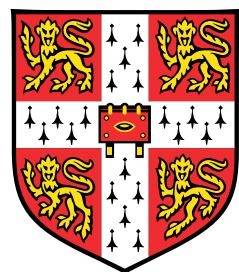


# **Repurposing Drugs for the Rapid Response to Epidemics and Pandemics**

## **Using Batch Active Learning**



**Ross Brown**

Department of Chemical Engineering and Biotechnology  
University of Cambridge

This dissertation is submitted for the degree of  
*Master of Engineering*

Robinson College

May 2022

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

I would like to dedicate this thesis to the loss of sleep never to be recovered. Its sacrifice in making this project come to fruition will never be forgotten.

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

## **Declaration**

The work described in this report is the result of my own research, unaided except as specifically acknowledged in the text, and it does not contain material that has already been used to any substantial extent for a comparable purpose. This report contains 39 pages and 9000 words (excluding this page, the title page, and the safety appendix).

Ross Brown

May 2022

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

## **Acknowledgements**

And I would like to acknowledge ...

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

## Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

# Table of contents

|  |             |
|--|-------------|
| <b>List of figures</b>                         | <b>xiii</b> |
| <b>List of tables</b>                          | <b>xv</b>   |
| <b>Nomenclature</b>                            | <b>xvii</b> |
| <b>1 Introduction</b>                          | <b>1</b>    |
| <b>2 Previous Work</b>                         | <b>3</b>    |
| 2.1 Active Learning . . . . .                  | 4           |
| 2.1.1 Current Data . . . . .                   | 5           |
| 2.1.2 Estimated Future . . . . .               | 13          |
| 2.2 Batch Active Learning . . . . .            | 13          |
| 2.3 Drug Data for Machine Learning . . . . .   | 17          |
| 2.3.1 Physical Properties . . . . .            | 17          |
| 2.3.2 Fingerprints . . . . .                   | 17          |
| <b>3 Methodology</b>                           | <b>19</b>   |
| 3.1 Outline . . . . .                          | 19          |
| 3.1.1 Data . . . . .                           | 19          |
| 3.1.2 Custom Algorithms . . . . .              | 19          |
| 3.2 Computational Methodology . . . . .        | 20          |
| 3.2.1 Integral Functions and Classes . . . . . | 20          |
| 3.2.2 Custom Base Functions . . . . .          | 21          |
| 3.2.3 Active Learning Algorithms . . . . .     | 23          |
| 3.2.4 Training Framework . . . . .             | 25          |

|                                      |           |
|--------------------------------------|-----------|
| <b>4 Results</b>                     | <b>27</b> |
| 4.1 Non-Parametric . . . . .         | 27        |
| 4.1.1 Monte Carlo . . . . .          | 27        |
| 4.1.2 Greedy . . . . .               | 27        |
| 4.1.3 ROD Sampling . . . . .         | 27        |
| 4.2 Parametric . . . . .             | 28        |
| 4.2.1 Hotspot Clusters . . . . .     | 28        |
| 4.2.2 ROD with Greed . . . . .       | 29        |
| 4.3 Special Case: COVID-19 . . . . . | 29        |
| <b>5 Discussion</b>                  | <b>33</b> |
| 5.1 Non-Parametric . . . . .         | 33        |
| 5.2 Parametric . . . . .             | 33        |
| <b>6 Conclusion</b>                  | <b>37</b> |
| <b>References</b>                    | <b>39</b> |

# List of figures

|     |   |    |
|-----|---|----|
| 2.1 | Example Dataset for Representation of Ideas . . . . . | 4  |
| 2.2 | Uncertainty Sampling Demonstration . . . . .          | 6  |
| 2.3 | Broad-Base Sampling Illustration . . . . .            | 9  |
| 2.5 | Cluster Hotspot Sampling Illustration . . . . .       | 12 |
| 2.6 | Batch Uncertainty Sampling . . . . .                  | 14 |
| 2.7 | Batch Broad-Base Sampling . . . . .                   | 15 |
| 2.8 | Batch Cluster Sampling . . . . .                      | 16 |
| 3.1 | Representation of the split function . . . . .        | 22 |
| 4.1 | Monte Carlo . . . . .                                 | 28 |
| 4.2 | Greedy . . . . .                                      | 29 |
| 4.3 | ROD . . . . .   | 30 |
| 5.1 | Non-parametric comparison . . . . .                   | 34 |

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

# List of tables

|   |    |
|---|----|
| 3.1 Schema for the Model Class. . . . . | 21 |
|---|----|

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

# Nomenclature

## Chapter 2

|                  |   |
|------------------|---|
| $N$              | Number of features/dimensions of $x$                            |
| $x$              | Data points where $x = \{x_0, x_1, \dots, x_{N-1}\}$            |
| $y$              | Labels for the dataset where $y = \{y_0, y_1, \dots, y_{N-1}\}$ |
| $\text{AC}_{50}$ | Half maximal effective molar concentration                      |
| $\text{EC}_{50}$ | Half maximal effective molar concentration                      |
| $\text{IC}_{50}$ | Half maximal inhibitory molar concentration                     |
| $\text{Ki}$      | Half maximal molar concentration for half receptor occupancy    |
| $\text{LD}_{50}$ | Median lethal dose  |
| $\text{XC}_{50}$ | Half maximal effective or inhibitory molar concentration        |

## Chapter 3

|                      |  |
|----------------------|--|
| $X_{\text{test}}$    | Datasets used to provide a score for the algorithms                      |
| $X_{\text{train}}$   | Datasets used for training the algorithms                                |
| $x_{\text{known}}$   | Data points where the true label is available to the algorithms used     |
| $x_{\text{unknown}}$ | Data points where the true label is not available to the algorithms used |
| $y_{\text{known}}$   | True labels available to the algorithms used                             |
| $y_{\text{unknown}}$ | True labels unavailable to the algorithms used                           |
| $n$                  | The number of samples per iteration                                      |

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

# Chapter 1

## Introduction

In 2019, human civilisation was on the precipice of a natural disaster: SARS-CoV-2 (COVID-19). First reported to the World Health Organization (WHO) on December 31st, it became officially recognised as a pandemic on March 11th 2020. As of the writing of this passage, 515 million cases and 18 million excess deaths have been recorded [Wan+22; Wor22]. This, however, is not the first time a pandemic has occurred, with the Black Death infamously killing a third of Europe's population and the Spanish Flu causing mass death throughout the world. Likewise, it is unlikely to be the last.

When such a disaster does strike, it is important to react quickly. Vaccinations are allowed accelerated timelines in development cutting development from years to month, and trials into potential treatments are encouraged with haste. Within the first stages of the pandemic, drugs such as hydroxychloroquine and bleach were amongst several that were promoted by the President of the United States of America demonstrating the desperation in finding therapeutic drugs against the virus.

In order to facilitate a more robust approach to finding treatments, the FDA instigated the Coronavirus Treatment Acceleration Program (CTAP) [Cen22]. Here, over 690 drugs are in the development stage with over 450 clinical trials underway to investigate the effectiveness, with 15 drugs currently authorised for emergency use and only one drug, remdesivir, with approval for use against COVID-19 [Cen22]. Indeed, remdesivir is an important case. This drug was developed initially for hepatitis-c before being used for several other conditions until finally being used for COVID-19 [Par+20]. This demonstrates how a discovered drug can be repurposed for new diseases providing a cheap means of drug "redevelopment".

Investigations into pre-existing drugs, however, were slow and largely carried out through labour intensive mechanisms without a rational methodical testing regime. This added time to finding treatments to COVID-19. Time many did not have. A hopeful fulfilment of this

1 problem is the "Robot Scientist"; a fully automated combination of software and hardware  
2 aimed at solving this problem. For the software side, a form of reinforcement machine  
3 learning is proposed: active learning. This is a methodology suited to fields with large  
4 amounts of unlabelled data which is difficult to label. In this case, the labelling requires  
5 chemical and biological experimentation costing both time and money. By using Active  
6 Learning, as few drugs as possible will be labelled within this stage to accurately predict  
7 the best drugs for the given problem. From here, clinical trials may begin. Additionally,  
8 due to the large importance of time, many drugs may be tested in parallel. This presents an  
9 additional problem: how does one set up a testing scheme for batches.

10 Thus, the purpose of this thesis. To present an algorithm which may be used to discover  
11 effective drugs within a short period of time. Additionally, a framework will be developed  
12 that allows for different algorithms to be rigorously compared to each other for increased  
13 robustness.

