

# Alzheimer's Disease and Support Vector Machines: An Introduction to Machine Learning

James O. Hortle<sup>1</sup>

## Abstract

An introduction to machine learning is provided, wherein the basic theory of a major concept, *support vector machines*, is developed. This idea is then applied to data from two studies on Alzheimer's disease. The results indicate that support vector machines can be an effective model for the analysis of MRI data for this disease, potentially aiding in diagnosis.

## A Introduction

Suppose that you have some data and that you want to know more about it. For example, consider a collection of photos saved on your smart phone. If you are a dog lover, it might be nice to tag each photo so that you know which ones contain dogs. In that way, you can easily search for all your canine photos. But, if you are an avid fan of furry friends, perhaps you have thousands or millions of photos. Going through them all would be tedious and expensive. If, instead, the task could be automated, then you would save a lot of time and energy.

Similarly, researchers and those working with data are interested in ways that we can use mathematics to help us solve problems such as detecting cancer, predicting the weather or even classifying photos as above. The example in the preceding paragraph is called a *classification task* and is a common problem. Keep it in mind as we will use it to explain a few key concepts.

This article serves as an introduction to the basics of what is commonly known as *machine learning*. Machine learning has its roots in statistics with a long history that can be traced back to Fisher's method for pattern recognition in so-called "taxonomic problems" [7]. With the recent advent of powerful computers, exceedingly large datasets can now be analysed. Historically, machine learning has been used to beat human players at board games, as was the case with Deep Blue [9]. Recently, machine learning techniques have overcome certain watersheds, which indicate the maturity of the techniques now being used [21]. Excitingly, machine learning is now being employed in solving real-world problems in such domains as medicine [19], finance [8] and meteorology [15].

---

<sup>1</sup>James O. Hortle is a software engineer currently working in AI and robotics in Tokyo, Japan. He holds a Bachelor of Arts and a Bachelor of Science (Advanced Mathematics) with Honours in Applied Mathematics from UNSW Sydney.

In order to make the presentation of machine learning more concrete, we will focus on and give an overview of one machine learning method, namely *support vector machines* (SVMs), and then apply it to data from a study on Alzheimer’s disease. This is well-suited for classification tasks and, for example, is quite good at classifying hand-written digits, as might be found in the postcode of a letter or on a cheque, and has been employed in postal services and banks.

## B The basics of support vector machines

### 2.1 Preparing the data

Suppose that we have a set of data, which we will call a *data set*. This data set can be anything, but typically it will be numerical in some way. This data set will be composed of *data points* which are individual entries. For example, each photo in the example above would be a data point, while the data set would be the whole collection of photos. If you have  $N$  photos, then you might call the first one  $X_1$ , the second one  $X_2$ , and so on, until you have called the last one  $X_N$ .

Now, suppose that you have already looked at some of your photos before giving up and have succeeded in tagging some as either containing a dog or not. Since you have spent so much time classifying these photos into either `dog` or `not dog`, it would be a shame to throw the tags away, so we will call your tags *labels*. Note that there are only two options, so these are *binary labels* and we might instead represent them by the numbers 1 (for positive) and  $-1$  (for negative).<sup>2</sup>

We are now ready to define our data set more rigorously. We will let  $X_1, \dots, X_N$  be data points with corresponding labels  $Y_1, \dots, Y_N \in \{-1, 1\}$ . So, the set of *ordered pairs*  $(X_i, Y_i)$  is the data that we will be working with. In the rest of this article, we must assume that there will always be at least one positive data point (that is, some  $X_i$  with  $Y_i = 1$ ) and one negative data point (that is, some  $X_j$  with  $Y_j = -1$ ). If our data set has only positive or only negative data points, or no data points at all, then it is called *trivial*.

For automating your classification task, we will most likely be using a computer. Hence, we need to transform our data into something that a computer can work with; in other words, something numerical. In our dog example, one way to change a picture into a number would be to measure the brightness of each pixel and add these values to produce a number. There are many other (probably better) ways. For the rest of this section, let us consider something more visual. We will let each data point  $X_i$  be a point in the plane. That is,  $X_i, \dots, X_N \in \mathbb{R}^2$ . We will write  $X_i = (x_i, y_i)$  when we need to look at the  $x$ - and  $y$ -coordinates of  $X_i$  individually. It is a good idea to look at Figure 1 while reading the rest of this section.

---

<sup>2</sup>You might be aware that *binary* often refers to the use of the two numbers 0 and 1. In our case, binary just means that we have two numbers to choose from. It will become apparent later why  $-1$  is a better choice than 0.

