 - p + pe^{0}\right) = 0,$$

$$\phi'(0) = -p + \frac{pe^{0}}{1 - p + pe^{0}} = 0, \text{ and}$$

$$\phi''(\nu) = \frac{pe^{\nu}}{1 - p + pe^{\nu}} - \frac{\left(pe^{\nu}\right)^2}{\left(1 - p + pe^{\nu}\right)^2}$$

$$= \rho - \rho^2 \qquad \text{where } \rho = \frac{pe^{\nu}}{1 - p + pe^{\nu}} .$$

Notice that $\frac{d}{d\rho}(\phi'') = 1 - 2\rho \overset{set}{=} 0$ i.e. $\rho = \frac{1}{2}$ is a critical point. Also, $\frac{d^2}{(d\rho)^2}(\phi'') = -2$,

implying that $\phi''(\rho = \frac{1}{2}) = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$ is a maximum. Hence,

$$\phi(u) = \phi(0) + u \cdot \phi'(u) + \frac{u^2}{2}\phi''(v) \qquad \text{or}$$

$$\leq \frac{u^2}{8}$$

$$= \frac{s^2(b_i - a_i)^2}{8}$$

and

$$E(e^{sY_i}) \leq e^{\frac{s^2(b_i-a_i)^2}{8}} \quad .$$

Now consider each $i \in \{1,\ldots,n\}$, from the main expression we can now write:

$$P\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - E(X_i)) > \varepsilon\right\} \leq e^{-sn\varepsilon}\prod_{i=1}^{n}\mathrm{E}\left(e^{sY_i}\right)$$

$$\leq e^{-sn\varepsilon}\prod_{i=1}^{n}e^{\frac{s^2(b_i-a_i)^2}{8}}$$

$$= e^{\left(-sn\varepsilon + \frac{s^2\sum_{i=1}^{n}(b_i-a_i)^2}{8}\right)} \quad .$$

This is true for all *s*, therefore minimize this upper bound by minimizing the exponent:

$$\lambda(s) = -sn\varepsilon + \frac{s^2 \sum_{i=1}^{n} (b_i - a_i)^2}{8} \ .$$

We have $\lambda'(s) = -n\varepsilon + \frac{s \sum_{i=1}^{n} (b_i - a_i)^2}{4} \underset{set}{=} 0$ which implies,

$$s = \frac{4n\varepsilon}{\sum_{i=1}^{n} (b_i - a_i)^2}$$

(Note that: $\lambda''(s) = \frac{\sum_{i=1}^{n} (b_i - a_i)^2}{4} > 0$, so a minimum exist for the above critical point.)

Substituting back into the equation:

$$\lambda(s) = \frac{-4n^2\varepsilon^2}{\sum_{i=1}^{n} (b_i - a_i)^2} + \frac{16n^2\varepsilon^2 \sum_{i=1}^{n} (b_i - a_i)^2}{\left(\sum_{i=1}^{n} (b_i - a_i)^2 \right)^2 \cdot 8}$$

$$= \frac{-2n^2\varepsilon^2}{\sum_{i=1}^{n} (b_i - a_i)^2}$$

and

$$P\left\{ \frac{1}{n} \sum_{i=1}^{n} \left(X_i - E(X_i) \right) > \varepsilon \right\} \le \exp\left(\frac{-2n\varepsilon^2}{\frac{1}{n} \sum (b_i - a_i)^2} \right) \ .$$

Working with the negative portion in the absolute value yields the same result which proves the inequality.

■

4.3 The Cross-Validation Selector

We are now at a point where the two main theorems of the Super Learner algorithm are stated and proved. The first places a bound on the cross-validation risk. The method of cross-validation studied by van der Laan and Dudoit (2003) is general and includes V-fold, Monte Carlo, Bootstrap cross-validation. Because of its low bias but high variance estimators, leave-one-out cross-validation is excluded from the studies since, among other things, it has been shown to perform poorly compared to the other forms of cross-validation (Breiman & Spector, 1992; Breiman, 1996).

Often as is the case in the field of data mining, many types of models (rules) are run on a data set. The model, or rule, which performs the "best", is selected as the model for the data set. Why not combine all the models together to form one grand model? Van der Laan et al. (2007) does just that as they combine several models together, as long as the number is polynomial in sample size, into one super learner. It is shown in Theorem 2 that this super learner will perform, on average, at least as well of any of the individual models used, at least in the asymptotic sense.

4.3.1 Notation

The notation can become quite convoluted due to the use of various subsets of the data distribution set involved. Consider the following data distribution sets: Define $\mathcal{P}$ to be the set of possible probability distributions for data (*X*,*Y*). Let $P \in \mathcal{P}$ (note the absence of a subscript) be a general theoretical distribution, and P_0 be a specific, usually unknown, true probability distribution. Define $\hat{P}_0$ to be the empirical distribution of the random sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from the data generating distribution P_0 where $X_i \in \Re^p$, *p* is the number of voxels, and $Y_i \in \Re$ is the univariate outcome. During K-fold cross validation, $\hat{P}_0$ is partitioned into *K* separate subsets which will be used as validation sets. Let $\hat{P}_k$, where $k = 1, \ldots, K$, denote the empirical distribution based on each of the *K* validation sets while $\hat{P}_{(-k)} = \hat{P}_0 / \hat{P}_k$, the complement of $\hat{P}_k$, $k = 1, \ldots, K$, are the empirical distributions for each of the respective training sets. Let n_k define the size of each validation set $\hat{P}_k$, which is approximately the same for all validation sets, i.e. $n_i \approx n_j$ where $i \neq j$ and $i, j \in \{1, \ldots, K\}$.

In similar fashion, let $\mathcal{F}$ be the set of all rules (models) which maps the distribution sets in $\mathcal{P}$ to a real value. Define $f \in \mathcal{F}$ to be a general theoretical rule. Let f_0 be the true, usually unknown, rule for the probability distribution P_0.

Define a loss function *L*(*X*,*Y*,*f*) which calculates a measure for the difference between the true outcome *Y* and the estimated or expected outcome $f(X)$. One of the more common loss function is the square loss function: $L(X, Y, f) = (Y - f(X))^2$.

In considering the *J* different rules which are fitted to the *K* different training sets, $\hat{P}_{(-k)}$, define the double subscript for the modeling rule $\hat{f}_{j,(-k)}$ as the modeling rule using the j^{th} algorithm based on the (-*k*) training set. We now need a way to determine the "best" preforming model. One way is to find the smallest average loss using the validation sets, $\hat{P}_k$, for each of the modeled rules, $\hat{f}_{j,(-k)}$. This "best" fitted training model with respect to the validation sets is represented by $\hat{f}_{\hat{j}}$, where

$$\hat{j} = \underset{j\in\{1,\ldots,J\}}{\arg\min} \sum_{k=1}^{K} \frac{n_k}{n} \int L(x, y, \hat{f}_{j,(-k)}) d\hat{P}_k(x, y) \ .$$

For the finite empirical distribution:

$$\hat{j} = \underset{j\in\{1,\ldots,J\}}{\arg\min} \sum_{k=1}^{K} \sum_{1=1}^{n_k} L(x, y, \hat{f}_{j,(-k)}) \ .$$

Extend the idea further by describing the "best" model of all the *J* different fitted models, $\hat{f}_{j,(-k)}$, which are again based on the *K* separate training sets $\hat{P}_{-k}$. However, use the entire distribution set P_0 when calculating the loss. Represent this model by $\hat{f}_{\tilde{j}}$, where

$$\tilde{j} = \underset{j\in\{1,\ldots,J\}}{\arg\min} \sum_{k=1}^{K} \frac{n_k}{n} \int L(x, y, \hat{f}_{j,(-k)}) dP_0(x, y) \ .$$

Finally, we will define the following for the three different quantities of risk. The optimal risk is the accumulation of loss using the best choice from all the rules in the universal set of estimator mappings $\mathcal{F}$. The data generating distribution is P_0. Note that

the optimal risk R_0 is not an estimate since it depends on the definite but unknown distributions P_0 and the true model $f_0 \in \mathcal{F}$. Therefore, define the optimal risk as:

$$R_0 \equiv \min_{f \in \mathcal{F}} \int L(x, y, f) dP_0(x, y) .$$

The conditional risk:

$$\hat{R}_j \equiv \sum_{k=1}^{K} \frac{n_k}{n} \int L(x, y, \hat{f}_{j,(-k)}) \, dP_0(x, y),$$

deals with the cross validation selector. It is the risk associated with the estimated model found by using a training data set in the cross validation procedure.

The conditional risk for the optimal selector:

$$\hat{R}_{\tilde{j}} \equiv \min_{j \in \{1,\ldots,J\}} \sum_{k=1}^{K} \frac{n_k}{n} \int L(x, y, \hat{f}_{j,(-k)}) \, dP_0(x, y) ,$$

is the best of all the conditional risks in the cross validation procedure.

4.3.2 The Convergence of Conditional and Optimal Risks

The proof of the following theorem is similar to the proof provided in the paper: "Asymptotics of Cross-Validated Risk Estimation in Estimator Selection and Performance Assessment" (Dudoit & van der Laan, 2003).

Theorem 1 Assume that $E[Y \mid X] = f_0(X)$, $|Y| \le M < \infty$ a.s., and $\max_{f \in \mathcal{F}} |f(X)| \le M < \infty$ a.s. where *M* is a constant. Then for any $\delta > 0$ and using the quadratic loss function,

$$0 \quad \le E\left[\hat{R}_{\hat{j}} - R_0\right] \le \quad (1+2\delta)E\left[\hat{R}_{\tilde{j}} - R_0\right] + 2\frac{C\,K\,(1+\ln\,J)}{n}$$

where $C = 2(1+\delta)^2\left(\frac{8M^2}{3} + \frac{16M^2}{\delta}\right)$.

Proof: Consider the difference between conditional risk and the optimal risk.

$$0 \le \hat{R}_{\hat{j}} - R_0$$

$$= \sum_{k=1}^{K} \frac{n_k}{n}\left[\int \left(L(x,y,\hat{f}_{\hat{j},(-k)}) - L(x,y,f_0)\right) dP_0(x,y)\right]$$
$$- (1+\delta)\sum_{k=1}^{K} \frac{n_k}{n}\left[\int \left(L\left(x,y,\hat{f}_{\hat{j},(-k)}\right) - L\left(x,y,f_0\right)\right) d\hat{P}_k(x,y)\right]$$
$$+ (1+\delta)\sum_{k=1}^{K} \frac{n_k}{n}\left[\int \left(L\left(x,y,\hat{f}_{\hat{j},(-k)}\right) - L\left(x,y,f_0\right)\right) d\hat{P}_k(x,y)\right]$$

$$\leq \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L(x, y, \hat{f}_{\hat{j},(-k)}) - L(x, y, f_0) \right) dP_0(x, y) \right]$$
$$- (1+\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\hat{j},(-k)}\right) - L(x, y, f_0) \right) d\hat{P}_k(x, y) \right]$$
$$+ (1+\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\tilde{j},(-k)}\right) - L(x, y, f_0) \right) d\hat{P}_k(x, y) \right]$$

$$= \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L(x, y, \hat{f}_{\hat{j},(-k)}) - L(x, y, f_0) \right) dP_0(x, y) \right]$$
$$- (1+\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\hat{j},(-k)}\right) - L(x, y, f_0) \right) d\hat{P}_k(x, y) \right]$$
$$+ (1+\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\tilde{j},(-k)}\right) - L(x, y, f_0) \right) d\hat{P}_k(x, y) \right]$$
$$- (1+2\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\tilde{j},(-k)}\right) - L(x, y, f_0) \right) dP_0(x, y) \right]$$
$$+ (1+2\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\tilde{j},(-k)}\right) - L(x, y, f_0) \right) dP_0(x, y) \right]$$

For simplicity define:

$$S_{\hat{j}} = \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L(x, y, \hat{f}_{\hat{j},(-k)}) - L(x, y, f_0) \right) dP_0(x, y) \right]$$
$$- (1+\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\hat{j},(-k)}\right) - L(x, y, f_0) \right) d\hat{P}_k(x, y) \right]$$
$$= \sum_{k=1}^{K} \frac{n_k}{n} S_{\hat{j},k}$$

$$T_{\tilde{j}} = (1+\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\tilde{j},(-k)}\right) - L(x, y, f_0) \right) d\hat{P}_k(x, y) \right]$$
$$- (1+2\delta) \sum_{k=1}^{K} \frac{n_k}{n} \left[\int \left(L\left(x, y, \hat{f}_{\tilde{j},(-k)}\right) - L(x, y, f_0) \right) dP_0(x, y) \right]$$
$$= \sum_{k=1}^{K} \frac{n_k}{n} T_{\tilde{j},k}$$

Then the inequality can be written as:

$$0 \le \hat{R}_{\hat{j}} - R_0 \le (1+2\delta)\left(\hat{R}_{\tilde{j}} - R_0\right) + \left[S_{\hat{j}} + T_{\tilde{j}}\right]$$

We now need only to show that $[S_{\hat{j}} + T_{\tilde{j}}] \le 4(1+\delta)^2\left(\frac{8M^2}{3} + \frac{16M^2}{\delta}\right) \cdot K \cdot \frac{1+\ln(J)}{n}$.