# Chapter 2

## Previous Work

Scores displayed in examples have been based on the entire data set. Although this usually leads to data leakage within machine learning, this is not a concern here as the true comparison comes from testing *intelligent* vs *dumb* learning methods. In both of these cases, the model is kept identical, but the selection process is not. The baseline simply takes the first  $n$  entries from the data set, with the *intelligent* method described where required. A simple function has been used to present data as a means of demonstrating these models, with  $x$  having two dimensions. The function for  $y$  is shown in 2.1 and displayed graphically in Figure 2.1.

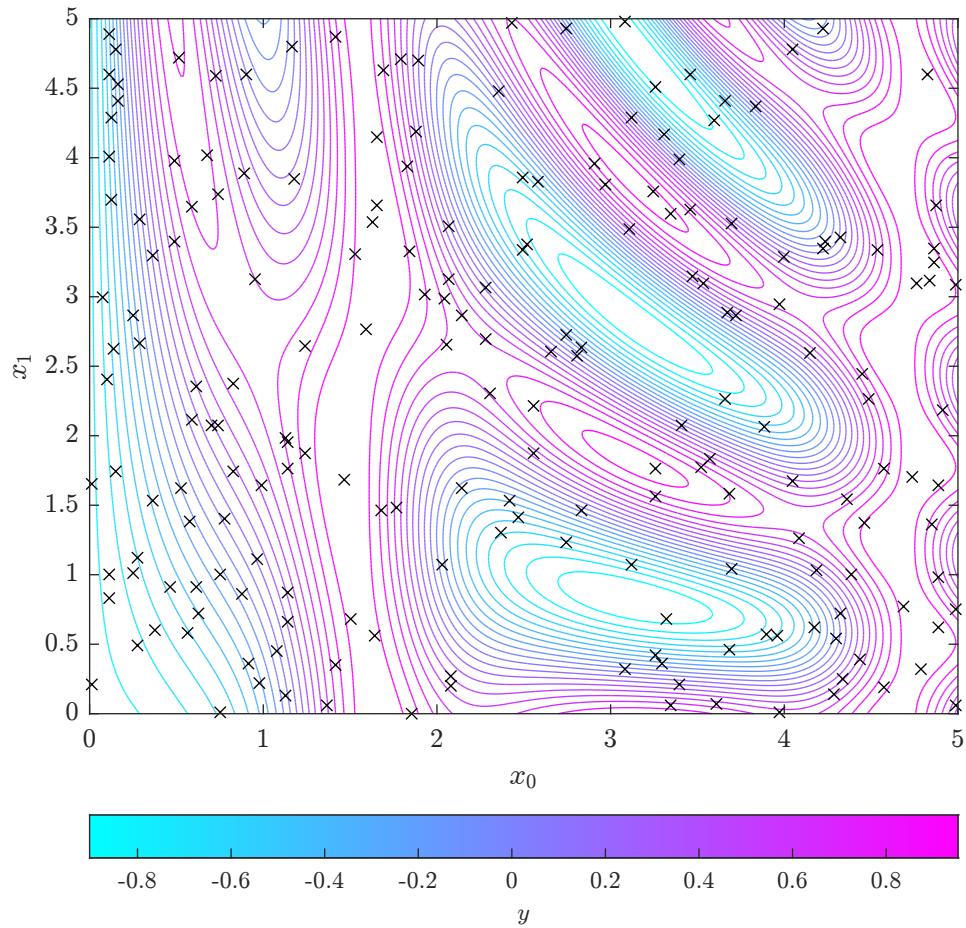


Fig. 2.1 Contour plot of the function used to demonstrate the algorithms presented in previous work. The crosses have been used to show the location of the 200 test data points used within this example.

$$y = \sin(x_0)^{10} + \cos(10 + x_0 x_1) \cos(x_0) \quad (2.1)$$

## 2.1 Active Learning

There are several schools of thought regarding active learning. These can be separated into two distinct categories: current data and future predictions. The former of these is computationally cheaper, as will be apparent on description.

---

**2.1.1 Current Data**

1

**Uncertainty Sampling and Regions of Disagreements**

2

The simplest is applicable to cases in which a certainty is provided with each prediction. Settles [Set09] suggests selecting the data point with the largest uncertainty according to the current model. Using the dataset ”, this is demonstrated in Figure 2.2 with the algorithm for deciding the next sample point given in Algorithm 1.

3

4

5

6

---

**Algorithm 1:** Uncertainty Sampling Selection

---

**Data:**  $X_{\text{known}}$ ,  $Y_{\text{known}}$ ,  $X_{\text{unknown}}$ **Result:** Next  $X$  to label

model = BayesianRidge();

model.fit( $X_{\text{known}}$ ,  $Y_{\text{known}}$ );standard\_deviation = model.standard\_deviation( $X_{\text{unknown}}$ );

---

**return**  $\max(\text{standard\_deviation})$

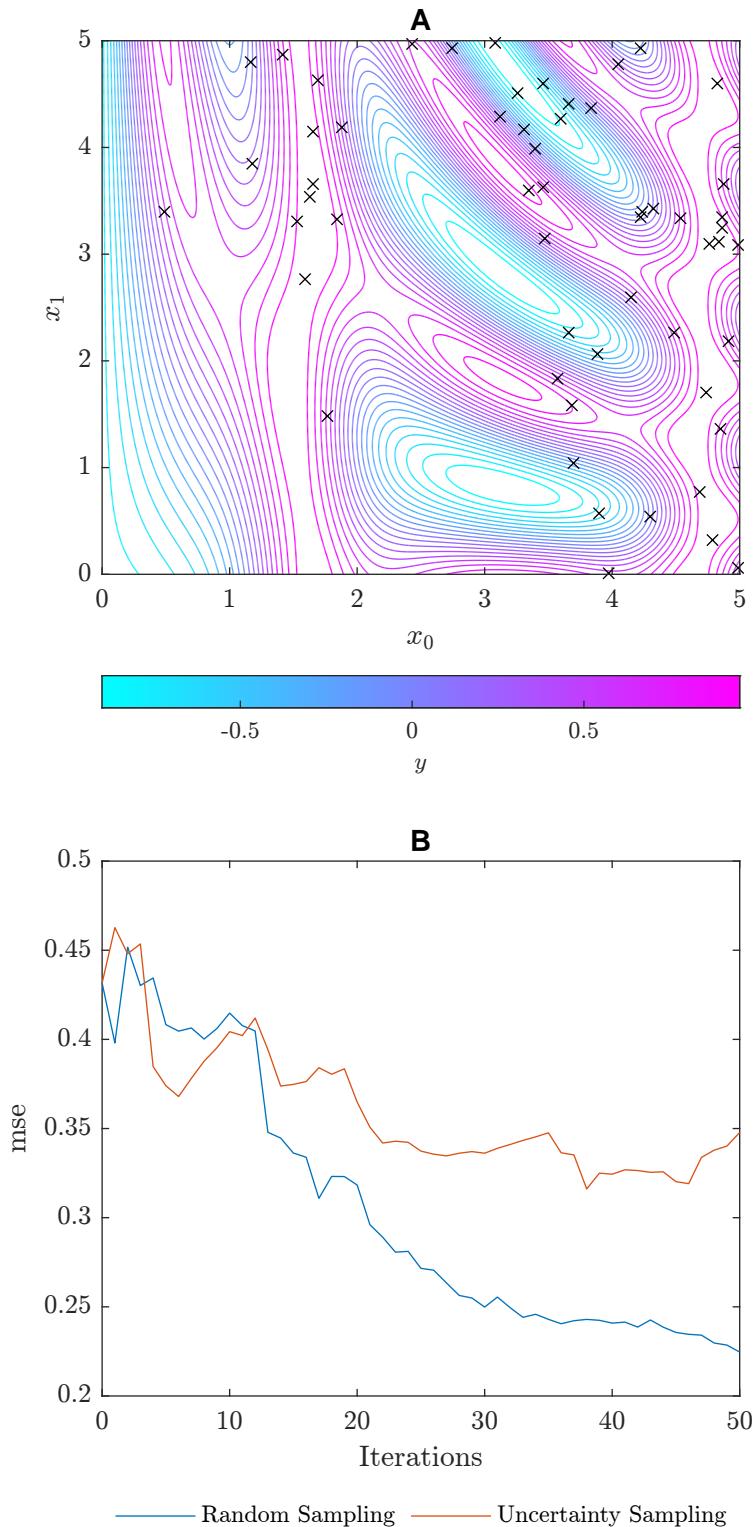


Fig. 2.2 The outcome of the investigating the areas of the highest uncertainty. An initial set of 5 random points was provided, and 50 further iterations were then carried out of sample size 1. A) Demonstrates the final set of points tested by the algorithm and B) shows the change in the mean squared error for the algorithm after each iteration.