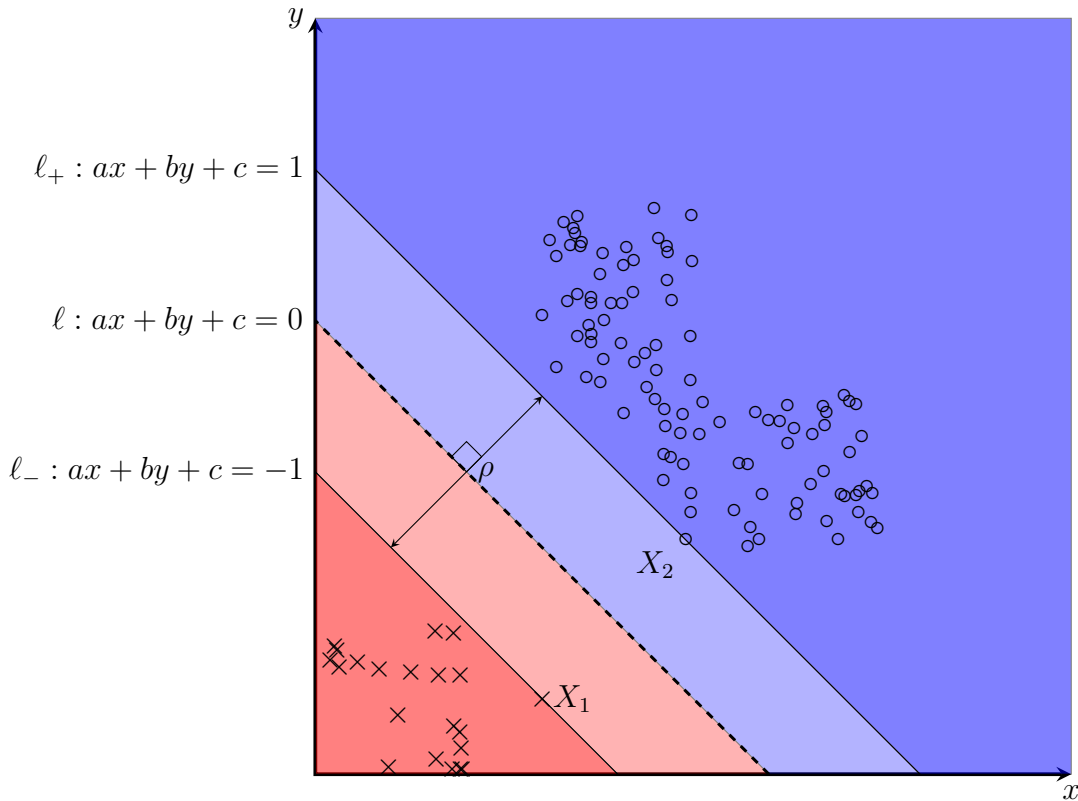


Figure 1: Linearly separable data separated by a straight line (dashed). Crosses represent negative cases, while circles represent positive cases. The points  $X_1$  and  $X_2$  are boundary cases, lying on the solid lines. The line given classifies points in the red area as negative and in the blue area as positive.

## 2.2 Separating the data with a best line

Since these points lie on the plane, we can draw a line through them. But what line should we draw? We would like something that separates the data such that all the points with negative labels ( $Y_i = -1$ ) are on one side of the line and all the points with positive labels ( $Y_i = 1$ ) are on the other. If all the negative data points are clustered together, and all the positive data points are as well, then there may be one or many lines that would separate the data nicely. In this case, we call the data set *linearly separable*. For the rest of this section, you can suppose that our data set is always linearly separable. So, now we need to find the “best” line.

Although the concept of “best” can sometimes be ambiguous, in this article we will define it to mean “the line that defines the biggest gap between the negative data points and the positive data points”. Here, the distance between any two points in the plane  $X_1 = (x_1, y_1)$  and  $X_2 = (x_2, y_2)$  is defined as

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

This best line, which we will call  $\ell$ , can be represented by the equation  $ax + by + c = 0$ ,

where  $a$ ,  $b$  and  $c$  are some values that we will figure out later. We will further define two more lines,  $\ell_+ : ax + by + c = 1$  and  $\ell_- : ax + by + c = -1$ . These two lines will be called *gutters* because they look like the gutters on a road, with the first line we defined being the centre of the road.