Using the quadratic loss function, $L(X,Y,f) = (Y - f(X))^2$, for any $(x,y) \in \hat{P}_{(-k)}$, let us make a small study of the difference of the two loss functions defined in $S_{\hat{j}}$ and $T_{\tilde{j}}$.

$$\begin{aligned} L\left(X,Y,\hat{f}_{j,(-k)}\right) - L(X,Y,f_0) &= \left(Y - \hat{f}_{j,(-k)}(X)\right)^2 - (Y - f_0(X))^2 \\ &= \left(Y^2 - 2Y\,\hat{f}_{j,(-k)}(X) + \hat{f}^2_{j,(-k)}(X)\right) - \left(Y^2 - 2Y\,f_0(X) + f_0^2(X)\right) \\ &= 2Y\left(f_0(X) - \hat{f}_{j,(-k)}(X)\right) - \left(f_0^2(X) - \hat{f}^2_{j,(-k)}(X)\right) \\ &= \left(f_0(X) - \hat{f}_{j,(-k)}(X)\right)\left(2Y - f_0(X) - \hat{f}_{j,(-k)}(X)\right). \end{aligned}$$

Since $|Y| \le M < \infty$ a.s. and $\max_{f \in F} |f(X)| \le M < \infty$ a.s., for some constant *M*, we can make the following observations:

$$\left|2Y - f_0(X) - \hat{f}_{j,(-k)}(X)\right| \le 4M \quad \text{and} \quad \left|f_0(X) - \hat{f}_{j,(-k)}(X)\right| \le 2M .$$

The expected value of this difference is,

$$\begin{aligned} E\left[L(X,Y,\hat{f}_{j,(-k)}) - L(X,Y,f_0)\right] &= E\left[E\left(L(x,y,\hat{f}_{j,(-k)}) - L(x,y,f_0)\right)\right] \\ &= E\left[\left(f_0(X) - \hat{f}_{j,(-k)}(X)\right)\left(2E(Y \mid X) - f_0(X) - \hat{f}_{j,(-k)}(X)\right)\right] \\ &= E\left[\left(f_0(X) - \hat{f}_{j,(-k)}(X)\right)^2\right] \end{aligned}$$

which implies:

$$E\left[L\left(x, y, \hat{f}_{j,(-k)}\right) - L(x, y, f_0)\right] \le E[(2M \cdot 4M)]$$

or simply,

$$L\left(x, y, \hat{f}_{j,(-k)}\right) - L(x, y, f_0) \le 8M^2$$

Finally, calculate the variance of this difference:

$$Var\left[L\left(X,Y, \hat{f}_{j,(-k)}\right) - L(X,Y, f_0)\right] \le E\left[\left(L(X,Y, \hat{f}_{j,(-k)}) - L(X,Y, f_0)\right)^2\right]$$

$$= E\left[\left(f_0(X) - \hat{f}_{j,(-k)}(X)\right)^2 \left(2E(Y \mid X) - f_0(X) - \hat{f}_{j,(-k)}(X)\right)^2\right]$$

$$\le (4M)^2 \cdot E\left[L(X,Y, \hat{f}_{j,(-k)}) - L(X,Y, f_0)\right]$$

By defining the following, the expressions involving the expected difference of the loss functions can be written as:

$$\hat{H}_j = \hat{H}_{j,k} = \int \left(L(x, y, \hat{f}_{j,(-k)}) - L(x, y, f_0)\right) d\hat{P}_k(x, y)$$

and

$$\tilde{H}_j = \tilde{H}_{j,k} = \int \left(L(x, y, \hat{f}_{j,(-k)}) - L(x, y, f_0)\right) dP_0(x, y)$$

Then

$$S_{\hat{j},k} = (1+\delta)(\tilde{H}_{\hat{j}} - \hat{H}_{\hat{j}}) - \delta \tilde{H}_{\hat{j}} \text{ and}$$

$$T_{\tilde{j},k} = (1+\delta)(\hat{H}_{\tilde{j}} - \tilde{H}_{\tilde{j}}) - \delta \tilde{H}_{\tilde{j}}$$

Now consider $S_{\hat{j},k} = (1+\delta)(\tilde{H}_{\hat{j}} - \hat{H}_{\hat{j}}) - \delta \tilde{H}_{\hat{j}}$. For any $s > 0$

$$\Pr\left(S_{\hat{j},k} > s \mid \hat{P}_{-k}\right) = \Pr\left(\tilde{H}_{\hat{j}} - \hat{H}_{\hat{j}} > \frac{s + \delta \tilde{H}_{\hat{j}}}{1+\delta} \mid \hat{P}_{-k}\right)$$

$$\leq J \cdot \max_{j \in \{1,\ldots,J\}} \Pr\left(\tilde{H}_{\hat{j}} - \hat{H}_{\hat{j}} > \frac{s + \delta \tilde{H}_{\hat{j}}}{1+\delta} \mid \hat{P}_{-k}\right)$$

Using the previous relation between the variance and expected value, the right side of the above equation and applying the Bernstein's inequality:

$$\max_{j \in \{1,\ldots,J\}} \Pr\left(\tilde{H}_{\hat{j}} - \hat{H}_{\hat{j}} > \frac{s + \delta \sigma_k^2 / 16M^2}{1+\delta} \mid \hat{P}_{-k}\right)$$

$$\leq J \cdot \exp\left(-\frac{n_k}{2(1+\delta)^2} \cdot \frac{\left(s + \delta \frac{\sigma_k^2}{16M^2}\right)^2}{\sigma_k^2 + \frac{8M^2}{3(1+\delta)}\left(s + \frac{\delta \sigma_k^2}{16M^2}\right)}\right)$$

$$= J \cdot \exp\left(-\frac{n_k}{2(1+\delta)^2} \cdot \frac{\left(s + \delta \frac{\sigma_k^2}{16M^2}\right)}{\frac{\sigma_k^2}{\left(s + \frac{\delta \sigma_k^2}{16M^2}\right)} + \frac{8M^2}{3(1+\delta)}}\right)$$

$$\leq J \cdot \exp\left(-\frac{n_k}{2(1+\delta)^2} \cdot \frac{s}{\frac{16M^2}{\delta} + \frac{8M^2}{3}}\right)$$

$$= J \cdot \exp\left(-\frac{n_k}{C} \cdot s\right) \qquad \text{where } C = 2(1+\delta)^2\left(\frac{16M^2}{\delta} + \frac{8M^2}{3}\right)$$

(Sub-lemma) Claim for any $u \in \Re$, $E(X) \le u + \int_u^\infty \Pr(X \ge x)dx$.

Proof:

$$\begin{aligned} E[X] &= \int_{-\infty}^{\infty} x\, f(x)dx \\ &= u\int_{-\infty}^{\infty} f(x)dx + \int_{-\infty}^{\infty} (x-u) f(x)dx \\ &= u + \int_{-\infty}^{u} (x-u) f(x)dx + \int_{u}^{\infty} (x-u) f(x)dx \\ &\le u + \int_{u}^{\infty} (x-u) f(x)dx \\ &= u + \int_{u}^{\infty} \Pr(X \ge x)dx \end{aligned}$$

So for any $u > 0$,

$$E[S_{\hat{j},k}] \le u + \int_u^\infty \Pr(S_{\hat{j},k} \ge s)ds$$

$$\le u + J\int_u^\infty \exp\left(-\frac{n_k}{C}\cdot s\right)ds$$

$$= u + J\cdot\frac{-C}{n_k}\exp\left(-\frac{n_k}{C}\cdot s\right)\Big|_u^\infty$$

$$= u + \frac{J\cdot C}{n_k}\exp\left(-\frac{n_k}{C}\cdot u\right).$$

Minimize this expression by using the first derivative, w.r.t. u, we get

$$\frac{d}{du}\left(u + \frac{J\cdot C}{n_k}\exp\left(-\frac{n_k}{C}\cdot u\right)\right) = 1 - J\exp\left(-\frac{n_k}{C}\cdot u\right) \overset{set}{=} 0$$

$$u = \frac{C\ln(J)}{n_k}$$

Note: $\frac{d^2}{du^2}\left(u+\frac{J\cdot C}{n_k}\exp\left(-\frac{n_k}{C}\cdot u\right)\right)=\frac{J\cdot n_k}{C}\exp\left(-\frac{n_k}{C}\cdot u\right)>0$ so a minimum is achieved.

Thus, we have

$$E\left[S_{\hat{j},k}\right]\le \frac{C\ln(J)}{n_k}+\frac{J\cdot C}{n_k}\exp(-\ln J))$$

$$=C\cdot\frac{(1+\ln(J))}{n_k}\ .$$

So it follows that

$$E\left[S_{\hat{j}}\right]\le\sum_{k=1}^{K}\frac{n_k}{n}E[S_{\hat{j},k}]=\frac{C\,K\,(1+\ln J)}{n}\ .$$

The same argument is applied to produce $E\left[T_{\tilde{j}}\right]\le\frac{C\,K(1+\ln J)}{n}$.

Therefore, the result follows:

$$0\ \le E\left[\hat{R}_{\hat{j}}-R_0\right]\le\ (1+2\delta)E\left[\hat{R}_{\tilde{j}}-R_0\right]+2\frac{C\,K\,(1+\ln J)}{n}\ .$$

■

Corollary 1 Using the results from Theorem 1, if

$$\frac{\ln J}{n\cdot \mathrm{E}\left[\hat{R}_{\tilde{j}}-R_0\right]}\to 0\ \text{ as }\ n\to\infty\ \text{, then }\ \frac{\mathrm{E}\left[\hat{R}_{\hat{j}}-R_0\right]}{\mathrm{E}\left[\hat{R}_{\tilde{j}}-R_0\right]}\to 1\ .$$

Similarly, if

$$\frac{\ln J}{n\cdot\left[\hat{R}_{\tilde{j}}-R_0\right]}\xrightarrow{P}0\ \text{ as }\ n\to\infty\ \text{, then }\ \frac{\hat{R}_{\hat{j}}-R_0}{\hat{R}_{\tilde{j}}-R_0}\xrightarrow{P}1\ .$$

The proof follows immediately from Lemma 5.