## 2.1 Active Learning

## 7

Interestingly, Figure 2.2B shows how the mean squared error for the random sampling method performed to worse within the iterations tested. This is likely due to a bias in the use of linear models in fitting leading to large uncertainties surrounding areas with high curvature. Evidence to this is provided in 2.2A with a large proportion of the sampled points at areas of high curvature.

As addressed by Settles [Set09], this can be extended to any probabilistic model through 2.2. Settles [Set09] also notes the use of information theory for probabilistic models(2.3), where  $y_i$  refers to all possible categorisations for  $x$ . This derives from the principle that the greatest entropy requires the most information to encode, and thus the least certain. However, Settles [Set09] fails to address non-probabilistic models in this instance, instead converting such models into probabilistic ones.

$$x_{\text{next}} = \underset{X}{\operatorname{argmax}} [s_g(X)] \quad (2.2)$$

In order to adapt non-probabilistic models into probabilistic ones, composite models may be used. These are an amalgamation of other models where the standard deviation of the individual models can be taken as the degree of certainty for a given point. Many authors have called this as minimising the region of disagreement as it attempts to produce a coherent hypothesis space. By minimising the region of disagreement between various models, a finer fit may be achieved. Indeed, this was the method used in Figure 2.2.

One way of achieving this, especially in a regression model where boundaries are not quite so distinct, is to declare  $n$  models  $M = \{m_1, \dots, m_n\}$ . Combining these allow for a model  $\hat{m}$  to be defined with prediction  $\hat{y}$ , being the mean prediction of  $M$ ,  $\frac{1}{n} \sum y_i$  and a sample standard deviation  $\hat{s}$  defined as the sample standard deviation of  $y_i$ . This standard deviation can be used as a measure of the disagreement between the models.

$$x_{\text{next}} = \underset{x}{\operatorname{argmax}} \left[ - \sum_i P(y_i|x) \ln P(y_i|x) \right] \quad (2.3)$$

**Broad Knowledge Base**

A second form stems from information theory. Here, the aim is to produce an evenly dispersed  $x$  allowing a well-informed knowledge base. This prevents poor model choice from influencing the algorithm as was seen in 2.2. There are two paths to proceed: density and nearest neighbours.

The former of these requires a definition of density in a sparsely populated space. As an analogy, although the density of a gas appears well-defined, it becomes non-smooth once the

<sup>1</sup> volume defined over is comparable to the distance between particles. Thus, a new definition  
<sup>2</sup> is required.

<sup>3</sup> Alternatively, nearest neighbour requires little explanation.  $x_{\text{next}}$  is the unlabelled data  
<sup>4</sup> point furthest from any labelled data point. The results of 2.4 can be seen in Figure 2.3.

<sup>5</sup>

$$x_{\text{next}} = \operatorname{argmax}_x \left( \sum \frac{1}{\text{sim}(x, x_i)} \right) \quad (2.4)$$

## 2.1 Active Learning

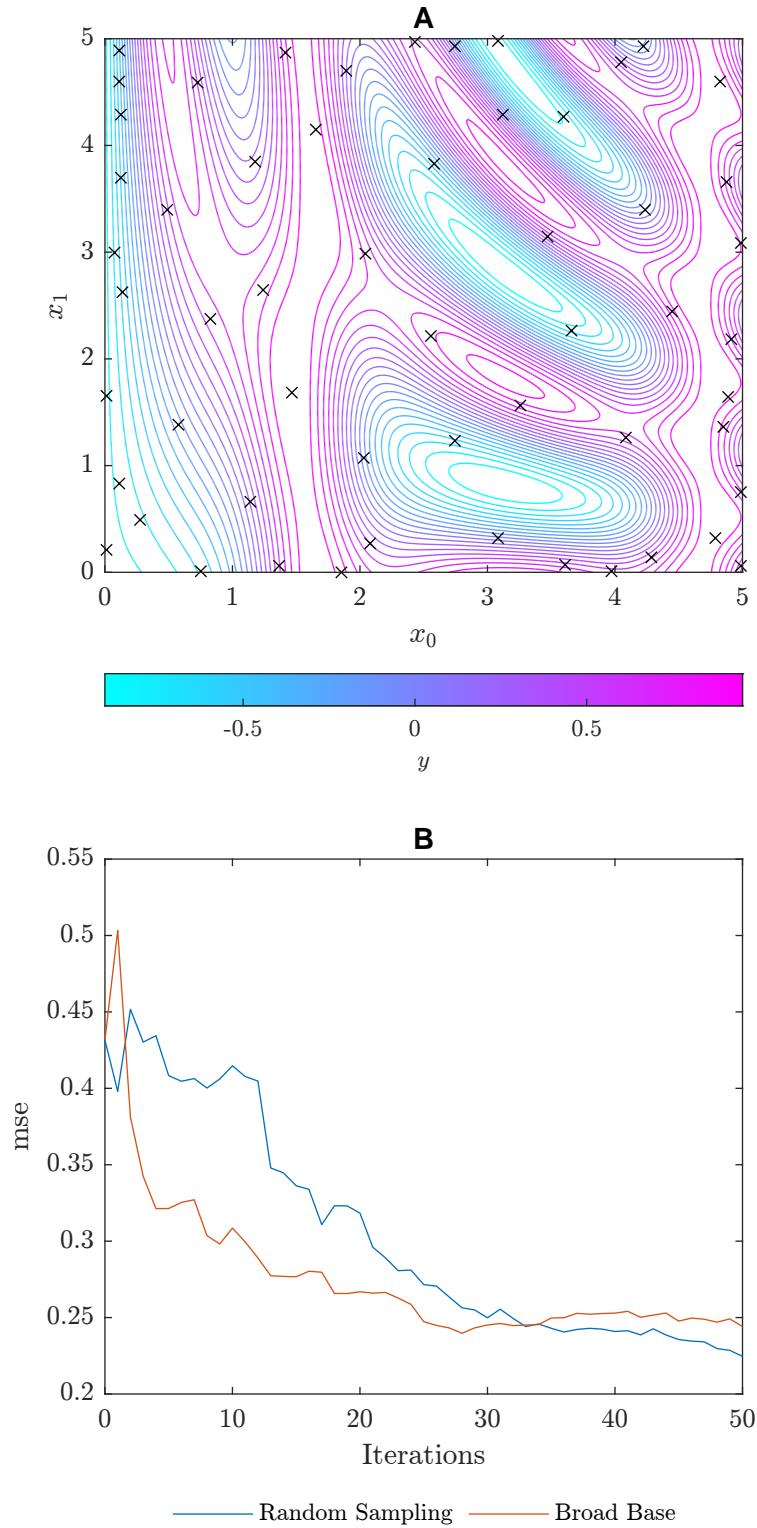


Fig. 2.3 The outcome of the investigating the areas of using a broad base. An initial set of 5 random points was provided, and 50 further iterations were then carried out of sample size 1. A) Demonstrates the final set of points tested by the algorithm and B) shows the change in the mean squared error for the algorithm after each iteration.

## 1 Density Hotspots

2 Conversely, a density weighted model has been suggested, as it escapes the introduction of  
 3 error from outliers (i.e. data points far away from alternative data points). Settles and Craven  
 4 [SC08] suggest (2.5) which can be broken down into two parts: a function for selection,  $\phi_A$ ,  
 5 and a function for similarity, sim. The former arises from another method described in this  
 6 section. The latter requires a function to describe the similarity between data points.

$$7 \quad x_{\text{next}} = \underset{x}{\operatorname{argmax}} \left[ \phi_A(x) \times \left( \frac{1}{U} \sum \text{sim}(x, x_i) \right)^\beta \right] \quad (2.5)$$

8 Settles and Craven [SC08] admits that sim is open for interpretation. It is also recognised  
 9 that this lays the foundation of a clusterisation algorithm. There exist many forms of these  
 10 algorithms, with the results of several of these algorithms on toy data sets presented in  
 11 Figure 2.4 [Sci].