Now, we will give some characteristics of these lines based on our assumption that the data are linearly separable. Firstly, suppose that  $X_i = (x_i, y_i)$  is a data point with corresponding label  $Y_i$ . Then, for our values of  $a$ ,  $b$  and  $c$ ,

- (a) all data points with positive labels  $Y_i = 1$  will have  $ax_i + by_i + c \geq 1$ ,
- (b) all data points with negative labels  $Y_i = -1$  will have  $ax_i + by_i + c \leq -1$ ,
- (c) the data points for which  $ax_i + by_i + c = 1$  or for which  $ax_i + by_i + c = -1$  are called *support vectors*<sup>3</sup> and there will be at least one positive one and one negative one.

The first two facts are true by assumption, while we give a short proof for the third one below. In fact, the characteristics that we just gave are a rigorous definition of what it means for a data set to be linearly separable. We can go one step further by saying that a data set is linearly separable if and only if, for all  $i = 1, \dots, N$ ,

$$Y_i(ax_i + by_i + c) \geq 1.$$

*Proof of point 3.* To see why there will always be at least one positive and at least one negative support vector, we will consider a very simple data set with only two data points, one negative and one positive. This is the smallest non-trivial data set. Since it is always possible to draw a straight line between any two points, this data set is clearly linearly separable. Suppose that we have found a best line and gutters. Suppose further that there are no negative support vectors. Then the negative gutter does not pass through the negative data point. However, by assumption the negative data point lies below the gutter. Hence, the gutter can be moved down until it reaches this point. This implies that our best line from before was in fact not the best line, as we have now increased the gap. Therefore, our assumption that there are no negative support vectors contradicted our assumption that we had found the best line. We can conclude that the negative data point must be a support vector. By a similar argument, the positive data point is also a support vector. Hence, there is always at least one positive support vector and always at least one negative support vector. This idea is easily extended to larger data sets.  $\square$

---

<sup>3</sup>The word *vector* refers to something like  $\mathbf{x} = (x_1, \dots, x_n)$ , where each of  $x_1, \dots, x_n$  are numbers. Hence, a point in the plane like our data point  $X_i$  is a vector where  $n = 2$ . We call these points *support vectors* since they define the line  $\ell$ ; they “support” it like pillars support a roof.

Our next question is “How do we find the values of  $a$ ,  $b$  and  $c$ ?”. Recall when we were discussing how to measure points in the plane. We can use this method to find the distance between the two lines  $\ell_+$  and  $\ell_-$ . Let us call this distance  $\rho$ .

We will begin by writing our lines in standard form:

$$\begin{aligned}\ell_+ : ax + by + c - 1 &= 0 \\ \ell_- : ax + by + c + 1 &= 0.\end{aligned}$$

Since these two lines are parallel, the distance between them is the distance between their intersection points with a perpendicular line. That is, the distance between their points of intersection with the line  $ay - bx = 0$ . You can verify that this line is indeed perpendicular to  $\ell_+$  and  $\ell_-$  by writing all the equations in the form  $y = mx + b$  and checking that the product of their gradients is equal to  $-1$ .

We can solve the two systems of linear equations to obtain the two intersection points as

$$\begin{aligned}A &= \left( \frac{a(1-c)}{a^2+b^2}, \frac{b(1-c)}{a^2+b^2} \right), \\ B &= \left( \frac{-a(1+c)}{a^2+b^2}, \frac{-b(1+c)}{a^2+b^2} \right).\end{aligned}$$

Then

$$\begin{aligned}\rho &= d(A, B) \\ &= \sqrt{\left( \frac{a(1-c) + a(1+c)}{a^2+b^2} \right)^2 + \left( \frac{b(1-c) + b(1+c)}{a^2+b^2} \right)^2} \\ &= \sqrt{\left( \frac{2a}{a^2+b^2} \right)^2 + \left( \frac{2b}{a^2+b^2} \right)^2} \\ &= \sqrt{\frac{4(a^2+b^2)}{(a^2+b^2)^2}} \\ &= \frac{2}{\sqrt{a^2+b^2}}.\end{aligned}$$

Recall that to find the best line, we need to make size of the gap, represented by  $\rho$ , as large as possible. Hence, we need to find  $a$  and  $b$  such that  $\sqrt{a^2+b^2}$  is as small as possible while maintaining that  $Y_i(ax_i + by_i + c) \geq 1$  for all  $i = 1, \dots, N$ .

Doing this in general can be quite complicated, so we will skip the details. What you should take away is that it is possible. In fact, this type of problem is known as a *quadratic programming problem* and many calculation and software packages have built-in methods (algorithms) for solving it.