The results of Theorem 1 can be improved. By using Hoeffding's inequality, the upper bound for $E[S_{\hat{j}}]$ and $E[T_{\tilde{j}}]$ can be improved upon in the upper "tails" of *s*. Theorem 2 combines the better portions of both Bernstein's inequality and Hoeffding's inequality to produce an improved bound.

Theorem 2: The bounds for $E[S_{\hat{j},k}]$ and $E[T_{\hat{j},k}]$ in Theorem 1 can be improved:

$$\begin{cases} \sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}} + J\sqrt{\frac{\pi\tilde{h}}{\tilde{n}}}\left(1-\Phi(\sqrt{2\ln J})\right) & \text{if} \quad \tilde{n} \le \frac{\tilde{b}^2 \ln J}{\tilde{h}} \\ \frac{\tilde{b}(1+\ln J)}{\tilde{n}} + J\left\{\frac{-\tilde{b}}{\tilde{n}}\cdot e^{\frac{-\tilde{n}\tilde{h}}{\tilde{b}^2}} + \sqrt{\frac{2\tilde{h}}{\tilde{n}}}\cdot\left(1-\Phi\left(\frac{\sqrt{2\tilde{n}\tilde{h}}}{\tilde{b}}\right)\right)\right\} & \text{if} \quad \tilde{n} > \frac{\tilde{b}^2 \ln J}{\tilde{h}} \end{cases}$$

where:

$$\tilde{h} = \tfrac{1}{2}(1+\delta)^2 M^2$$
$$\tilde{b} = 2(1+\delta)^2\left(\frac{8M^2}{3} + \frac{16M^2}{\delta}\right)$$
$$s > 0, \quad \delta > 0, \quad \tilde{n} = n_k$$

Proof: The beginning of this proof mirrors the proof of Theorem 1 up through the statement. For any $s > 0$

$$\Pr\left(S_{\hat{j},k} > s \mid \hat{P}_{-k}\right) \le J \cdot \max_{j \in \{1,\ldots,J\}} \Pr\left(\tilde{H}_{\hat{j}} - \hat{H}_{\hat{j}} > \frac{s + \delta\sigma_k^2/16M^2}{1+\delta} \mid \hat{P}_{-k}\right)$$

Case 1: When the Bernstein's inequality is applied,

$$\Pr\left(S_{\hat{j}} > s \mid \hat{P}_{-k}\right) \le J \cdot \exp\left(-\frac{\tilde{n}s}{\tilde{b}}\right) \quad \text{where} \quad \tilde{b} = 2(1+\delta)^2\left(\frac{16M^2}{\delta} + \frac{8M^2}{3}\right)$$

Case 2: When Hoeffding's inequality is applied,

$$\Pr\left(S_{\hat{j}} > s \mid \hat{P}_{-k}\right) \le J \cdot \exp\left(-\frac{2\tilde{n}^2}{(1+\delta)^2} \cdot \frac{\left(s + \frac{\delta\sigma_k^2}{16M^2}\right)^2}{\sum_{i=1}^{K}(b_i - a_i)^2}\right)$$

$$= J \cdot \exp\left(-\frac{\tilde{n}}{\frac{1}{2}(1+\delta)^2} \cdot \frac{\left(s + \frac{\delta\sigma_k^2}{16M^2}\right)^2}{\frac{1}{\tilde{n}}\sum_{i=1}^{K}(b_i - a_i)^2}\right)$$

$$\le J \cdot \exp\left(-\frac{\tilde{n}}{\frac{1}{2}(1+\delta)^2} \cdot \frac{s^2}{M^2}\right)$$

$$= J \cdot \exp\left(\frac{-\tilde{n}s^2}{\tilde{h}}\right) \quad \text{where} \quad \tilde{h} = \tfrac{1}{2}(1+\delta)^2 M^2$$

Compare the two cases to calculate when the Hoeffding's bound (case 2) is better than the Bernstein's bound (case 1). Hoeffding's bound is better when:

$$J \cdot \exp\left(\frac{-\tilde{n}s^2}{\tilde{h}}\right) \le J \cdot \exp\left(\frac{-\tilde{n}s}{\tilde{b}}\right) \quad \text{, or when:}$$

$$s \ge \frac{\tilde{h}}{\tilde{b}}$$

(Equivalently, when: $\left(\frac{s^2}{\tilde{h}}\right) = \frac{s}{\tilde{h}} \cdot s \ge \frac{s}{\tilde{h}} \cdot \frac{\tilde{h}}{\tilde{b}} = \left(\frac{s}{\tilde{b}}\right)$) .

Therefore, develop a function that combines both Bernstein's and Hoeffding's inequalities such that when $s \ge \frac{\tilde{h}}{\tilde{b}}$, Hoeffding's bound is used, and when $s < \frac{\tilde{h}}{\tilde{b}}$, Bernstein's bound is used.

Consider the previously sub-lemma of Theorem 1 where for any $u \in \Re$,

$$E(X) \le u + \int_u^{\infty} \Pr(X \ge x)dx .$$

Then for any $u > 0$, define the function $f(u)$in the following fashion:

$$E[S_{\hat{k}}] \le u + \int_u^{\infty} \Pr(S_{\hat{k}} \ge s)ds$$

$$= u + J\int_u^{\max\{u, \frac{\tilde{h}}{\tilde{b}}\}} e^{\frac{-\tilde{n}s}{\tilde{b}}} ds + J\int_{\max\{u, \frac{\tilde{h}}{\tilde{b}}\}}^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}} ds$$

$$= f(u) .$$

It is easily seen that:

When $u \ge \frac{\tilde{h}}{\tilde{b}}$, $\quad f(u) = u + J\int_u^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}} ds$

When $u < \frac{\tilde{h}}{\tilde{b}}$, $\quad f(u) = u + J\int_u^{\frac{\tilde{h}}{\tilde{b}}} e^{\frac{-\tilde{n}s}{\tilde{b}}} ds + J\int_{\frac{\tilde{h}}{\tilde{b}}}^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}} ds .$

Continue the proof of the theorem by finding the minimum of $f(u)$ by the use of the first and second derivative, w.r.t. u.

When $u \geq \frac{\tilde{h}}{\tilde{b}}$, $\qquad f(u) = u + J\int_u^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}}\, ds$

$$f'(u) = 1 - J \cdot e^{\frac{-\tilde{n}u^2}{\tilde{h}}} \overset{set}{=} 0$$

$$\frac{-\tilde{n}u^2}{\tilde{h}} = \ln \tfrac{1}{J}$$

$$u = \sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}}$$

Which implies:

$$u = \sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}} \geq \frac{\tilde{h}}{\tilde{b}} \quad \text{or} \quad \frac{\tilde{b}^2 \ln J}{\tilde{h}} \geq \tilde{n}\,.$$

Note that: $f''(u) = J \cdot e^{\frac{-\tilde{n}u^2}{\tilde{h}}} \cdot \frac{2\tilde{n}u}{\tilde{h}} > 0$, thus a minimum is achieved at the critical point.

Then,

$$E\left[S_{\hat{k}}\right] \leq f\left(\sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}}\right)$$

$$= \sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}} + J\int_{\sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}}}^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}}\, ds$$

Notice that the integral takes the form of a normal probability distribution

$$\sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}} + J\sqrt{2\pi}\int_{\sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}}}^{\infty} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\cdot\frac{2\tilde{n}}{\tilde{h}}\cdot s^2}\, ds \qquad \text{where we}$$

$$let\ u = \sqrt{\frac{2\tilde{n}}{\tilde{h}}} \cdot s$$

$$du = \sqrt{\frac{2\tilde{n}}{\tilde{h}}} \cdot ds$$

$$E[S_{\hat{k}}] \le \sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}} + J\frac{\sqrt{2\pi}}{\sqrt{\frac{2\tilde{n}}{\tilde{h}}}}\int_{\sqrt{2\ln J}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}\, du$$

$$= \sqrt{\frac{\tilde{h}\ln J}{\tilde{n}}} + J\sqrt{\frac{\pi\tilde{h}}{\tilde{n}}} \cdot \left(1 - \Phi(\sqrt{2\ln J}\right)$$

where $\Phi \sim N(0,1)$

When $u < \frac{\tilde{h}}{\tilde{b}}$, $\quad f(u) = u + J\int_u^{\frac{\tilde{h}}{\tilde{b}}} e^{\frac{-\tilde{n}s}{\tilde{b}}} ds + J\int_{\frac{\tilde{h}}{\tilde{b}}}^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}} ds$

$$f(u) = u - J\frac{\tilde{b}}{\tilde{n}} e^{\frac{-\tilde{n}s}{\tilde{b}}}\Big|_u^{\frac{\tilde{h}}{\tilde{b}}} + J\int_{\frac{\tilde{h}}{\tilde{b}}}^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}} ds$$

$$f'(u) = 1 - J \cdot e^{\frac{-\tilde{n}u}{\tilde{b}}} \overset{set}{=} 0$$

$$u = \frac{\tilde{b}\ln J}{\tilde{n}}$$

Which implies:

$$u = \frac{\tilde{b}\ln J}{\tilde{n}} < \frac{\tilde{h}}{\tilde{b}} \quad \text{or} \quad \frac{\tilde{b}^2\ln J}{\tilde{h}} < \tilde{n}\,.$$

Note that: $f''(u) = J \cdot \frac{\tilde{n}}{\tilde{b}} e^{\frac{-\tilde{n}u}{\tilde{b}}} > 0$, thus a minimum is achieved at the critical point.

Then,

$$E\left[S_{\hat{k}}\right] \le f\left(\frac{\tilde{b}\ln J}{\tilde{n}}\right)$$

$$= \frac{\tilde{b}\ln J}{\tilde{n}} + J\int_{\frac{\tilde{b}\ln J}{\tilde{n}}}^{\frac{\tilde{h}}{\tilde{b}}} e^{\frac{-\tilde{n}s}{\tilde{b}}}\,ds + J\int_{\frac{\tilde{h}}{\tilde{b}}}^{\infty} e^{\frac{-\tilde{n}s^2}{\tilde{h}}}\,ds$$

$$= \frac{\tilde{b}\ln J}{\tilde{n}} - J\frac{\tilde{b}}{\tilde{h}}\left[e^{-\frac{\tilde{n}}{\tilde{b}}\frac{\tilde{h}}{\tilde{b}}} - e^{-\frac{\tilde{n}}{\tilde{b}}\frac{\tilde{b}\ln J}{\tilde{n}}}\right] + J\sqrt{\frac{\pi\tilde{h}}{\tilde{n}}}\left(1-\Phi\left(\sqrt{\frac{2\tilde{n}}{\tilde{h}}}\cdot\frac{\tilde{h}}{\tilde{b}}\right)\right)$$

$$= \frac{\tilde{b}}{\tilde{n}}\left(1+\ln J\right) - J\left(\frac{\tilde{b}}{\tilde{n}}e^{-\frac{\tilde{n}\tilde{h}}{\tilde{b}^2}} - \sqrt{\frac{\pi\tilde{h}}{\tilde{n}}}\cdot\left(1-\Phi\left(\frac{\sqrt{2\tilde{n}\tilde{h}}}{\tilde{b}}\right)\right)\right)$$

where $\Phi \sim N(0,1)$

The same arguments can be applied to $E\left[T_{\tilde{k}}\right]$ which would produce the desired results.