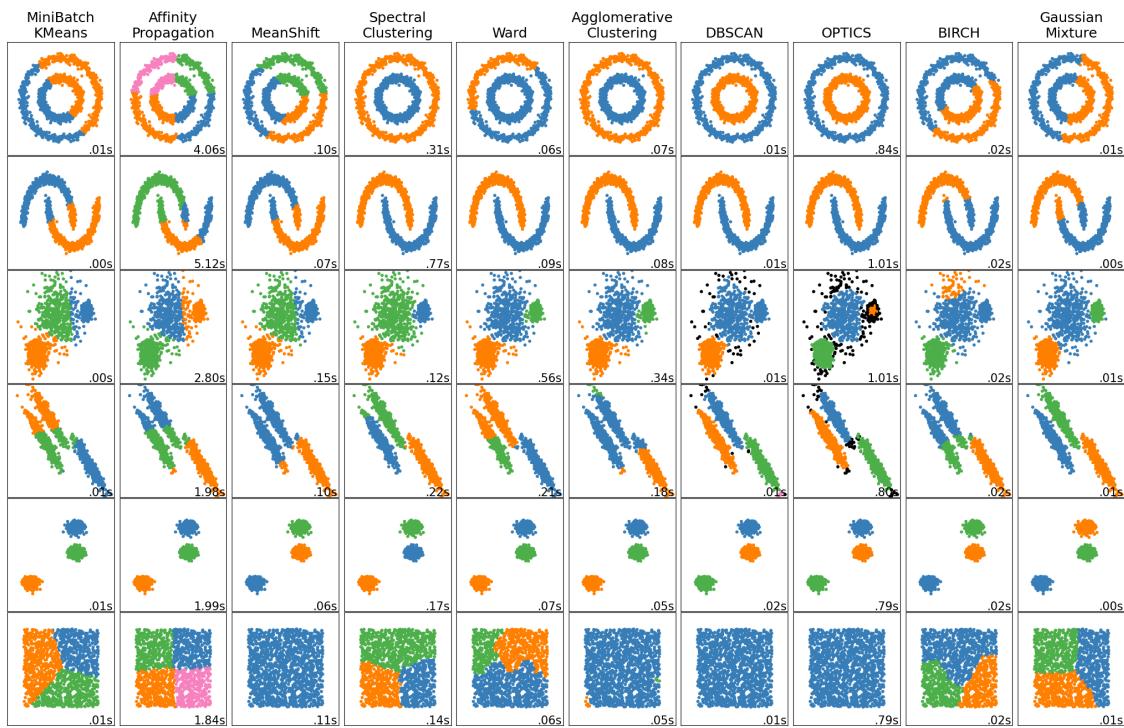


Fig. 2.4 Clusterisation algorithms used on sample two-dimensional data sets to demonstrate resultant clusters.

12 As demonstrated by 2.4 demonstrates, there are multiple different interpretations of the  
 13 solution to the problem of clustering. The makers of the Sci-kit learn package also discuss the

---

**2.1 Active Learning****11**

scalability of each algorithm [Sci]. In order to prepare a high number of features (beyond the two used within this section for demonstration) and large number of data points, it is required that the algorithm scales accordingly. Further, for an adaptive process, it is more suitable for an algorithm to be adaptive to differing distribution. This limits the suitable algorithms to K-Means, Ward and Birch - columns one, five, and nine of Figure 2.4 respectively. Targetting the largest clusters without a known result,

1  
2  
3  
4  
5  
6

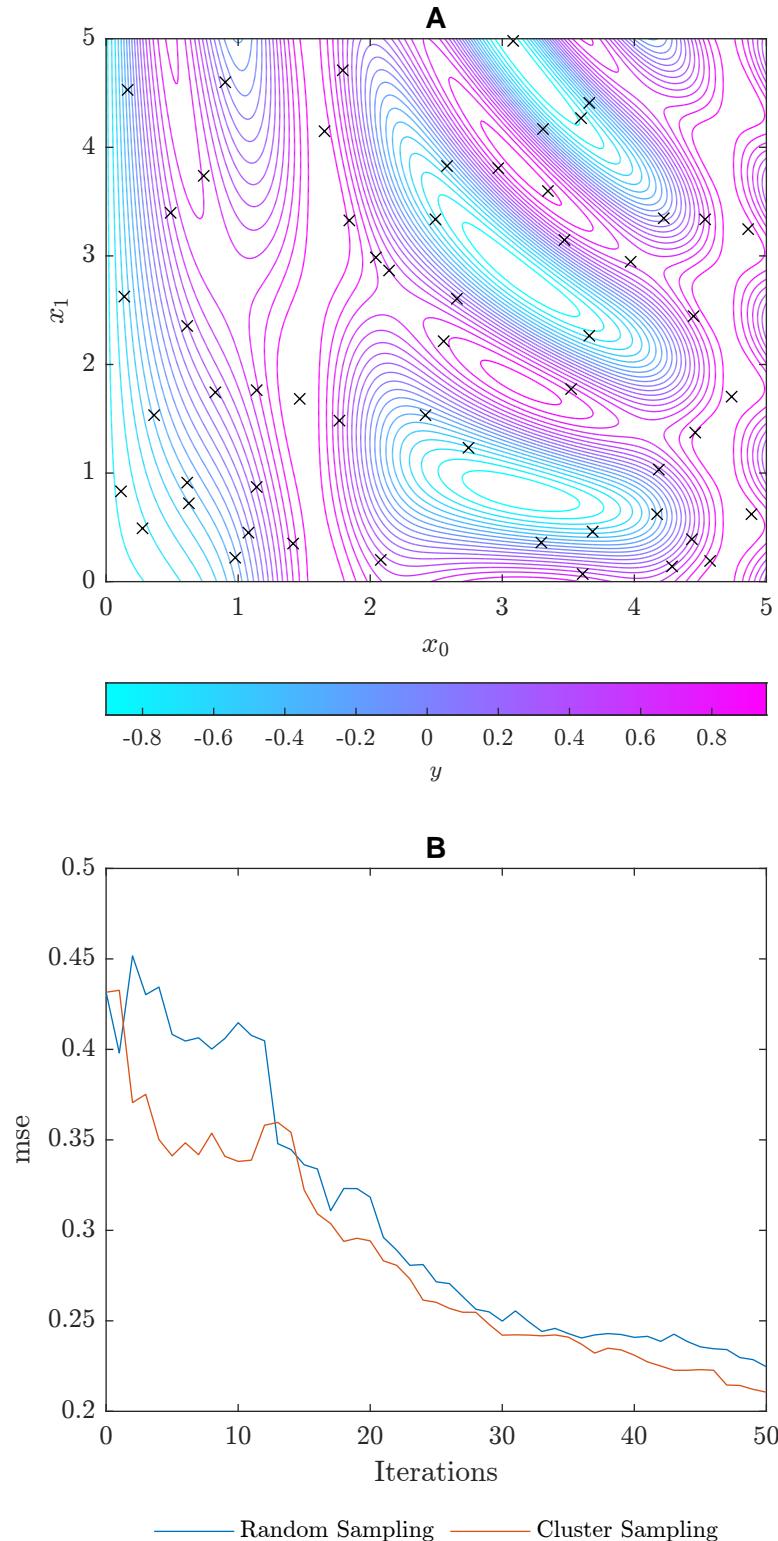


Fig. 2.5 The outcome of the investigating the areas of using a cluster hotspot sampling methodology. An initial set of 5 random points was provided, and 50 further iterations were then carried out of sample size 1. A) Demonstrates the final set of points tested by the algorithm and B) shows the change in the mean squared error for the algorithm after each iteration.

### 2.1.2 Estimated Future

These methods attempt to minimise a future attribute of the model. This works by predicting changes given with the inclusion of more data with a higher degree f theoretical underpinning that the sampling methods discussed thus far.

#### Expected Model Change

As the name implies, this method chooses points which are likely to have the largest impact on the final model. By instigating each potential point, the impact on the eventual model can be found. However, this requires a method for quantifying the model change.

Settles and Craven [SC08] and Settles [Set09] investigate models which can be trained "online": i.e. models which can use the previous iteration to reduce the time taken for convergence. They present a method called "Expected Gradient Length" (EGL) which has a couple of prerequisites: **1)** A probabilistic model is used **2)** Linear gradient based optimisation is used **3)** The model can be improved from previous iterations. Given these prerequisites, the problem becomes less computationally inexpensive given a small dataset or extensive parallelisation, and scales as  $\mathcal{O}(n)$ . However, it does have the distinct drawback of requiring close control of the data models used. Here, the length of the training gradient (the gradient used in re-fitting the parameters with gradient based optimisation) can be used as a measure of model change. In the case of a small model change, as is expected, the lenth of the training gradient can be written as  $\|\nabla l(\langle x, y_i \rangle; \theta)\|$ . Combining this with the probability distribution of  $y$ , the next sample to undergo labelling is given by 2.6.

$$x_{EGL}^* = \operatorname{argmax}_x \sum_i P(y_i|x; \theta) \|\nabla l(\langle x, y_i \rangle; \theta)\| \quad (2.6)$$

## 2.2 Batch Active Learning

Little literature exists with respect to batch active learning. Naive implimentation exist whereby the methods explored earlier present a stack of data points to be chosen, and the top  $N$  are used. However, this method does not take into account the equivalence of the data points. This can be seen by the formation of clusters within the broad and uncertainty sampling methods, although it is not present within the clusterisation algorithm.