## 2.3 Classifying new data

Great; we have successfully found a line that divides the data that you have so laboriously labelled. Now, how do we use the lines we have defined to classify new data, data that the algorithm has not seen before? Simple: you take the new data point  $X_{N+1} = (x_{N+1}, y_{N+1})$  and then look at the sign of  $ax_{N+1} + by_{N+1} + c$ . If

$$ax_{N+1} + by_{N+1} + c < 0$$

then the point will be classified as negative. If, on the other hand,

$$ax_{N+1} + by_{N+1} + c > 0$$

then it will be classified as positive. If, on the third hand,

$$ax_{N+1} + by_{N+1} + c = 0$$

then you have found a data point that the algorithm cannot reliably classify! We call this sort of data point an *outlier*.

The careful reader may have already noticed that the first two of the three equations above are not exactly the same as the equations that we gave for  $\ell_+$  and  $\ell_-$  or when we were characterising a linearly separable data set. Why? Because the line  $\ell$  is in fact the one that does the classifying and the two lines  $\ell_+$  and  $\ell_-$  are there to help us create  $\ell$ . When we try and classify a new data point, we see where it lies in the plane with respect to  $\ell$ ; is it above or below  $\ell$ ?

## 2.4 What's in a name?

So, what exactly are we referring to when we say "SVM"? SVM actually refers to the idea or concept of being able to find a best line that separates positive and negative data and being able to use this line to classify new data. The actual set of methods that we use to do this (*i.e.*, methods for maximising  $\rho$  and calculating the sign of  $ax_{N+1} + by_{N+1} + c$ ) along with this idea of a separating line are often referred to as the *SVM algorithm*.

The set of steps that you write down when you tell a computer (or anything else that can do "computations") what to do is called an *algorithm*. Hence, SVMs can also be considered to be a conceptual model for machine learning with associated algorithms for its implementation.

In reality, this distinction is often ignored and you will most likely see people refer to SVMs as an algorithm or set of algorithms, a technique or an idea, concept or model.

In practice, our data are often not 2-dimensional. Instead, consider some more general  $p$ -dimensional data. That is,  $X_i = (x_1, \dots, x_p)$  for some  $p \geq 2$ . Then, the SVM is the idea of creating a  $(p-1)$ -dimensional *separating hyperplane* that we can use to classify new data, analogously to the above. Because of this, SVMs are part of a broader class of algorithms or ideas called "linear classifiers" which all attempt to classify data using the idea of a separating line.

## 2.5 Further problems

Recall the concept of outliers introduced above. We would like to modify our algorithm so that it can still work. When Corinna Cortes and Vladimir Vapnik published a modern version of the technique in 1995 [5], they developed so-called *soft-margin* SVMs, which allow a few of those outliers to pollute the data set, while still generating a relatively good line that separates the data. The next problem was dealing with sets for which the data may be too complicated. Although the details are also quite complicated, this problem is overcome by transforming the data into some higher dimensional space via what is called the *kernel trick*, constructing a *separating hyperplane* in that higher dimensional space (the higher-dimensional equivalent of our best line above) and then using that separating hyperplane to classify new data in much the same way as above. For a nice illustration of this, I recommend this short video: <https://youtu.be/3liCbRZPrZA>. In the video, you can see that 2D points are projected up into 3D space, where a simple plane is then able to separate them.

# C Using SVMs to help diagnose Alzheimer's disease

## 3.1 Background on Alzheimer's disease

Alzheimer's disease (AD) is a progressive neurodegenerative disease primarily affecting the elderly. It is the most common cause of dementia [2], and is characterised by a decline in memory, language, problem-solving and other cognitive skills [2]. AD was first diagnosed and presented in 1906 by German physician Alois Alzheimer [1, 17]. Alzheimer's disease has been subject to substantial scientific enquiry, with roughly 18% of actively publishing neuroscientists having contributed at least one publication on the topic [22].

It is generally recommended that neuroimaging (CT or MRI) be performed if dementia is suspected. This is to rule out any potentially treatable injuries [13]. However, neuroimaging as part of the diagnosis of AD also provides valuable data for further analysis since it gives us information about the patient's *cerebral cortex*. The cerebral cortex is a layer surrounding the brain, which varies between 1 and 5 millimetres thick [11, p.324] [20]. Reduction in the thickness of the cerebral cortex and other associated regions has been identified in patients with AD, however, this can also occur due to other illnesses [10].

Lerch *et al.* identified nine regions of the cerebral cortex whose changes in thickness are important when it comes to diagnosing AD [14]. By using these nine important regions, we can perform an analysis of MRI scans taken from two studies on AD. We will apply our idea of a SVM to classify data from clinical trials of AD.