■

4.3.3 Discussion

A study of the result of Theorem 1 proves to be quite interesting.

$$0 \quad \le E\left[\hat{R}_{\hat{j}} - R_0\right] \le (1+2\delta)E\left[\hat{R}_{\tilde{j}} - R_0\right] + 2C\frac{K(1+\ln J)}{n}$$

As long as the number of models, *J*, does not grow exponentially with n, note that the bounding term approaches zero as the number of observations grows large, i.e.

$2C\frac{K(1+\ln J)}{n} \to 0$, as $n \to \infty$. Now observe that $\hat{R}_{\tilde{j},k} \le \hat{R}_{\hat{j},k}$, the conditional oracle risk model is less than or equal to lowest conditional risk in relation to the models in use. This implies $\left(\hat{R}_{\tilde{j},k} - R_0\right) \le \left(\hat{R}_{\hat{j},k} - R_0\right)$. Thus as $n \to \infty, (1+2\delta)\, E[\hat{R}_{\tilde{j}} - R_0] \to E[\hat{R}_{\hat{j}} - R_0]$ from above for all $\delta > 0$. In other words, as the number of observations grows the model found by cross validation becomes closer, on average as expressed by the expected value, to the oracle model. Thus the theorem has the following implications:

First, the theorem provides some justification for K-fold cross validation. In K-fold cross validation, the *K* different training/validation sets produces *K* different models. The combinations of these *K* different models will perform, on average as indicated by the expected value in the theorem, at least as well as any of the validation models separately in an asymptotic sense.

The theorem also provides a safe way of choosing a model from among a set of candidate models. Here the combination of the *J* many separate rules from the set of all plausible rules, $\mathcal{F}$, for the given, but unknown, distribution. The combination of the rules in conjugate will not, on average, perform any worse than the best performing rule separately even if one of the separate rules is the true rule of the distribution (van der Laan et al., 2007).

Theorem 2 improves on the bound in Theorem 1 by combining the use of both Bernstein's inequality and Hoeffding's inequality. Bernstein's bound produces a lower upper bound for $\Pr\left(S_{\hat{j}} > s \mid \hat{P}_{-k}\right)$ when $s < \frac{\tilde{h}}{\tilde{b}}$, i.e. when *s* is less that the ratio of the Hoeffding's constant to the Bernstein's constant. When $s \ge \frac{\tilde{h}}{\tilde{b}}$, Hoeffding's bound is a

lower upper bound. Figure 1 illustrates this concept. Though the example is not necessarily realistic with *K* equal to only 3 models and the number in the training set, *n*, having 5,000 samples, the point is well illustrated showing that the upper bound produced from Bernstein's inequality is lower than the upper bound produces from Hoeffding's inequality up to the point $s = \frac{\tilde{h}}{\tilde{b}}$ (≈ 0.0134 in our example). After this point the roles switch and the bound produced by Hoeffding's inequality provides a lower upper bound.

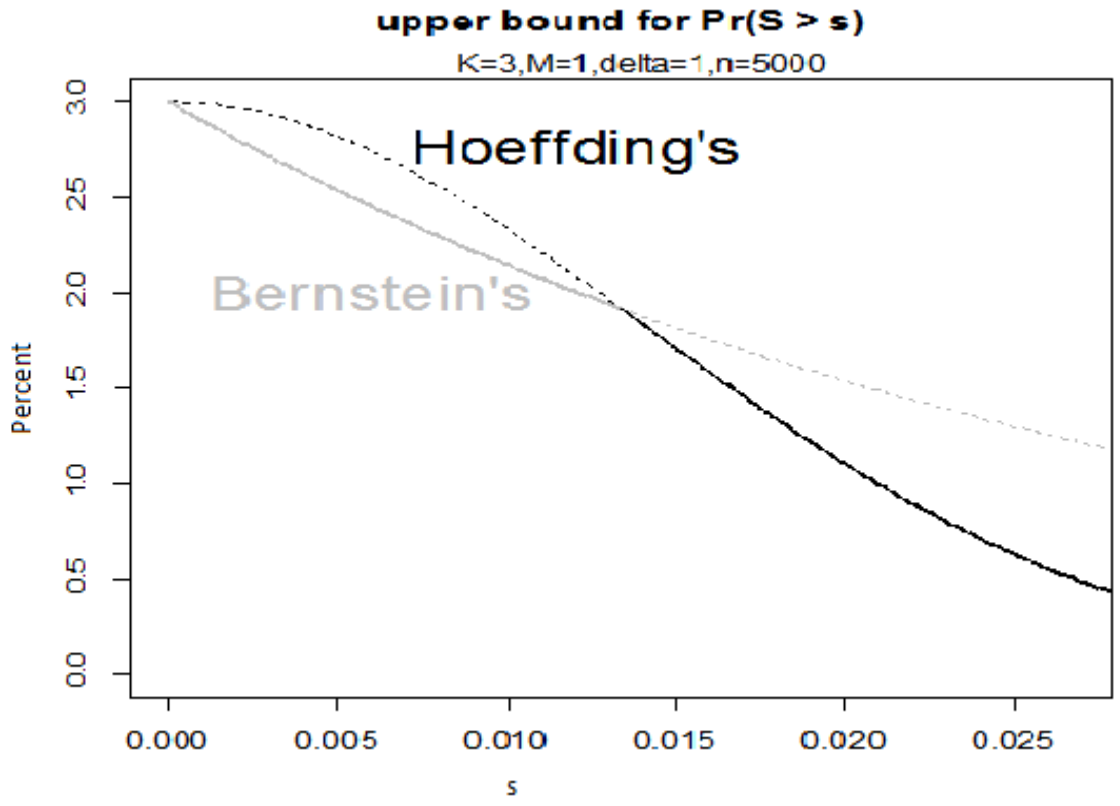

Figure 3: Hoeffding's inequality vs. Bernstein's inequality w.r.t. s. In the example: $K = 3, M = 1, \delta = 1$ *and* $n = 5{,}000$, *the use of Hoeffding's inequality produces a lower upper bound for* $\Pr\left(S_j > s \mid \hat{P}_{-k}\right)$ *when s>0.0134.*

Figure 2 illustrates how the improved bound of Theorem 2 is lower than the Bernstein's bound produced in Theorem 1. In this example, a training sample size of around n = 200 is needed to achieve an upper bound of 0.2 for $E[S_j]$ using the improved bound where as a training sample of nearly n = 1570 is needed for the Bernstein's bound.

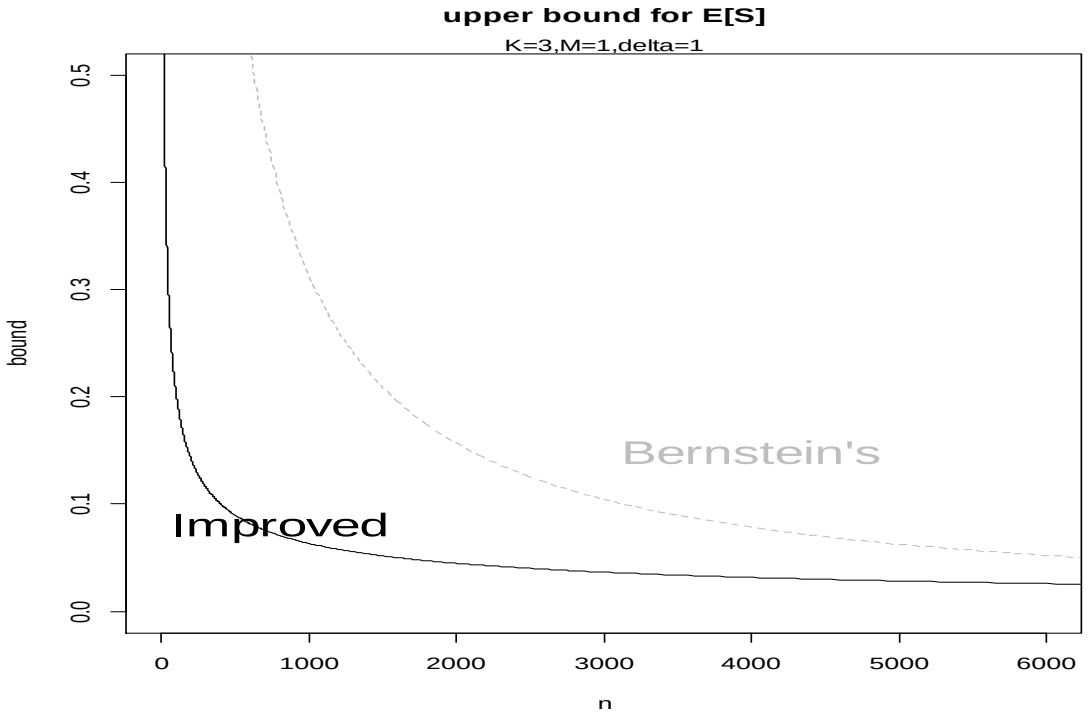

Figure 4: Hoeffding's inequality vs. Bernstein's inequality w.r.t. n, the number in the training sample size. The use of Hoffding's inequality requires a smaller training sample sizes than the Bernstein's inequality in order to produce equivalent results.

Figure 3 shows a comparison for the minimum lower bound of $E[S_j]$ using both the Bernstein's bound and the improved bound for various deltas. The improved bound of the minimum lower bound given delta is nearly linear whereas the Bernstein's bound is quadratic at best. In this example the two bound are closest at $\delta = 1.21$ with a difference of 0.03386. This observation of where the two curves have the smallest deviation is not particularly important being that this is just a single example, however, the observation of where dramatic deviation occurs is of interest: when δ approaches zero and when δ is large. This observation seems to be typical.

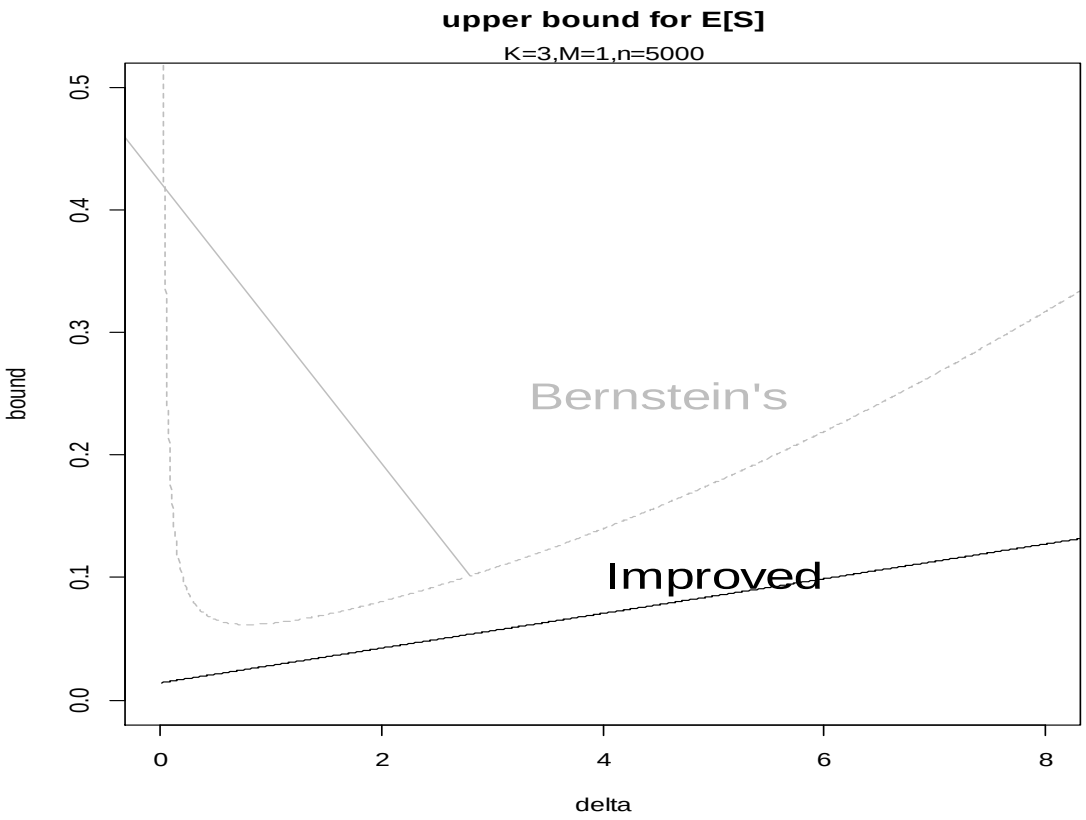

Figure 5: Hoeffding's inequality vs. Bernstein's inequality w.r.t. δ. The use of Hoeffding's inequality produces lower upper bounds no matter what delta is used.