It stands to reason that the area which has the highest uncertainty will see this for the data points nearest neighbours. Thus, this singular data point suffers the potential of being surrounded by  $N - 1$  other data points. The benefit this provides in fitting the model is thus

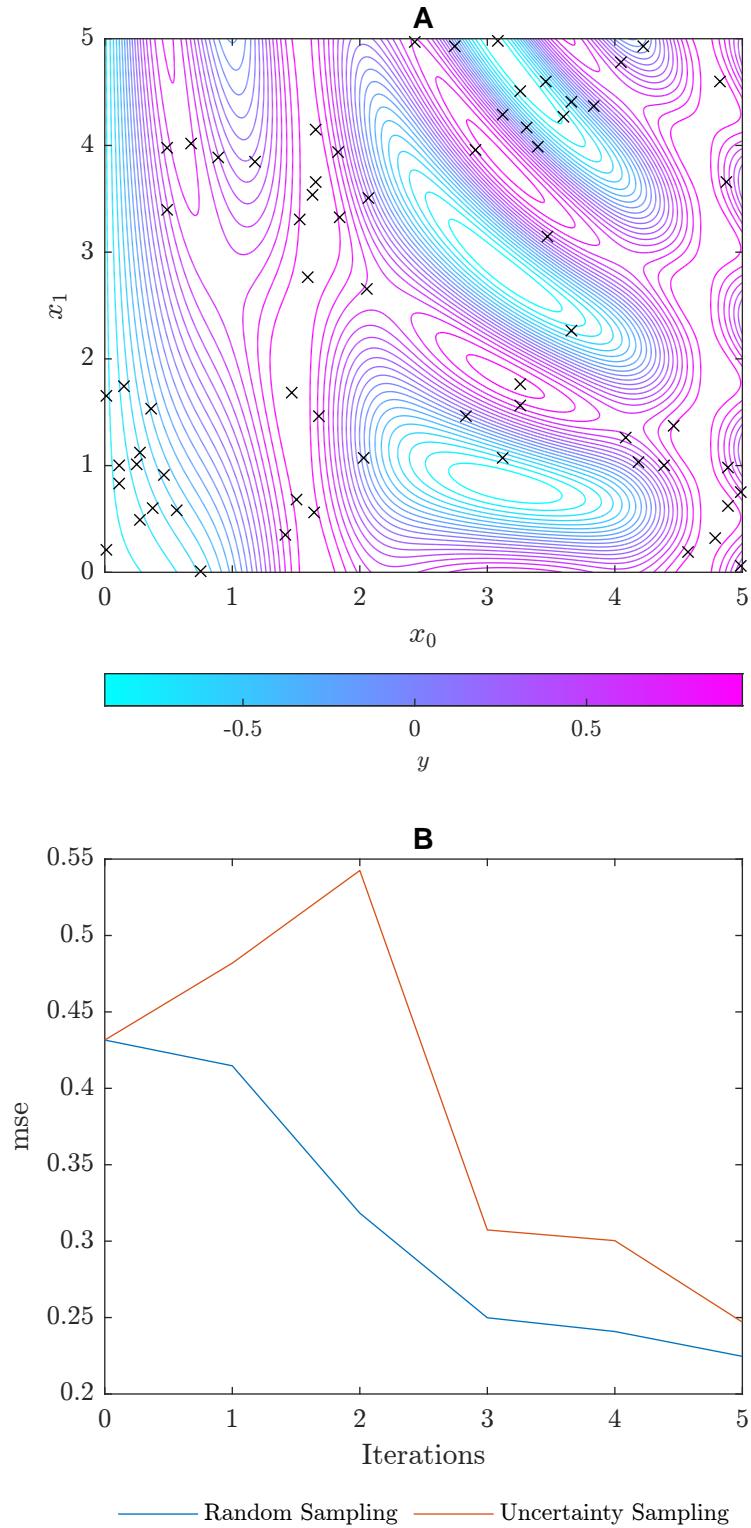


Fig. 2.6 The outcome of the investigating the areas of using uncertainty sampling. An initial set of 5 random points was provided, and 5 further iterations were then carried out of sample size 10. A) Demonstrates the final set of points tested by the algorithm and B) shows the change in the mean squared error for the algorithm after each iteration.

## 2.2 Batch Active Learning

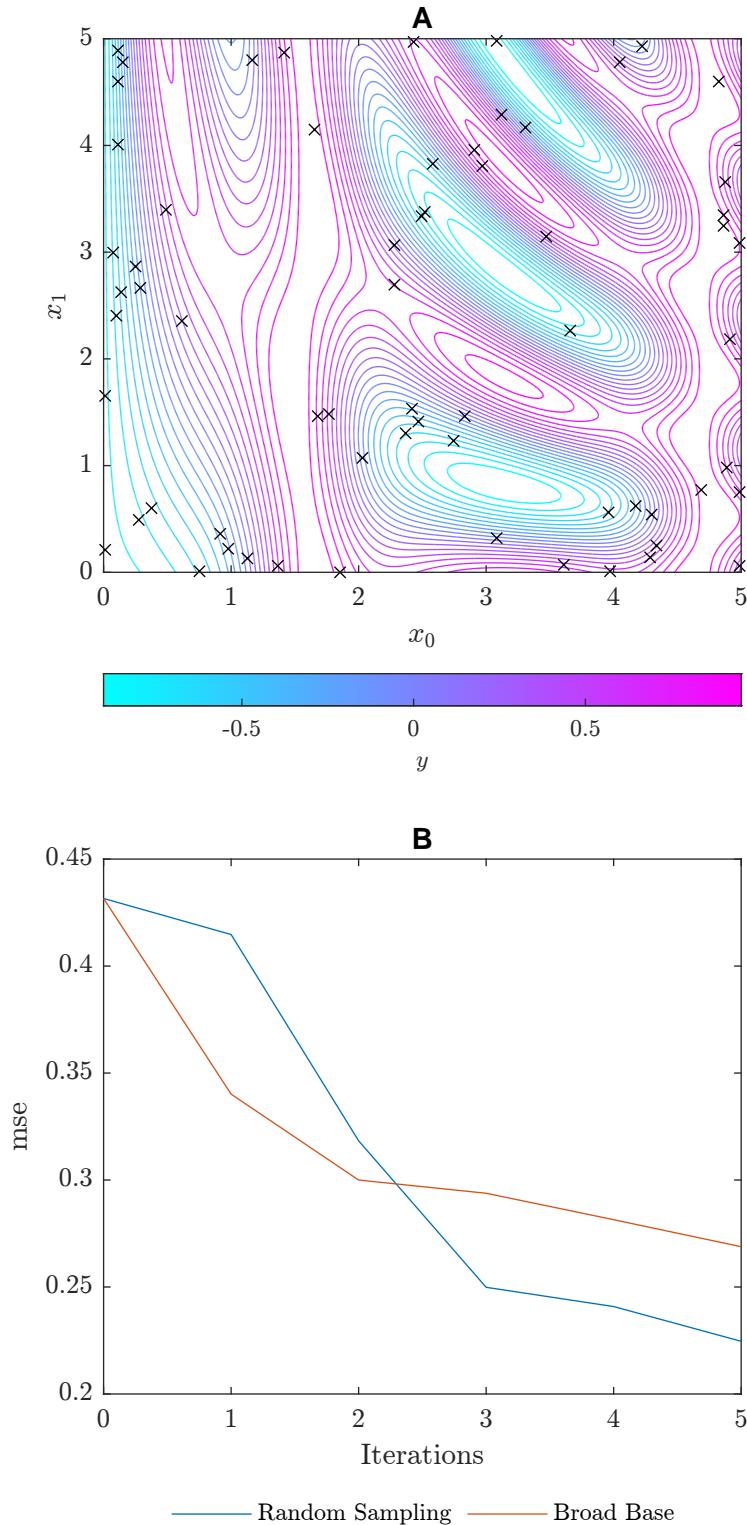


Fig. 2.7 The outcome of the investigating the areas of using broad-base sampling. An initial set of 5 random points was provided, and 5 further iterations were then carried out of sample size 10. A) Demonstrates the final set of points tested by the algorithm and B) shows the change in the mean squared error for the algorithm after each iteration.