### 3.2 Applying the SVM model to MRI data

To begin, we need to transform the MRI data into something a computer can use. Since we are interested in the thickness of the cerebral cortices, we will use the average thickness of the regions as our numbers. Then, each data point  $X_i = (x_1^{(i)}, \dots, x_9^{(i)})$  is a vector in  $\mathbb{R}^9$ . That is,  $X_i$  lies in the 9-dimensional plane. As before, the corresponding label  $Y_i$  will be either  $-1$  or  $1$ , with  $-1$  corresponding to *controls* (i.e., patients without AD) and with  $1$  corresponding to patients who have been diagnosed as having AD.

If we assume that our data set is linearly separable, then we are now looking for the 9-dimensional equivalent of a straight line, which is an 8-dimensional surface. Let us call this surface  $L : a_1x_1 + \dots + a_9x_9 + a_{10} = 0$ . Then, as in the same way as in our initial 2-dimensional case, we must maximise

$$\rho = \frac{2}{\sqrt{a_1^2 + \dots + a_9^2}}$$

subject to  $Y_i(a_1x_1^{(i)} + \dots + a_9x_9^{(i)} + a_{10}) \geq 1$  for all  $i = 1, \dots, N$ , where we assume that our data set is linearly separable.

As I said above, real-world data is most likely quite messy and not necessarily linearly separable. In fact, the data that we used are not linearly separable. Hence, we will make use of soft-margins and the kernel trick (although you will not see any details of this here).

### 3.3 Testing the model

To see how well our algorithm performs, we need some way of testing it. To do this, we will split our data up and reserve some of it for training, called the *training data set*, and use the rest for testing, called the *testing data set*. If we repeat this many times and divide up the data differently each time, we can get an average of how well the algorithm can correctly classify MRI scans from new patients.

We can present this information quickly in the form of a table. “Training size” represents the percentage of data used to train the algorithm; the remainder was used for testing. “Testing CCR” gives the percentage of testing data correctly classified by the algorithm (i.e., the correct classification rate). It is defined simply by

$$\text{CCR} = \frac{\text{number of testing data points correctly classified}}{\text{total number of testing data points}}.$$

“Support vectors” gives the average number of support vectors, while “Percentage SV” is the percentage of training data points that were used as support vectors. The number of support vectors gives an indication of how well the SVM performed [4, 12, 23] since the more training data points were used as support vectors, the more likely *overfitting* was to have occurred. Overfitting is a problem since it means that the algorithm was too sensitive and was not able to generalise well.



Table 1: Performance of the SVM algorithm on the ADNI data set (100 scans) averaged over 100 runs.

Training size	40%	50%	60%	70%
Testing CCR	86%	89%	91%	91%
Support vectors	9	10	10	11
Percentage SV	23%	20%	17%	16%

Table 2: Performance of the SVM algorithm on the MIRIAD data set (193 scans) averaged over 100 runs.

Training size	40%	50%	60%	70%
Testing CCR	94%	95%	95%	96%
Support vectors	8	9	9	10
Percentage SV	10%	9%	7%	7%

You can see that the SVM algorithm performed very well on the data sets, with all the testing CCR results for the MIRIAD study above 90%. This is probably due to the fact that the MIRIAD data set was almost twice as large as the ADNI data set, so the algorithm had more examples to learn from. This is another important fact; the quality and size of your training data set will determine how accurate your algorithm can be.

Since the data points live in  $\mathbb{R}^9$ , I cannot show you exactly what the SVM would look like in that space. What I can show you instead is what the SVM looked like when we considered only two regions at a time (that is, we looked at only two regions so that we could plot them in the Cartesian plane). Because we used soft-margins and the kernel trick for our messy data, the lines drawn are not straight (but they would be straight in their native high-dimensional space!). Healthy controls are light blue dots; AD patients are red dots. Support vectors are circled. The purple region is the negative region (healthy) and the green region is positive (AD).

## D Conclusion

SVMs are a widely-used technique for classification of data because they are easy to implement and are reliable and accurate. We have shown that using a SVM on relatively small datasets for AD can lead to a quite good rate of correct classifications. Perhaps SVMs and/or other machine learning techniques should be adopted for regular use by clinicians in order to reduce the time between diagnosis and treatment of AD and other diseases for which medical imaging data is readily available.

Figure 2: The 2-dimensional regions defined by the SVM for the ADNI data.