4.4 Variable Bagging

The idea of variable bagging was motivated by bagging or bootstrap aggregating (Breiman, 1994). As the name suggest, bagging repeatedly takes a random subset from a set of data point in the form of (*Y, X*), where *Y* is a numeric resultant and *X* is a multi-dimensional vector of inputs, and develops a predictive or regression model for each of the random samples. The final model is the aggregate mean of the random samples.

Variable bagging, on the other hand, randomly selects a subset of positions of the *X* vector or the variables. In this instance the number of data points used to train the model is not decreased but rather the dimension of each data point is decreased. This dimension reduction proves quite valuable for those algorithms which would be restricted due to the dimension.

Inspiration came, per say, not by viewing the procedure as reducing the number of variables but by rewriting the data set and taking liberties in the bagging procedure. Consider a training set of l data point in the form (*Y, X*), where *Y* is a numeric resultant and *X* is vector of length *n*. Typically for bagging, random samples of around 80 to 90% of the data points are taken and models are developed. However, in cases where $n >> l$, dimensionality eliminates from consideration many modeling algorithms due to the instability of the l x n matrix.

Now consider rewriting the data set such that for each data point (Y_i, X_i), where X_i is of length *n*, into a combination of ${}_nC_m$ subsets $(Y_i, X_{i,j})$ where $X_{i,j}$ is of length *m* with $m < l << n$. Now by randomly selecting the *m* variables from the X_i's , you are actually selecting *l* elements which will be used for modeling from the data set containing

$l \cdot_n C_m$ elements. The bagging theory does not specify what percentage of points needs to be selected for each trial in the bagging process.

Intuitively, using the cross-validation selector within bagging could do better than using just one algorithm for all the trials in the bagging process.

Using cross-validation on any individual trial can at most perform as well as the best algorithm available. The previous theorem shows that cross validation will come close to the best. In other words, cross-validation will not necessarily choose the best performing algorithm but will instead choose the safest algorithm. This is quite reassuring especially when the best performing algorithm is not known beforehand.

Bagging, as shown in previous sections, has the ability of improving predictability. The amount of improvement depends on the variability of the data. In combination with cross-validation within bagging, the aggregate model can not only be a safe model and come close to the best model which uses only a single algorithm, but has the possibility of beating this best uni-algorithm model. This variability caused by the multi-algorithm cross-validation selection within variable-bagging aids in the predictability. In short, there are instances (but not all the time) where the aggregate of the cross-validation selectors will perform better than the aggregate of using only a single algorithm. This property is illustrated in the next chapter of Examples.

5 EXAMPLES

This chapter explores the question: “How would you use logistic regression in the development of a prediction model to determine whether or not a brain has Alzheimer’s?”

Logistic regression is a powerful tool for classification problems invloving two groups. It is unfortunate that the only attempt I found using this powerful device defined the discriminators as the thickness of the entorhinal cortex, the thickness of the supramarginal gyrus and the volume of the hippocampus (Marcus et al., 2007). However, logistic regression has its limitations: dimensionality. This could be the reason for its lack of use. The algorithm becomes unstable as the number of variables approaches the number of observations. With 2,122,945 voxels (possible variables) in each image and only 149 images available for training, dimensionality is a problem.

Example 1 provides a typical solution to the question. With the use of reason, a lot of work and luck, the dimension of the brain was reduced form 2,122,945 voxels to just a handful of 35 voxels. Logistic regression was able to produce a model with a success rate of 84.93% which used the limited amount of information that the 35 voxels were able to provide.

Using logistic regression again, example 2 demonstrates the prediction power of variable bagging. The three cases within this example provide an illustration of how the combination of predictive models, even poor models, can yield a stronger model. Case 3 goes on to demonstrate the importance of variability within each model by developing models with as high as 87% success rates.

Why limit ourselves to using only logistic regression? By incorporating other algorithms such as neural networks, support vector machines and decision trees, example 3 does not limit itself to just logistic regression. The cross-validation selector selects the best reasonable, or safe, algorithm to use on each of a given set of 35 randomly selected voxels. The result of the use of the cross-validation selector on each of the individual models within a bagged set of models produced success rates as high as 89.97%.

The 295 MRI brain images were provided by the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu), and were divided, 149 training and 146 test images, in accordance with the paper by Cuingnet et.al. (2010) paper which is used as a guide as reasonable outcomes for our examples.

5.1 Example One: "Hand Picking" the Variables

The ultimate goal is to develop a model having a high success rate in the classification Alzheimer's from a MRI brain scan. This example tries to develop a predictive model by way of combining specially selected variables. The method of selecting these variables is through reasoning with the hope of coming up with a useful model.

5.1.1 Dimension Reduction

A common procedure to test if there is a difference between two groups is the Welch's t-test (Salas-Gonzalez et al., 2010). Since Alzheimer's disease is a deterioration of portions of the brain, the identification of voxels with significant large positive and negative t-values would indicate areas with differences between the two groups: the normal control (NC) group and Alzheimer's disease (AD) group. Therefore, it is these areas of interest which should be the most helpful in classification. The Welch's t-statistic was calculated at each voxel using:

$$t_{value} = \frac{\bar{x}_{NC} - \bar{x}_{AD}}{\sqrt{\frac{s_{NC}^2}{n_{NC}} + \frac{s_{AD}^2}{n_{AD}}}} \quad ,$$

where $\bar{x}_{NC}$ and $\bar{x}_{AD}$ are the average intensity, s_{NC}^2 and s_{AD}^2 are the variances of the intensities, and n_{NC} and n_{AD} are the numbers of the normal control and Alzheimer's disease groups respectively.

From the 149 samples in the training group, 80 were classified as NC and 69 as AD. The t-values ranged from -6.410 to 5.382. Historically, those areas with t-values less than -1.96 and those greater than 1.96 are considered significant. Using these historic criteria located 111,773 voxels with t-values less than -1.96 and 64,259 voxels with t-values greater than 1.96.

5.1.2 Further Reduction of Variables

The logistic regression algorithm is used to determine whether or not a given MRI brain scan should be classified as having Alzheimer's disease or not. The problem with using logistic regression is one of dimensionality. The algorithm becomes unstable as the number of variables becomes close to and greater than the number of data values.

Voxels with large t-values (both positive and negative) should be good candidates for variables. This criterion reduced the number of voxels to about 175,000 from over 2.1 million voxels available in a brain scan. Still this number is too large if the logistic regression algorithm is to be used. To further reduce the number of variables, a second thought was considered, use only those voxels which have a high predictability property.

To locate voxels with high predictability, local logistic regression was performed on 16,128 voxel clusters. A voxel cluster consists of the center and the 6 closest voxels which are one voxel unit away. The centers were chosen to be 5 voxel units apart. Using the 149 training-sample MRI scans, a randomly selected set of 100 would serve as the training set, at each point cluster, using the local logistic regression algorithm. The remaining 49 scans were used as a validation set. Each of the voxel cluster models were tested against the corresponding voxel clusters of the validation set and success rates were calculated.

A graph using the t-values and success rates was produced to see if there was a large correlation between the t-values and the success rates at each point cluster. If there was a correlation between the two, a V-shaped image would be visible in the graph. Figure 1 show no distinct V-shape indicating a low correlation between t-values and success rates. However, by inspection, the figure does indicate that the majority of

voxels with t-values between -2.5 and 2.5 had low accuracy rates (falling below 70%) but there is very little correlation that the t-values with the greatest negative or positive values produced the most accurate models.

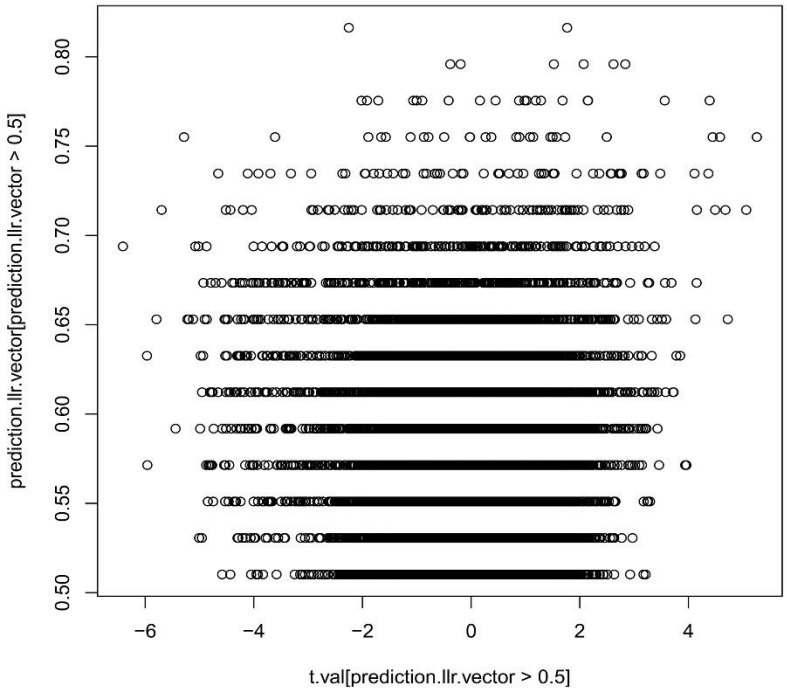

Figure 6: The graph of success rates for t-values vs. local logistic regression calculated at every 5th voxel.

Both ideas were combined to again reduce the dimensionality. By considering those voxels with t-values greater than two standard deviations and the point clusters having a validation rate greater that 75% reduced the variables down to 12 voxel clusters centered at the following locations:

Voxel number	Centered at:
Vox 1	(80,55,50)
Vox 2	(85,50,45)
Vox 3	(45,60,60)
Vox 4	(80,85,40)
Vox 5	(30,65,35)

Vox 6	(75,80,30)
Vox 7	(40,90,35)
Vox 8	(45,80,45)
Vox 9	(75,80,40)
Vox 10	(80,85,35)
Vox 11	(45,80,40)
Vox 12	(45,85,35)

Remembering the smoothing practices in the preprocessing procedure, redundancy became a slight concern and an excuse to reduce the dimension, again, with respect to the distance between voxel clusters. Voxels within 10 units of each other were grouped together. The voxel clusters with the highest validation rate were then chosen to represent the grouping. This reduced the number of variables to only 5 voxel clusters. In other words, out of over 2.1 million voxels available, 35 were selected.