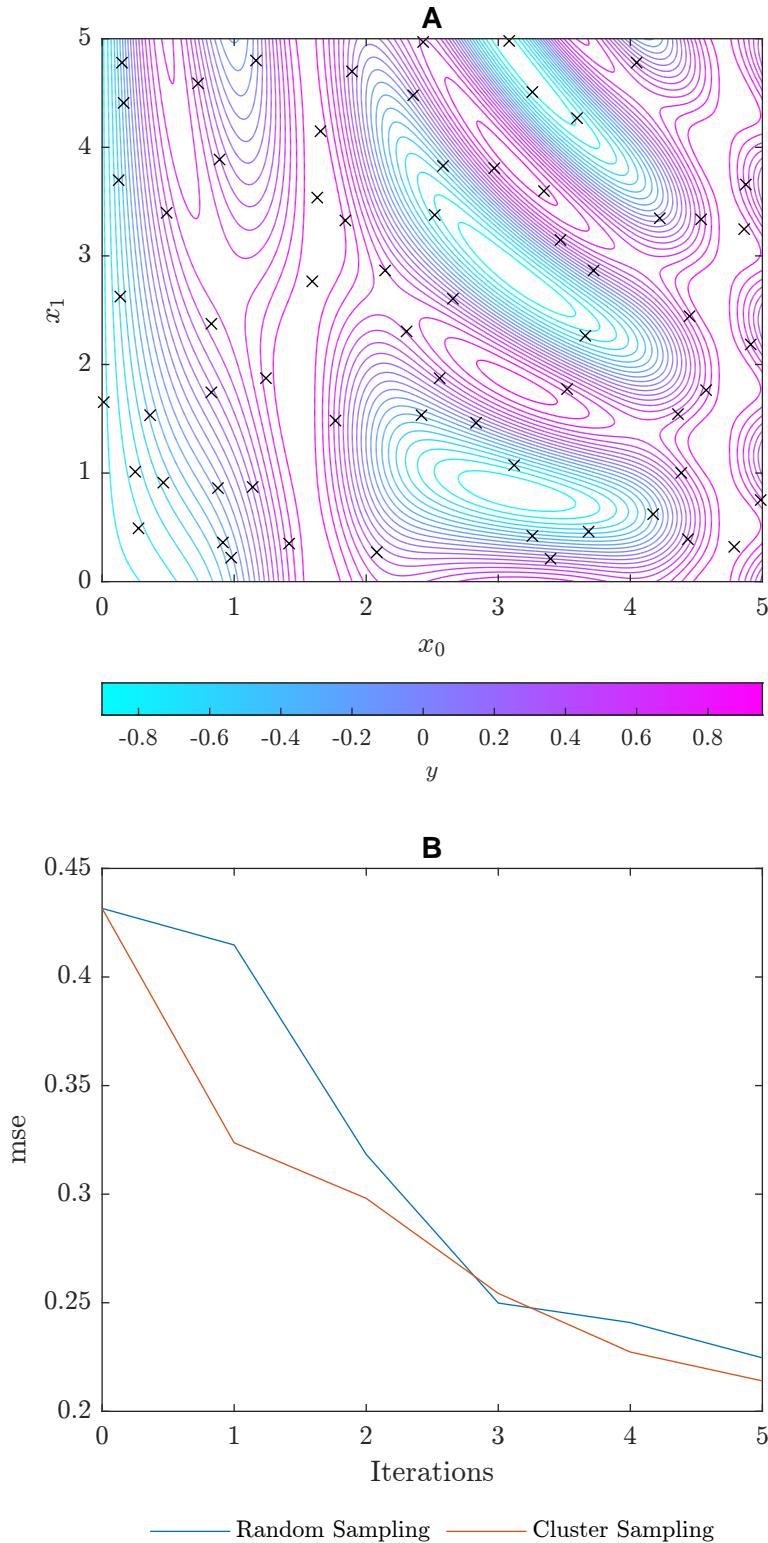


Fig. 2.8 The outcome of the investigating the areas of using cluster sampling. An initial set of 5 random points was provided, and 5 further iterations were then carried out of sample size 10. A) Demonstrates the final set of points tested by the algorithm and B) shows the change in the mean squared error for the algorithm after each iteration.

## 2.3 Drug Data for Machine Learning

17

extremely limited, and only slightly greater than if one data point had been chosen. A simple fix would be to simulate the model after 1 iteration, and select the next point from here. By doing this  $N - 1$  times, a better solution may be found, although this may prove to be computationally very expensive.

### 2.3 Drug Data for Machine Learning

There are numerous data categories that can be used to represent a chemical in a suitable form for machine learning. Indeed, the field of chemoinformatics is dedicated to the pursuit of describing chemicals for computational models. Each of these methods have various strengths and weaknesses. Some are directly based upon the chemical structure whereas others are based upon physical properties. These can be combined to produce models with high predictive capabilities.

#### 2.3.1 Physical Properties

A selection of physical properties from chemicals are known, from melting points to solubility. Many of these provide important aspects for consideration and allow human scientists to predict interactions, especially when determining new drugs. These data are often reported in tables within textbooks such as Perry's [] or provided through software [chembl ...].

Several of these data can be predicted through theoretical models, although the difficulty increases for larger molecules. For example, models exist for density predictions, but predicting the LD<sub>50</sub> of a drug is far more challenging task. Indeed, even with animal testing, this property is deemed difficult to truly assess.

Within drug discovery, physical and biological properties are usually the sought after labels. An example of this is supplied by EMBL-EBI [EMB09] with a custom property named pChEMBL, as defined by 2.7 where "l" is synonymous with or.

$$\text{pChEMBL} = -\log_{10} (\text{IC}_{50} | \text{XC}_{50} | \text{EC}_{50} | \text{AC}_{50} | \text{Ki} | \text{LD}_{50} | \text{Potency}) \quad (2.7)$$

24

#### 2.3.2 Fingerprints

25

Another methodology is to develop a fingerprint: a unique code based on the chemical structure, either of the atomic arrangement, or by the electron cloud distribution. The latter of these is more fundamental to the activity of molecules but far harder to calculate. Indeed,

26

27

28

1 for accurate representation of the latter, both atomic structure is needed *and* solutions for the  
2 Schrödinger equations corresponding to molecule in question.

3 According to Capecchi, Probst, and Reymond [CPR20], the most popular fingerprint in  
4 use are Morgan Fingerprints, a form of Extended Chemical Fingerprint (ECFP). ECFPs use  
5 a simple algorithm in order to generate a unique identifier, as described by Rogers and Hahn  
6 [RH10]:

7 1. **Initial Assignment:** Each atom has an integer assigned as an identifier.

8 2. **Iterative Updating:** Updating the identifier assigned to atoms based on adjacent atoms  
9 and structural duplications.

10 3. **Duplicate Removal:** Duplicate features are removed for hashing.

11 The iteration process involves each atom and adjacent atoms sharing numbers before  
12 in an array. A hash function is applied to this array and becomes the atoms new identifier.  
13 Fingerprints of this class are labelled according to the number of iterations,  $n$ , with the  
14 final name given as ECFP\_⟨2n⟩. Morgan fingerprints, the most common form, are thus also  
15 called ECFP\_4 [CPR20; RH10]. Thus, these come under the remit of fingerprints based  
16 upon two-dimensional chemical structure, rather than three-dimensional or even electron  
17 distribution. Morgan fingerprints are readily available for millions of compounds from the  
18 publicly accessible ChEMBL database [EMB09].

19 Alternative common fingerprints include SMILES, InChI, and the MDL molfile.

# Chapter 3

## Methodology

### 3.1 Outline

#### 3.1.1 Data

Each dataset used consists of a 1024 bit Morgan fingerprint for the features and these associated pChEMBL values. The sets used for parameter fitting and score reporting make up a set of 2094 files from EMBL-EBI [EMB09]. These were filtered to prevent datasets with fewer than 1000 entries to be admitted into the main script.

Morgan fingerprints were chosen due to the ease in which it is to calculate the vectors, the popularity of them within the chemoinformatics sphere, and the success enjoyed by others when using them for predictive purposes. It was decided that physical properties would not be used as this could increase the onus on data sanitation and preparation rather than active learning.

#### 3.1.2 Custom Algorithms

As well as the algorithms used mentioned in Chapter 2, several custom algorithms were developed and added to the testing set. These methods do use parameters, and so require the minimisation technique. Additionally, these algorithms take a composite methodology, using other active learning methods in order to reach a conclusion, so some concepts will be assumed knowledge for Chapter 2.

## **3.2 Computational Methodology**

The methodology presents a novel means of assessing different parametrised active learning methods on existing data sets, allowing for a robust answer into the use of active learning in drug rediscovery. Results can thus be given with a given belief. This approach has taken principles commonly used in machine learning and applied it to more traditional algorithmic methods.