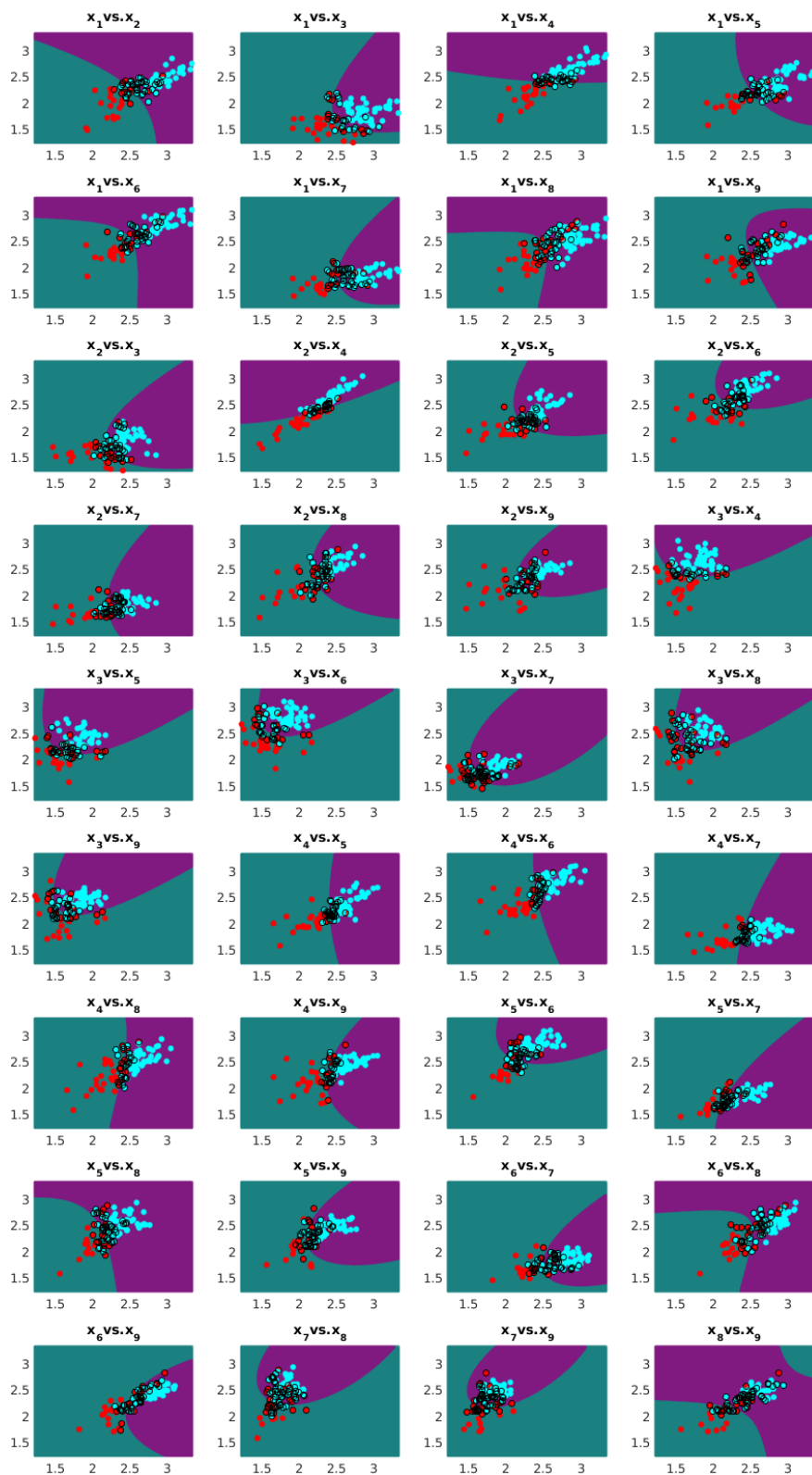
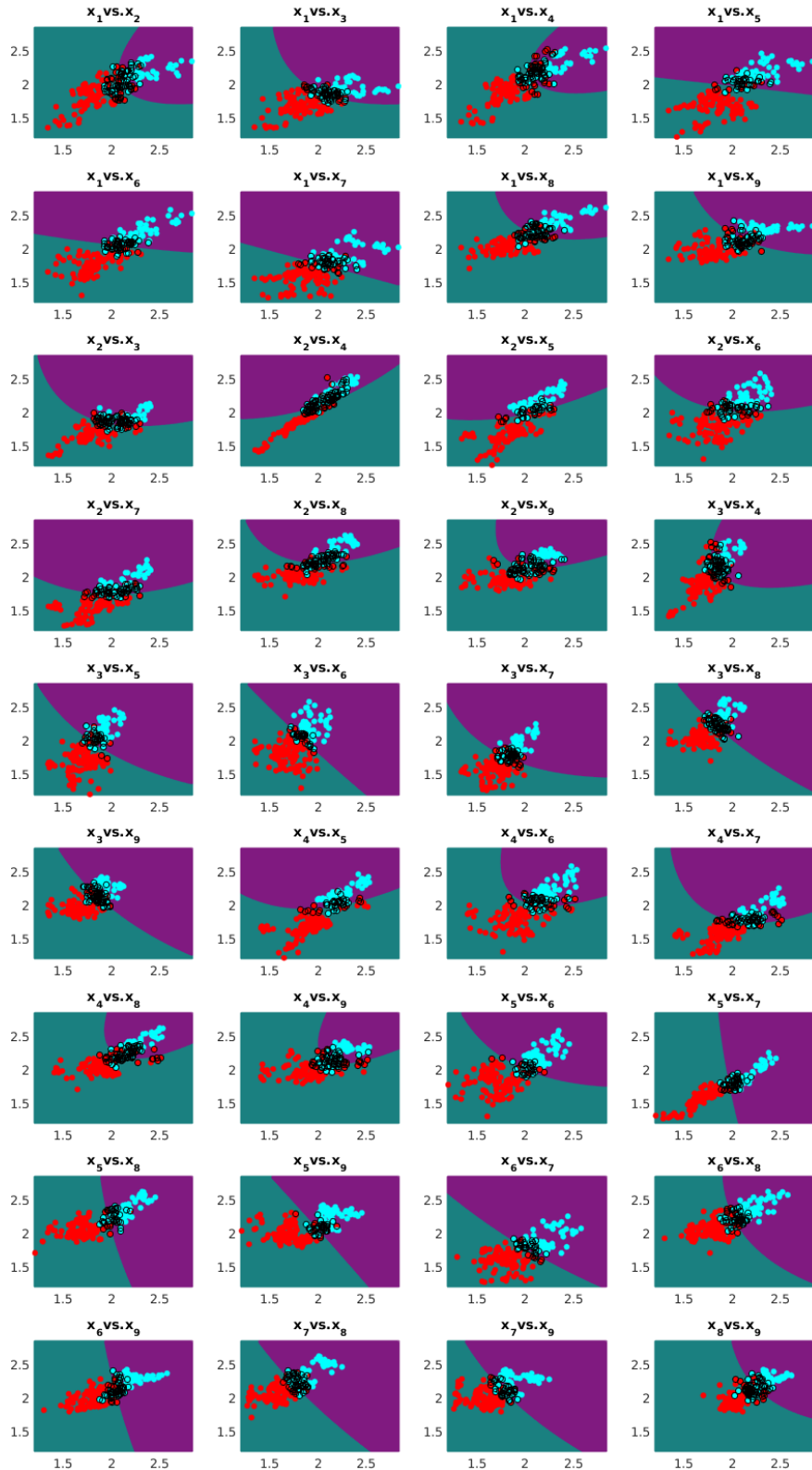