Vox 1 represents the grouping of {Vox 1, Vox 2}

Vox 3 represents itself

Vox 4 represents the grouping of {Vox 4, Vox 6, Vox 9, Vox 10}

Vox 5 represents itself

Vox 7 represents the grouping of {Vox 7, Vox 8, Vox 11, Vox 12}

Other statistics were also introduced in order to get a feel for the data. Standard deviations of the rectangular areas covered were represented by the grouping of Vox 4, Vox 7, and the total 12 groupings were considered: SD 4 is the standard deviation of voxels ranging from x = 74 to x = 81, y = 79 to y = 86, z = 29 to z = 41; SD 7 is the standard deviation of voxels ranging from x = 39 to x = 46, y = 79 to y = 91, z = 34 to z =

46; SD Total is the standard deviation of voxels ranging from x = 29 to x = 86, y = 49 to y = 91, z = 29 to z = 51.

5.1.3 Results

Local logistic regression models were developed on combinations of the point clusters and the standard deviations of the areas covering the sets. The success rates using the 146 test images are given in Table 1 below.

		SD 4	**SD 7**	**SD Total**	**Sd 4,7**	**Sd 4,7,T**
		0.6849	0.6369	0.5548	0.6918	0.7055
Vox 1	0.6164	0.7192	0.7877	0.7328	0.7808	0.7808
Vox 3	0.6986	0.6918	0.7260	0.6849	0.7192	0.7192
Vox 4	0.6986	0.6986	0.7123	0.6986	0.7123	0.6999
Vox 5	0.5822	0.6096	0.6233	0.5959	0.6233	0.6369
Vox 7	0.7534	0.7466	0.7740	0.7740	0.7603	0.7603
Vox 1,3	0.7055	0.7329	0.7466	0.6849	0.7603	0.7603
Vox 1,4	0.7397	0.7458	0.7740	0.7466	0.7739	0.7808
Vox 1,5	0.6781	0.7192	0.7534	0.6986	0.7603	0.7603
Vox 1,7	0.7466	0.7466	0.7466	0.7329	0.7534	0.7329
Vox 3,4	0.7397	0.7329	0.7740	0.7329	0.7671	0.7612
Vox 3,5	0.6781	0.6849	0.6986	0.6644	0.7055	0.7055
Vox 3,7	0.7603	0.7603	0.7671	0.7397	0.7603	0.7397
Vox 4,5	0.6644	0.6575	0.6918	0.6644	0.7055	0.6849
Vox 4,7	0.7534	0.7603	0.7466	0.7534	0.7466	0.7603
Vox 5,7	0.6849	0.6849	0.6781	0.6712	0.6712	0.6712
Vox 1,3,4	0.7534	0.7397	0.7877	0.7534	0.7808	0.7603
Vox 1,3,5	0.7192	0.7466	0.7671	0.7260	0.7671	0.7808
Vox 1,3,7	0.7603	0.7740	0.7740	0.7534	0.7740	0.7671
Vox 1,4,5	0.7877	0.8014	0.8082	0.8014	0.7945	0.7945
Vox 1,4,7	0.7612	0.7740	0.7808	0.7397	0.7808	0.7192
Vox 1,5,7	0.7534	0.7534	0.7397	0.7466	0.7534	0.7397
Vox 3,4,5	0.7329	0.7466	0.7671	0.7397	0.7808	0.7466
Vox 3,4,7	0.7945	0.8114	0.7808	0.7260	0.7740	0.7329
Vox 3,5,7	0.7466	0.7466	0.7466	0.7466	0.7534	0.7330
Vox 4,5,7	0.7663	0.7466	0.7740	0.7671	0.7466	0.7466
Vox 1,3,4,5	0.7877	0.7877	0.8356	0.7945	0.7945	0.8219

Vox 1,3,4,7	0.7740	0.7740	0.7945	0.7945	0.7808	0.7534
Vox 1,3,5,7	0.7945	0.7740	0.8082	0.8014	0.8014	0.7877
Vox 1,4,5,7	0.8082	0.8082	0.8082	0.7808	0.8082	0.7877
Vox 3,4,5,7	0.8014	0.8014	0.8151	0.7808	0.8082	0.7945
Vox 1,3,4,5,7	0.8493	0.8493	0.8425	0.8082	0.8493	0.8356

Table 1: The success rate using the five specific point clusters. Models were developed using the 149 training images and the local logistic regression algorithm. Success rates were calculated using the 146 test images. The highlighted cells are the maximums success rates for each of the combination numbers.

Looking at the first column of Table 1, the success rates generally increased with the addition of another voxel cluster. The exception, it seems, occurs when a poor predictor is given too much weight in the grouping. Of the 5 point clusters, Vox1 and Vox5 are the two lowest performers. Though not particularly useful in the early combinations of two and three, Vox1 and Vox5 were part of the set of four which achieved the highest success rate in the combination number class. Also note the nearly 5% increase in success when Vox1 was added to the set Vox3,4,5,7. Even though a point cluster might seem insignificant by itself, its inclusion within a set adds diversity and thus becomes a better representation of the population. Adding more and more data which represents the population produces better models of prediction.

Achieving a success rate of 84.9% is fairly good and does fall within the range described in the Cuingnet et. al's paper: "Automatic classification of patients with Azheimer's disease from Structural MRI: A Comparison of ten methods using the ADNI database." However, it took a lot of work to find the voxels to achieve these results. This raises the question: is this procedure repeatable for other algorithms or is finding the variables a matter of luck?

5.2 Example Two: Variable Bagging

In this example, the logistic regression algorithm is used again to help illustrate the power and reliability of the variable bagging procedure. When using variable bagging, a combination of statistical models performs better, on average, than any one singular model using the same algorithm. Also, dimensionality is reduced by considering the t-values with significant values.

5.2.1 Case 1: Variables with t-values < -1.96

Logistic regression was first performed using the entire training set and the variables defined as a random sample of 35 points from the set of voxels having t-values < -1.96 . This model was then used to find predictive values, represented by the probability of having Alzheimer's, for each of the 146 images of the test data. This procedure was repeated ninety-nine more times to produce a bagged model which found the mean of the 100 individual predictions to produce a final prediction. Success rates are described as the ratio of number of correct predictions to the total number of images.

For each of the individual models, success rates typically ranged from around 0.55 to around 0.70, with a mean of 0.6357 (standard deviation 0.0391). Below I have listed the success rates of the first and last 10 trials from a set of 100 models:

(0.6506849, 0.5890411, 0.6917808, 0.6917808, 0.5410959, 0.7397260, 0.6712329, 0.6917808, 0.6438356, 0.5958904, . . . , 0.6917808, 0.6712329, 0.6095890, 0.7054795, 0.6506849, 0.6095890, 0.6506849, 0.6780822, 0.6712329, 0.6369863)

For comparison, for the variable bagged model, the mean of the 100 predicted values for the test data of the individual models had an overall success rate of 0.7328767. While this rate is not particularly high, this aggregated rate is higher than the success rates of 99 out of the 100 individual models. It is also nearly 10 percentage points, or 2.4 standard deviations, higher than the mean of the individual success rates.

To illustrate consistency, the entire procedure was repeated 10 more times with the results shown below in Table 2.

Success rates of the mean prediction values	Percent of individual success rates beaten by the average predicted success rate
0.712	96%
0.733	99%
0.740	100%
0.719	98%
0.726	98%
0.712	98%
0.767	100%
0.747	100%
0.733	99%
0.736	99%

Table 2: Applying variable-bagging with the variables having t-values < -1.96. These success rates show the results of models using the mean of 100 predictive values and then compares these aggregate models with the success rate of the individual models used in the aggregate.

The chart illustrates that the success rate of the mean predicted values, or aggregate values, consistently beat most, if not all, of the success rates of the individual models. Intuitively, the individual models can be thought as forming a committee which "votes" on the final outcome. This aggregate vote produces a model which becomes stronger than most, if not all, of the individual models.

5.2.2 Case 2: Variables with t-values > 1.96

The procedure was repeated a second time but the 35 random variables selected for each individual model were selected from the set of t-values having values greater than 1.96. The results were not as strong as in case 1 but made the same points. Typically, the success rates for the individual models ranged from around 0.45 to almost 0.70 success rate, with a mean of 0.578 (standard deviation of 0.0488) which is nearly 5 percentage points lower than in the case above.

Below the success rates of the first and last 10 individual trials, out of a set of 100 models, are listed:

(0.5821918, 0.5753425, 0.4726027, 0.5136986, 0.5684932, 0.5821918, 0.5342466, 0.6301370, 0.6438356, 0.5890411, . . . , 0.5205479, 0.5205479, 0.5342466, 0.5547945, 0.5821918, 0.6027397, 0.6095890, 0.5821918, 0.6095890, 0.5479452)

The success rate of the aggregate model was 0.6506849. This rate is better than 94 out of the 100 individual model's success rates. Upon repeating the procedure, this

seems to be an extreme case in relation to the number of times the aggregate model success rate beat the individual models success rates. Table 3 shows the results of the procedure repeated ten more times.

Success Rates of the aggregate model	Percent of individual success rates beaten by the aggregate model
0.651	89%
0.658	91%
0.601	80%
0.589	68%
0.623	78%
0.623	81%
0.589	70%
0.610	79%
0.623	79%
0.603	68%

Table 3: Applying variable-bagging with the variables having t-values > 1.96. These success rates show the results of models using the mean of 100 predictive values and then compares these aggregate models with the success rate of the individual models used in the aggregate.

Though the results are not particularly impressive, Table 3 does illustrate that the success rate of the aggregate model consistently beat most of the success rates of the individual models.

Also, note that there are several individual models with success rates below or near the 0.50 value, a value that represents a guess. Thus, this case shows that variable bagging can incorporate poor and non-predictive models to produce a more successful model.

5.2.3 Case 3: Mixture of Data with t-values < -1.96 and > 1.96

In previous discussions about Bagging, the strength of the aggregate model becomes stronger as the variables become more diverse. Therefore, we can guarantee diversity by selecting a set of randomly selected variables which come from both the set of t-value having values less than -1.96 and t-values having values that are greater than 1.96. A mixture of 18 voxels with t-values that were less than -1.96 and 17 voxels greater than 1.96 were randomly picked from their respective sets. The procedure was repeated a third time resulting with some surprising results.

The success rates for the individual model typically ranged from about .65 to .75 with a mean of 0.6953 (standard deviation 0.0444). Right away the improvement can be seen to be due to the diversity introduced. The improvements show 5 percentage points better than the models using only the t-values less than -1.96 and about 10 percentage points better than the models using the data points from the set of t-values greater than 1.96.

The first and last 10 success rates for the individual models are listed below:

(0.6917808, 0.7328767, 0.7260274, 0.6506849, 0.6849315, 0.7602740, 0.6232877, 0.6780822, 0.6986301, 0.6917808, . . . , 0.6643836, 0.6506849, 0.7397260, 0.6780822, 0.7465753, 0.7328767, 0.7465753, 0.7671233, 0.6369863, 0.7328767)

The aggregate model for this combined data had the success rate of 0.8287671. This success rate is around 13 percentage points (3 standard deviations) better than the

average success rates of the individual models and was better than 99% of the 100 individual models of this set.

Table 4 shows the results from 10 different trials. Note that it was not uncommon for the success rate of the aggregate model to be better than all of the 100 individual success rates.

Success Rates of the mean prediction values	Percent of individual success rates beaten by the average predicted success rate
.870	100%
.856	100%
.836	100%
.856	100%
.849	100%
.829	100%
.849	100%
.849	100%
.829	100%
.869	100%

Table 4: Applying variable-bagging with the variables having both t-values< -1.96 and t-values > 1.96. These success rates show the results of models using the mean of 100 predictive values and then compares these aggregate models with the success rate of the individual models used in the aggregate.

5.2.4 Discussion

The main point of this example is to demonstrate the power of variable bagging. First, by combining, or bagging, the separate models which predict the same event resulted in an aggregate model that was stronger, in most instances, than any of the

individual models. As illustrated in Case 2, even in situations where the underlying models are weak or even non-predictive, the aggregate model improved the prediction performance.