Firstly, a collection of pre-existing data sets,  $X$ , are used.  $X$  is then split into two sub sets:  $X_{\text{train}}$  and  $X_{\text{test}}$ . Similarly to machine learning, the former of these subsets is used in fitting the parameters of the equation, and the latter is used to provide a result without the risk of data leakage into the training set. This is represented in []. Parallelisation is used to efficiently train the algorithms allowing the time for training to be  $\sim \mathcal{O}(c)$ . Datasets used have at least 1000 entries. This results in 164 datasets used for training, and a further 42 used for testing. Examining the smaller details, each algorithm is provided with the sets  $x_{\text{known}}$ ,  $y_{\text{known}}$ , and  $x_{\text{unknown}}$ . Various algorithms are given these sets and allowed to generate a subset of  $x_{\text{unknown}}$  to be added into  $x_{\text{known}}$  alongside corresponding  $y_{\text{known}}$ . This can then repeat until a predefined stopping point is reached. Scores are reported using a weighted mean squared error [] based upon  $y_{\text{predict}}$  for all  $x$ . This is similar to a standard machine learning methodology with a couple of differences. Firstly, no distinction is made between the training and testing set within a dataset contrary to standard practice. This is due to two reasons. Firstly, the datasets are not large enough for an accurate representation of the data within the testing set, and secondly, the scoring to each dataset is not used within the machine learning algorithms to fit parameters as is usually the case. All algorithms used rely upon a simple custom composite model to allow for flexibility and consistency.

In Section [], it was discussed that there are various methodologies of representing chemicals and drugs. ... (if time)

### **3.2.1 Integral Functions and Classes**

Several key methods and classes are required for the smooth operation of the computational frameworks used

### 3.2.2 Custom Base Functions

#### Split

The split function allows for each dataset to be split into  $x_{\text{known}}$ ,  $y_{\text{known}}$ ,  $x_{\text{unknown}}$ , and  $y_{\text{unknown}}$ , as demonstrated in Figure 3.1. This is required as a fundamental step for the algorithmic testing. To demonstrate the validity of this function, ...

#### Repartition

Upon each iteration, the sets provided to the algorithms need to be repartitioned to allow for the continual operation of the algorithm. This consists of two parts: expanding the known sets and removing entries from the unknown sets.

#### Model

The machine learning model is the only custom class used. Here, a similar structure is used when compared with sci-kit's machine learning [Ped+], as is demonstrated in Table 3.1. To manage this, it has four methods: `__init__`, `fit`, `predict`, and `predict_error`. The last of these is not seen in all sci-kit's machine learning models and is reserved for those which can report a certainty of prediction. Here, this was achieved by taking a standard deviation of the models.

|            | Name  | Description  |
|------------|---|--|
| Attributes | Models: List  | List of models to be used in composite   |
| Methods    | <code>fit(X: int[][], Y: double[])</code><br><code>predict(X: int[][]): double[]</code><br><code>predict_error(X: int[][]): double[][]</code> | Fits the models in Models<br>Takes a set of labels and returns mean predicted label from all the models.<br>Takes a set of labels and returns the mean predicted label from all the models and standard deviations of model predictions. |

Table 3.1 Schema for the Model Class.

The models used for the composite model were ... which will be consistent across all algorithms. This allows direct comparison of the algorithms without interference from machine learning models used.

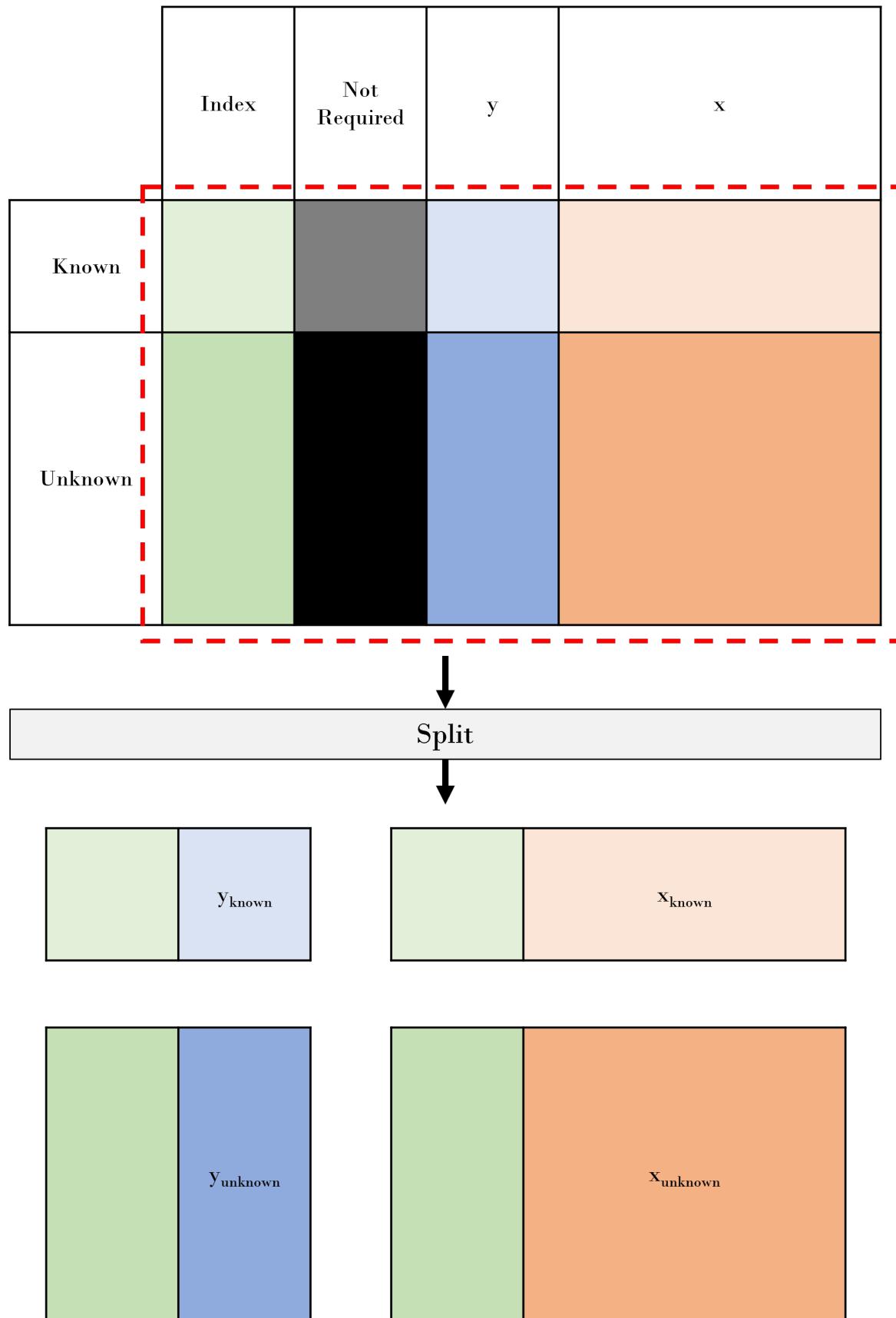


Fig. 3.1 Graphical representation of the split function. The red dashed boundary represents the input (additional colour coding has been performed to assist the reader in understanding the transposition of the base components).

**Validate**

This is a simple method with greater potential than has been explored. By providing this as a separate method, a more computationally intensive validation model could be used without interference as parallelisation could be exploited. However, as it currently stands, it returns the weighted mean squared error using the standard method provided by [].

**3.2.3 Active Learning Algorithms****Dumb**

The dumb algorithm, also referred to as random sampling or Monte Carlo sampling, refers to an algorithm that calls upon random samples to be tested. This represents the computationally least expensive approach, and is thus used as a baseline in comparing other algorithms. Since the datasets are shuffled prior to being used, the algorithm is extremely simple, as demonstrated in Algorithm 2.

**Algorithm 2:** Uncertainty Sampling Selection

---

**Data:**  $X_{\text{unknown}}$

**Result:**  $X$  ordered according to priority for sampling

**return** ones\_like( $X_{\text{unknown}}$ )

---

**Greedy**

Since the largest activity is sought, a methodology proposed is to simply seek the predicted highest label. Here, the predict() method (see Table 3.1) was used to return a prediction and a standard deviation. The indices of  $x_{\text{unknown}}$  were then returned, ordered descending with respect to the afore mentioned standard deviations.

**Algorithm 3:** Greedy Sampling Selection

---

**Data:**  $X_{\text{known}}, Y_{\text{known}}, X_{\text{unknown}}, \text{Model}$

**Result:**  $X$  ordered according to priority for sampling

$\text{Model.fit}(X_{\text{known}}, Y_{\text{known}});$

$\text{prediction} = \text{Model.predict\_error}(X_{\text{unknown}});$

**return**  $-\text{prediction}$

---

---

## 1 Region of Disagreement

2 Similarly to the region of disagreement method, this is a very simple algorithm. Here, the  
 3 predict\_error() method (see Table 3.1) is used to return a prediction and a standard deviation.  
 4 The prediction is ignored, and instead the standard deviation is returned, multiplied by  $-1$  to  
 5 ensure the largest uncertainty has the lowest "score". This is shown with Algorithm 4.