Figure 3: The 2-dimensional regions defined by the SVM for the MIRIAD data.



## Acknowledgements

The author is indebted to the University College London for making available the Minimal Interval Resonance Imaging in Alzheimer’s Disease (MIRIAD) dataset<sup>4</sup> [16] and also to the University of Southern California and the Alzheimer’s Disease Neuroimaging Initiative<sup>5</sup> (ADNI) [3,18,24] for their datasets.

Cortical thickness data were extracted using FreeSurfer by Alex Tsui at University of California, Davis<sup>6</sup> using the methods outlined in [6].

## References

- [1] A. Alzheimer, *Über einen eigenartigen, schweren Erkrankungsprozeß der Hirnrinde*, Neurologische Centralblatt **25** (1906), 1134, Eng.: *On a particular severe disease process of the cerebral cortex*.
- [2] Alzheimer’s Association, *2016 Alzheimer’s disease facts and figures*, Alzheimer’s & Dementia **12** (2016), 459–509.
- [3] Alzheimer’s Disease Neuroimaging Initiative, *Access Data*, <http://adni.loni.usc.edu/data-samples/access-data/> (2016).
- [4] P. Bartlett and J. Shawe-Taylor, *Generalization performance of support vector machines and other pattern classifiers*, Advances in Kernel methods—support vector learning (1999), 43 – 54.
- [5] C. Cortes and V. N. Vapnik, *Support-vector networks*, Machine Learning **20** (1995), no. 3, 273–297.
- [6] B. Fischl and A. M. Dale, *Measuring the thickness of the human cerebral cortex from magnetic resonance images*, Proceedings of the National Academy of Sciences of the United States of America **97** (2000), no. 20, 11050–11055.
- [7] R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics **7** (1936), no. 2, 179–188.

---

<sup>4</sup>Some of the data used in the preparation of this article were obtained from the MIRIAD database. The MIRIAD investigators did not participate in analysis or writing of this report. The MIRIAD dataset is made available through the support of the UK Alzheimer’s Society (Grant RF116). The original data collection was funded through an unrestricted educational grant from GlaxoSmithKline (Grant 6GKC).