Secondly, diversity in the variables strengthen the model overall both for the individual models and especially in the aggregate models. As in Case 1 and 2, the diversity came by randomly selecting 35 voxels or variables from a large sample set, Case 1 used negative t-values and Case 2 used positive t-values. This diversity delivered a stronger model in the aggregate than in any of the individual models most of the time. However, by selecting variables from both the sets of negative and positive t-values, diversity was guaranteed and as a result the success rates of both the individual and aggregate models were improved.

In the previous section it was questioned whether or not finding the 35 variables was luck. This example produced similar success rates, a couple even higher, without the extensive dimension reduction. One of the things that favored Example 1 was that the final model did have variables that included both positive and negative t-values. However, Example 2 did indicate that finding an individual model by randomly selecting variables from the positive and negative t-values did not guarantee such high success rates.

5.3 Example Three: Using K-Fold Cross Validation Selector

So far in this section of examples, the logistic regression algorithm was used to model the data. But is logistic regression the best algorithm to use? Admittedly, logistic

regression was initially selected with spite because there were few procedures in the past that considered logistic regression. Those that did, used the variable associated with areas of measurement, such as the thickness of the supramarginal gyrus and the volume of the hippocampus (Marcus et al., 2007), and never with individual voxels.

V-fold cross-validation is a procedure which uses subsets of the training data to train a model and the remaining portion of the training set as a validation set to test the model. This procedure eliminates bias. However, the elimination of bias comes with the price of increased variance.

V-fold validation randomly divides the training data into V nearly equal disjoint validation sets. This example will use V=5. The compliment of each validation set will become the training set for the modeling algorithm. Each model will then be "validated" or tested by the corresponding validation set. Thus, from the five disjoint validation sets come prediction values for each data entry covering the entire training set. From here a success rate can be calculated in order to assess how well the modeling algorithm preforms. Please note that we are testing how well the modeling algorithm performs and not a particular model which was produced by the modeling algorithm since five different- but similar- models make up the resultant predicted values of the training set.

5.3.1 Using the Cross-Validation Selector – Individual Case

This example will use cross-validation to select an algorithm to be used for a given set of training data. By first randomly selecting 35 variables or voxels (18 from the negative t-values and 17 from the positive t-values), cross-validation will be performed

using the modeling algorithms: logistic regression (lr), neural networks (nnet), support vector machines (svm) and decision trees (rpart). Success rates will be calculated for each algorithm. The algorithm with the highest success rate (fielders choice for ties) will be chosen to model the entire training set.

To illustrate and to get a feeling for the procedure, the cross-validation selection was performed 100 times on the training data on 100 different randomly selected sets of variables. Preliminary results are given below:

	Training Data				**Test Data**			
	lr	**nnet**	**svm**	**rpart**	**lr**	**nnet**	**svm**	**rpart**
Mean success rate	0.718	0.748	0.741	0.634	0.695	0.727	0.717	0.620
Standard deviation	0.033	0.036	0.040	0.050	0.044	0.050	0.050	0.050
Number of times chosen as maximum	14	56	38	0	15	49	41	1

Table 5: Running the cross-validation selector on 100 individual trials.
Note: the number of times chosen as maximum is greater than 100 due to ties.

Looking at the results from the training data, one would hope that the neural networks algorithm (nnet) would be chosen as the algorithm to use in modeling since it had the highest mean success rate (0.748) and was the algorithm in which the cross-validation selector chose the most times (56). Indeed, when modeling the training data, neural networks did have the highest mean success rate (0.727) and the success rate was the highest most often (49). However, the cross-validation selector selects the algorithm on the individual basis and not in aggregate. The better question to ask is how many times the cross-validation selector correctly selected the best algorithm. If one were to randomly guess, one should pick the correct algorithm about 25% of the time. Even

knowing that the decision-tree algorithm usually came in last in the rankings, still one should pick the correct algorithm around 33% of the time. The cross-validation selector successfully chose the winning algorithm 43.5 out of 100 times. (The decimal accounts for ties.)

Choosing the correct algorithm only 43.5% of the time may not impress many people but it is still better than random guessing. Reproducing the example three other times produced results of 49, 46 and 48 out of 100 times for predicting the best algorithm used on the test data.

5.3.2 Using the Cross-Validation Selector Inside Variable-Bagging

For this example, within each variable bagging model are 100 individual models. The cross-validation selector is not applied to the entire bagged model but to each individual trial within each bag. Thus, it is possible for all algorithms to be represented within any one bag depending on the voxels used in the individual models. For example, logistic regression might be used to model the first set of training data while support vector machines might have been selected to model the data which used a different set of randomly selected variables. In case there is a tie between any of the algorithms during the selection portion, all algorithms associated with the tie will be used and the average of their predictions will be used as the result of the trial.

The example was run 100 times where each time the variable bagging model had 100 individual trials.

	Training Data				Test Data			
	lr	**nnet**	**svm**	**rpart**	**lr**	**nnet**	**svm**	**rpart**
Mean success rate	0.8464	0.8430	0.8489	0.7625	0.8516	0.8614	0.8540	0.7942
Standard deviation	0.0124	0.0111	0.0104	0.0167	0.0127	0.0121	0.0107	0.0150
Number of times maximum	44	26	63	0	25	53	26	0

Table 6: Comparison of 100 variable-bagged trials for each algorithm. Note: the number of times chosen as maximum is greater than 100 due to ties.

By looking at the results using the training data, it is hard to know which algorithm would be best to use for the prediction of the test data. Considering the mean and/or the number of times an algorithm produced maximum results with respect to the 100 bagged trials of the cross-validated training data, there is no way to suggest that nnet would give the best results when using the test data. The more likely candidate would be the svm algorithm, if you were forced to pick.

However, you do not have to choose which algorithm to use for the overall variable bagging trials if the cross-validation selector is used on each individual trial within a bagging model (averaging the predictors for ties). The results using the cross-validation selector on each of the individual trials within a variable-bagged set of 100 are:

	100 variable bagging trials using the cross-validation selector at each individual trial.
mean	0.8611
Standard deviation	0.0118

The mean result did not beat the mean result of the highest mean using single algorithms in the variable bagging procedure, which was neural networks, but it is very

close. The theorem does not guarantee the best and actually says that the cross validation selector model is bounded above by the best possible model available in the individual results. In other words, I did not know which model to choose in modeling the test data, however the cross-validation selector produced models, on average, that where very close to the best and was better than the model I would have guessed for this example.

The breakdown of how well the cross-validation selector in the variable bagging models in relation to the other uni-algorithmic variable bagging models is given below:

1st	20
Tied for 1st	31
2nd	16
Tied for 2nd	12
3rd	7
Tied for 3rd	10
4th	3
Tied for 4th	1
5th	0

Table 7: How the cross-validation within variable-bagging ranked in comparison to uni-algorithmic variable bagging models of 100 trials.

In other words, the variable bagging model which used the cross-validation selector at each trial had the highest success rate or tied for the highest success rate 51 out of 100 times. Furthermore it came in second or better 79 out of 100 times.

6 DISCUSSION

6.1 Overview

Several procedures have been used in the past which combine multiple statistical models in order to produce a single model which better describes the data to improve predictability and/or classification. Such ensemble methods include Stacking (Leo Breiman, 1996), Boosting, Blending(Hastie et al., 2009), and the Superlearner (M. J. van der Laan et al., 2007). One of the most notable, if not the most famous, examples came from the Netflix Prize collaborative filtering competition (Bell, Koren, & Volinsky, 2008). The lesson learned from the winning team, Bellkor's Pragmatic Chaos, was that the combination of many approaches from a diverse group performed better than a small number of more powerful algorithms. The winning algorithm averaged the results of over 800 different algorithms to win the million dollar prize. The final team itself, Bellkor's Pragmatic Chaos, was a combination of three separate teams created at the beginning.

This idea of bring together many disperse models is also supported by the theory developed in several papers by Mark J. van der Laan and his collaborators (M. J. van der

Laan & Dudoit, 2003). One of their theorems shows that, as the number of points in the training set grows, the expected value of the conditional risk for the model found by cross validation will be bounded above by a quantity close to the expected value of the conditional risk for the best model in consideration. I was able to improve upon this bound by incorporating both the Bernstein's inequality and Hoeffding's inequality (Györfi et al., 2002).

This current work uses these ideas of combining several different algorithms to come up with an overall algorithm which approaches the best algorithm in consideration. In practice when working big data problems, there is often no indication beforehand to know which algorithm should perform best. This is the case in this work with real (non-simulated) data on MRI brain scans for the prediction of Alzheimer's disease. The study used 149 scans as training data and 146 scans as test data. These numbers and grouping of scans were chosen in accordance to a paper by Rémi Cuingnet et. al. (2011) to serve as a benchmark. Much credit must be given to the ADNI (Alzheimer's Disease Neuroimaging Initiative) database for the data provided; data and information on this dataset may be found at www.adni.loni.ucla.edu/ .

After preprocessing the scans for standardization, there were over 2.1 million voxels to work with for each subject. Since we have only 149 training subjects, the number of voxels is far too large to directly use some of the most common classification algorithms such as logistic regression and neural networks due to dimensionality. Dimension reduction was achieved by considering those voxels with large (both positive and negative) t-values in the training set. This left around 175 thousand voxels to consider, still too many for the common algorithms we considered. To alleviate this

problem, a method referred to as Variable Bagging was introduced. Similar to Bagging, but instead of repeatedly taking random samples of subjects and averaging the predictions, all the subjects were used and random samples repeatedly taken of the variables. The average of the individual predictions provided the overall prediction, much like a vote of a committee. Thus instead of using all 175 thousand voxels in the model, several sets of random samples containing 35 voxels were used. Since Bagging works best when there is diversity in the variables, diversity was guaranteed by choosing nearly half the voxels having negative t-values with the rest having positive t-values.

The procedure was refined by introducing the cross-validation selector to select which algorithm should be used on each individual set of random variables and thus removing the decision as to what algorithm to use.

6.2 Advantages

First and foremost, variable bagging allows models with dimensionality issues to be considered in the modeling process. Thus, instead of using the entire brain image of 2,122,945 voxels, variable bagging allows for the averaging of several models using just 35 voxels each as variables.

Taking a small random sample of the variables from the much larger set of voxels reduces the probability of over fitting the data to an absolute minimum. It also reduces bias which provides confidence in the model chosen.

For the aggregate model, variable bagging boosted the success rates of the individual models. Any increase in success is the result of the diversity within the individual models. The greater the diversity, the more favorable the results.

The cross-validation selector has the feature of ridding the investigator of having to make the decision of which algorithm to be used in modeling. On the individual models, cross-validation can do no better than the best model in consideration; however, it is possible to perform better than any one algorithm available in the aggregate model. As the examples illustrate, the choices made by the cross-validation selector do not necessarily provide the best algorithm. However, it will produce results that are, on average, very close to the best. Any minor reduction in accuracy is a small price to pay for the insurance that nearly eliminates the chance of a random guess about which algorithm to use.

The procedure reduces the effort of dimension selection, increases overall success rates, and atomizes the selection of algorithms used in the modeling process.

6.3 Disadvantages

Obviously, if the underlying distribution were known in advance then the single more accurate model could be produced instead of the model produced by the cross-validation selector but this is typically not the case especially in real world situations.

The cross-validation selector does rid the investigator of the burden of choosing the proper algorithm, but in so doing, it is vital that the investigator must know a variety

of data mining algorithms which would be appropriate for the modeling situation. Additionally, the investigator needs to know the format of the outputs of each algorithm so that the format the outputs are of a uniform manor.