<sup>5</sup>Other data used in preparation of this article were obtained from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) database ([adni.loni.usc.edu](http://adni.loni.usc.edu)). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: [http://adni.loni.usc.edu/wp-content/uploads/how\\_to\\_apply/ADNI\\_Acknowledgement\\_List.pdf](http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf)

<sup>6</sup>E-mail: [atsui@ucdavis.edu](mailto:atsui@ucdavis.edu).

- [8] J. B. Heaton, N. G. Polson, and J. H. Witte, *Deep learning for finance: deep portfolios*, Applied Stochastic Models in Business and Industry **33** (2017), no. 1, 3–12.
- [9] F.-H. Hsu, *Behind Deep Blue: Building the computer that defeated the world chess champion*, Princeton University Press, US, 2002.
- [10] C. R. Jack, R. C. Petersen, Y. C. Xu, S. C. Waring, P. C. O’Brien, E. G. Tangalos, G. E. Smith, R. J. Ivnik, and E. Kokmen, *Medial temporal atrophy on MRI in normal aging and very mild Alzheimer’s disease*, Neurology **49** (1997), 786–794.
- [11] E. R. Kandel, J. H. Schwartz, and T. M. Jessel, *Principles of neural science*, 4 ed., McGraw-Hill, US, 2000.
- [12] V. Kecman, T.-M. Huang, and M. Vogt, *Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance*, Support Vector Machines: Theory and Applications (L. Wang, ed.), Springer-Verlag, Berlin, 2005, pp. 255 – 274.
- [13] B. J. Kelley and R. C. Petersen, *Alzheimer’s disease and mild cognitive impairment*, Neurologic Clinics **25** (2007), 577–609.
- [14] J. P. Lerch, J. C. Pruessner, A. Zijdenbos, H. Hampel, S. J. Teipel, and A. C. Evans, *Focal decline of cortical thickness in Alzheimer’s disease identified by computational neuroanatomy*, Cerebral Cortex **15** (2005), no. 7, 995–1001.
- [15] J. Li, J. K. Ward, J. Tong, L. Collins, and G. Platt, *Machine learning for solar irradiance forecasting of photovoltaic system*, Renewable Energy **90** (2016), 542 – 553.
- [16] I. B. Malone, D. Cash, G. R. Ridgway, D. G. MacManus, S. Ourselin, N. C. Fox, and J. M. Schott, *MIRIAD — public release of a multiple time point Alzheimer’s MR imaging dataset*, NeuroImage **70** (2013), 33 – 36.
- [17] H.-J. Möller and M. B. Graeber, *The case described by Alois Alzheimer in 1911*, European Archives of Psychiatry and Clinical Neuroscience **248** (1998), 111–112.
- [18] S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. R. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett, *Ways toward an early diagnosis in Alzheimer’s disease: The Alzheimer’s Disease Neuroimaging Initiative*, Alzheimer’s & Dementia: The Journal of the Alzheimer’s Association **1** (2005), no. 1, 55–66.
- [19] S. Parodi, G. Riccardi, N. Castagnino, L. Tortolina, M. Maffei, G. Zoppoli, A. Nencioni, A. Ballestrero, and F. Patrone, *Systems medicine in oncology: Signaling network modeling and new-generation decision-support systems*, Systems Medicine (U. Schmitz and O. Wolkenhauer, eds.), Springer New York, New York, NY, 2016, pp. 181–219.

- [20] S. Shipp, *Structure and function of the cerebral cortex*, *Current Biology* **17** (2007), R443–R449.
- [21] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, *Mastering the game of Go with deep neural networks and tree search*, *Nature* **529** (2016), 484–489.
- [22] A. A. Sorensen, *Alzheimer’s disease research: Scientific productivity and impact of the top 100 investigators in the field*, *Journal of Alzheimer’s Disease* **16** (2009), no. 3, 451–465.
- [23] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.
- [24] B. T. Wyman, D. J. Harvey, K. Crawford, M. A. Bernstein, O. Carmichael, P. E. Cole, P. Crane, C. DeCarli, N. C. Fox, J. L. Gunter, D. Hill, R. J. Killiany, C. Pachai, A. J. Schwarz, N. Schuff, M. L. Senjem, J. Suhy, P. M. Thompson, M. Weiner, C. R. Jack, Jr., and the Alzheimer’s Disease Neuroimaging Initiative, *Standardization of analysis sets for reporting results from ADNI MRI data*, *Alzheimer’s & Dementia: The Journal of the Alzheimer’s Association* **9** (2013), no. 3, 332–337.