To guarantee diversity, voxels were classified into two sets, positive t-values and negative t-values. Since bagging procedures rely of diversity to improve accuracy, it would make sense to identify more sets of diversity. But, how many is too many? By increasing the number of these diversity sets, the size of these sets themselves become smaller. A smaller sample size decreases randomness which could create an artifact which would cause over fitting. Too many diversity sets could cause, ironically, less diversity. Many diversity sets would cause the voxels to fall into a more particular classification. Thus, even though the sets are diverse, the voxels within each set are not causing the different models which represent all the sets to be similar. Variable bagging would be of little use in such cases.

6.4 Ideas for Improvement and for the Future

It is suspected that better voxel-based preprocessing method might help in the final outcome. Admittedly my knowledge of preprocessing is limited and though it is my belief that the SPM preprocessing program was the best free-source available, I still have to wonder if there could be a program available that would be better for voxel-based imagery.

The results of this paper were based on only the gray matter of the brain. White matter and spinal fluid amounts were not considered. To increase diversity without

decreasing randomness it would seem advisable to consider both the white matter and spinal fluid.

Of the data mining algorithms available, this paper only considered four: logistic regression, neural networks with one internal node, support vector machines, and decision trees. The inclusion of more algorithms would certainly make better use of the cross-validation selector. Other algorithms might include, but not limited to: the lasso, principal components, neural networks with multiple hidden nodes and other learning type algorithms.

Other investigation for the future would be to test whether it is better to include all variables of the different diversity sets in each individual model and applying the bagging process on these models or having the models represent each diversity group separately then using the bagging process on the diverse models.

REFERENCES

Ashburner, J. (2007). A fast diffeomorphic image registration algorithm. *Neuroimage, 38*(1), 95-113. doi: 10.1016/j.neuroimage.2007.07.007

Ashburner, J., & Friston, K. J. (2005). Unified segmentation. [Research Support, Non-U.S. Gov't]. *Neuroimage, 26*(3), 839-851. doi: 10.1016/j.neuroimage.2005.02.018

Bell, R. M., Koren, Y., & Volinsky, C. (2008). The BellKor 2008 Solution to the Netflix Priz. Retrieved from http://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf

Billingsley, P. (1995). *Probability and measure* (3rd ed.). New York: Wiley.

Breiman, L. (1984). *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group.

Breiman, L. (1994). Bagging Predictors. *Technical Report, No. 421*. Retrieved from Department of Statistics website:

Breiman, L. (1996). Stacked regression. *Machine learning, 24*, 49-64.

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression. The X random case. *International Statistical Review, 60*(3), 291 - 319.

Brett, M., Leff, A. P., Rorden, C., & Ashburner, J. (2001). Spatial normalization of brain images with focal lesions using cost function masking. *Neuroimage, 14*(2), 486-500. doi: 10.1006/nimg.2001.0845

Brieiman, L. (1996). Heuristics of instability and stabilization in model selection. *Annals of Statistics, 24*(6), 2350 - 2383.

Cawley, G. C., & Talbot, N. L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research, 11*, 2079-2107.

Chambers, J. M., & Hastie, T. (1993). *Statistical models in S*. New York: Chapman & Hall.

Collins, D. L., Neelin, P., Peters, T. M., & Evans, A. C. (1994). Automatic 3-D inter-subject registration of MR volumetric data in standard Talairach space. *JCAT, 18*(2), 192-205.

Dawbarn, D. & Allen, S.J. (2007). *Neurobiology of Alzheimer's disease* (3rd ed.). Oxford; New York: Oxford University Press.

Dudoit, S., van der Laan, (2003). Asymptotics of Cross-Validated Risk Estimation in Estimator Selection and Performance Assessment. *U. C. Berkeley Division of Biostatistics Working Paper Series, Paper 126*. Retrieved from http://bioiostats.bepress.com/ucbbiostat/paper126

Everitt, B. (2002). *The Cambridge dictionary of statistics* (2nd ed.). Cambridge, UK ; New York: Cambridge University Press.

Fox, P. T., Perlmutter, J. S., & Raichle, M. E. (1985). A stereotactic method of anatomical localization for positron emission tomography. *J Comput Assist Tomogr, 9*(1), 141-153.

Friedland, R. P. (2010) Professor of Neurology, University of Louisville School of Medicine, personal conversation.

Friedland, R. P., & Wilcock, G. (2000). Dementia. In J. G. Evans & T. F. W. O. U. Press (Eds.), *Oxford Textbook of Geriatric Medicine* (2nd ed., pp. 922-932).

Goedert, M., Spillantini, M. G. (2000). Tau mutations in frontotemporal dementia FTDT-17 and their relevance for Alzheimer's disease. *Biochimica et Biophysica*, 1502, 110 – 121.

Gomez-Isla, T., West, H. L., Rebeck, G. W., Harr, S. D., Growdon, J. H., Locascio, J. J., Hyman, B. T. (1996). Clinical and pathological correlates of apolipoprotein E epsilon 4 in Alzheimer's disease. [Research Support, U.S. Gov't, P.H.S.]. *Ann Neurol, 39*(1), 62-70. doi: 10.1002/ana.410390110

Györfi, L., Kohler, M., Krzyżak, A., & Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. New York: Springer,.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning : data mining, inference, and prediction* (2nd ed.). New York: Springer.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association, 58*, 13 - 58.

Kantardzic, M. (2003). *Data mining : concepts, models, methods, and algorithms*. Hoboken, NJ: Wiley-Interscience : IEEE Press.

Kim, J., Avants, B., Patel, S., & Whyte, J. (2008). Spatial normalization of injured brains for neuroimaging research: An illustrative introduction of available options. *NCRRN Methodology Papers*. Retrieved from http://www.mrri.org

Laboratory of Neuro Imaging (LONI), www.loni.usc.edu.

Magnin, B., Mesrob, L., Kinkingnehun, S., Pelegrini-Issac, M., Colliot, O., Sarazin, M., Benali, H. (2009). Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI. *Neuroradiology, 51*(2), 73-83. doi: 10.1007/s00234-008-0463-x

Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., & Buckner, R. L. (2007). Cross-sectional MRI data in young, Middle aged, nondemented, and demented older adults. *Journal of Congnitive Neuroscience, 19*(9), 1498-1507.

Mazziotta, J. C., Toga, A. W., Evans, A., Fox, P., & Lancaster, J. (1995). A probabilistic atlas of the human brain: theory and rationale for its development. The International Consortium for Brain Mapping (ICBM). *Neuroimage, 2*(2), 89-101.

McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in neural activity. *Bulletin of Mathematical Biophysics, 5*, 115-133.

Nowak, R. (2007). *Lecture 7: Chernoff's Bound and Hoeffding's Inequality*
Retrieved from http://nowak.ece.wisc.edu/SLT07/lecture7.pdf

Office, G. A. (1998). *Alzheimer's Diease; Estimates of Prevalence in the United States.* (GAO/HEHS-98-16). Retrieved from www.gao.gov.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*: Morgan Kaufmann Publishers.

Ripley, B. (2013). Package 'tree'. Retrieved from cran.r-project.org/web/packages/tree/tree.pdf

Rousseeuw, P., & Leroy, A. (1987). Robust regression and Outlier Detection. *Wiley Series in Probability and Mathematical Statistics.*

Royden, H. L. (1988). *Real analysis* (3rd ed.). New York, London: Macmillan;

Rumelhart, D. E., & McClelland, J. L. (1986). Paralled Distributed Processing: Explorations in the Microstructure of Cognition. *Foundations* (Vol. 1). Cambridge, MA: MIT Press.

Salas-Gonzalez, D., Gorriz, J. M., Ramirez, J., Lopez, M., Alvarez, I., Segovia, F., . . . Puntonet, C. G. (2010). Computer-aided diagnosis of Alzheimer's disease using support vector machines and classification trees. *Physics in Medicine and Biology, 55*, 2807 - 2817. doi: 10.1088/0031-9155/55/10/002

Schnabel, J. (2010). Amyloid-beta 'oligomers' may be link to Alzheimer's dementia, *The DANA Foundation*, www.dana.org

Shalizi, C. (2009). Classification and Regression Trees. www.stat.cmu.edu/~cshalizi/350/lectures/22/lecture-22.pdf

Sled, J. G., & Pike, G. B. (1998). Standing-wave and RF penetration artifacts caused by elliptic geometry: an electrodynamic analysis of MRI. *IEEE Trans Med Imaging, 17*(4), 653-662. doi: 10.1109/42.730409

Taylor, C. What Is Markov's Inequality? *About.com Statistics*.

Therneau, T. M., Atkinson, E. J., & Foundation, M. (2013). An Introduction to Recursive Partitioning Using the RPART Routines. *CRAN R Project*. Retrieved from http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

Tosun-Turgut, D. (2012, June 29, 2012) Assistant Professor at University of California San Francisco (UCSF) School of Medicine, personal correspondence.

van der Laan, M. J., & Dudoit, S. (2003). Unified Cross-Validation Methodology For Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples. *U. C. Berkeley Division of Biostatistics Working Paper Series, Paper 130*. Retrieved from http://biostats.bepress.com/ucbbiostat/paper130 website:

van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super Learner. *U. C. Berkeley Division of Biostatistics Working Paper Series, Paper 222*. Retrieved from http://bioiostats.bepress.com/ucbbiostat/paper222

van der Laan, M. J., & Rose, S. (2011). *Targeted learning causal inference for observational and experimental data*. New York: Springer,.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Vincent, T., Tenorio, L., & Walkin, M. Concentration of Measure: fundamentals and tools. www.stat.rice.edu/~jrojo/PASI/lectures/TyronCMarticle.pdf

CURRICULUM VITAE
2014

Name: **Michael W. Godbey, Ph.D.**

Current Position:

Visiting Assistant Professor of Mathematics
School of Natural Sciences
Indiana University Southeast

Home Address:

4969 Winding Spring Circle, Louisville, KY 40245

Business Address:

School of Natural Sciences
Indiana University Southeast
4201 Grant Line Road
New Albany, IN 47150
Telephone: 812-941-2422
Fax: 812-941-2637
Email: mwgodbey@ius.edu

Place of Birth:

Huntington, West Virginia

Marital Status:

Married. Wife, Nancy L.

Citizenship:

U.S.A.

Undergraduate Studies:

West Virginia University, Morgantown, WV
B.A., May 1982; Major: Mathematics,

Graduate Studies:

Marshall University, Huntington, WV
M.A., December 1990 (Mathematics)

University of Louisville, Louisville, KY
Ph.D., December 2014 (Applied and Industrial Mathematics)

Teaching and Research Positions:

Adjunct Instructor, Ohio University at Ironton, Department of Mathematics, 1997 – 1998

Term Instructor, Marshall University, Department of Mathematics, 1998 – 2003

Term Instructor, Rio Grande University and Community College, Department of Mathematics 2003 – 2004

Term Instructor, Marshall Technical and Community College, Department of Mathematics 2004 – 2006

Graduate Teaching Assistant, University of Louisville, Department of Mathematics, 2006 – 2013

Term Instructor, University of Louisville, Department of Mathematics, 2013 – 2014

Visiting Assistant Professor, Indiana University Southeast, School of Natural Sciences, 2014 – Present

Undergraduate Teaching Experience:

Contemporary Mathematics
Pre-Algebra
College Algebra
Trigonometry
Elementary Statistics
Business Calculus
Pre-Calculus
Calculus I.

Graduate Teaching Experience:

None

Honors and Awards:

The Graduate Dean's Citation