---

### Algorithm 4: Uncertainty Sampling Selection

---

**Data:**  $X_{\text{known}}$ ,  $Y_{\text{known}}$ ,  $X_{\text{unknown}}$ , Model  
**Result:**  $X$  ordered according to priority for sampling  
 Model.fit( $X_{\text{known}}$ ,  $Y_{\text{known}}$ );  
 $\_, \text{error} = \text{Model.predict\_error}(X_{\text{unknown}})$ ;  
**return**  $-\text{error}$

---

## 6 Hotspot Cluster I

7 This is the first of the clustering algorithms and the first parametric algorithm. This algorithm  
 8 is based upon the ideology presented in Section 2.1.1, and is shown in Algorithm 5. Here,  $c$   
 9 is the number of cluster sought, and is a parameter that requires fitting. Bounds can be placed  
 10 upon this. The lower limit can be set as the number of known data points, and the upper as  
 11 the total number of data points in the data set, although it is hypothesised that beyond the  
 12 sum of the known points and the samples sought would make little, to no difference. To test  
 13 this hypothesis, the upper limit will be set at  $\text{len}(X_{\text{unknown}}) + 1.5n$ . The combined limits have  
 14 been shown in 3.1.

$$15 \quad \text{len}(X_{\text{known}}) < c < \text{len}(X_{\text{unknown}}) + 1.5n \quad (3.1)$$

---

### Algorithm 5: Uncertainty Sampling Selection

---

**Data:**  $X_{\text{known}}$ ,  $X_{\text{unknown}}$ ,  $c$   
**Result:**  $X$  ordered according to priority for sampling  
 combined\_x = concat(X, x);  
 clusters = cluster(number\_of\_clusters=c);  
 clusters.fit(combined\_x);  
 predicted\_custers = clusters.predict( $X_{\text{unknown}}$ );  
 distances = clusters.distance\_to\_nearest\_centroid( $X_{\text{unknown}}$ );  
**return**  $-\text{error}$

---

**Hotspot Cluster II**

This is similar to the previous algorithm with one difference, the labels. Both known and predicted are used within the algorithm to ...

**Hotspot Cluster III**

The final hotspot clustering algorithm also encompasses the uncertainty from the prediction models.

**Region of Disagreement with Greedy Sampling**

This is the first composite function, combining both the greedy sampling, and the uncertainty sampling algorithms. This metric is shown in 3.2.

$$\text{score}_{\text{greedy}\&\text{uncertainty}} = \text{score}_{\text{greedy}}^{\alpha} \text{score}_{\text{uncertainty}}^{1-\alpha} \quad (3.2)$$

Here,  $\alpha$  is a parameter which needs to be found, bounded as  $0 < \alpha < 1$ . Note here that the limits are identical to the uncertainty sampling and the greedy methodologies.

**3.2.4 Training Framework**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

**Parallelisation**

The large number of datasets used presents a problem: time. Indeed, each iteration sees a new fitting of a machine learning model. Within the training stage, this would correspond to 984 models trained: a considerable number. Thus, by exploiting parallelisation, the time can be reduced in execution to the case, where given an infinite number of processes, the training

<sup>1</sup> and testing framework would scale as  $\mathcal{O}(c)$ . This requires circumventing pythons global  
<sup>2</sup> interpreter lock. Pathos was used to accomplish this due to several shortcomings found with  
<sup>3</sup> the default multiprocessing package [McK+12; MA10].

#### <sup>4</sup> **Minimisation**

<sup>5</sup> The methodology for minimisation was dependent on the circumstance. Firstly, the range was  
<sup>6</sup> proved viable using a grid of points within the bound suggested earlier. Upon confirmation of  
<sup>7</sup> the bounds, a more vigorous method was used: particularly in the case where doubles could  
<sup>8</sup> be used. In several examples, such as in Clusterisation I, the parameter used was a integer.  
<sup>9</sup> This was minimised through a binary search, as most available optimisers fail to adapt to  
<sup>10</sup> integer arguments. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis  
<sup>11</sup> facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante.  
<sup>12</sup> Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh  
<sup>13</sup> lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut  
<sup>14</sup> porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis  
<sup>15</sup> fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum  
<sup>16</sup> augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris.  
<sup>17</sup> Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit  
<sup>18</sup> amet ipsum. Nunc quis urna dictum turpis accumsan semper.

# Chapter 4

## Results

### 4.1 Non-Parametric

Non-parametric equations have the benefit of not requiring the minimisation function. Due to this, all testing of these algorithms were undertaken on a standard laptop. These also tend to be the easiest to implement, as uncovered in Chapter 3. Particularly important is the Monte Carlo method as this allows shows what should be a minimum baseline to achieve.

#### 4.1.1 Monte Carlo

The first non-parametised algorithm discussed in Chapter 3 was the Monte Carlo method. Due to the non-parametric nature of this algorithm, execution was simply carried out on the test data set. Results are presented in Figure 4.1, demonstrating a final weighted mse of  $0.74 \pm 0.20$ .

#### 4.1.2 Greedy

Likewise, the greedy algorithm was tested, with results presented in Figure 4.2. Here, a final weighted mse of  $1.2 \pm 0.7$  was found indicating a worse scoring despite more precise results when compared to Monte Carlo - the base case.

#### 4.1.3 ROD Sampling

The final non-parametric algorithm to be tested was ROD. A final weighted mse of  $0.89 \pm 0.30$ , over performing the previous two algorithms in both accuracy and precision.

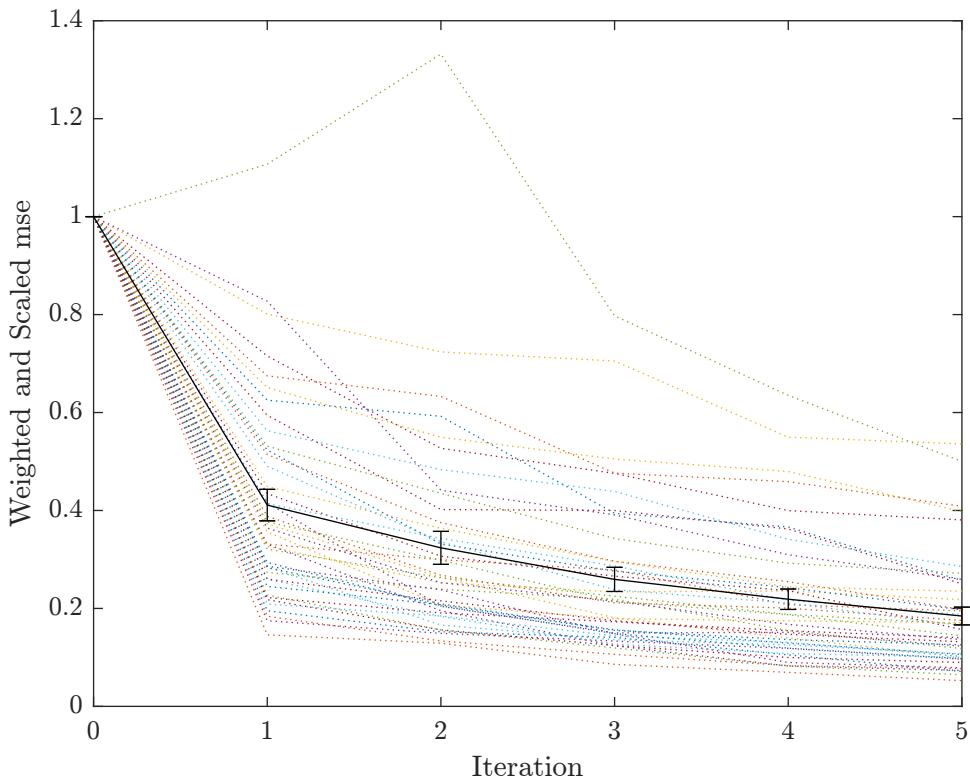


Fig. 4.1 Results of Monte Carlo sampling on the test datasets. Dotted lines represent the individual scoring for the data sets and the solid line shows the mean results at each iteration with error bars of 1 standard deviation.

## **4.2 Parametric**

Parametric algorithms require a minimisation procedure on the training set. This leads to a computationally challenging script, and for this the author is grateful for the services provided by the HPC [Uni22].

### **4.2.1 Hotspot Clusters**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus

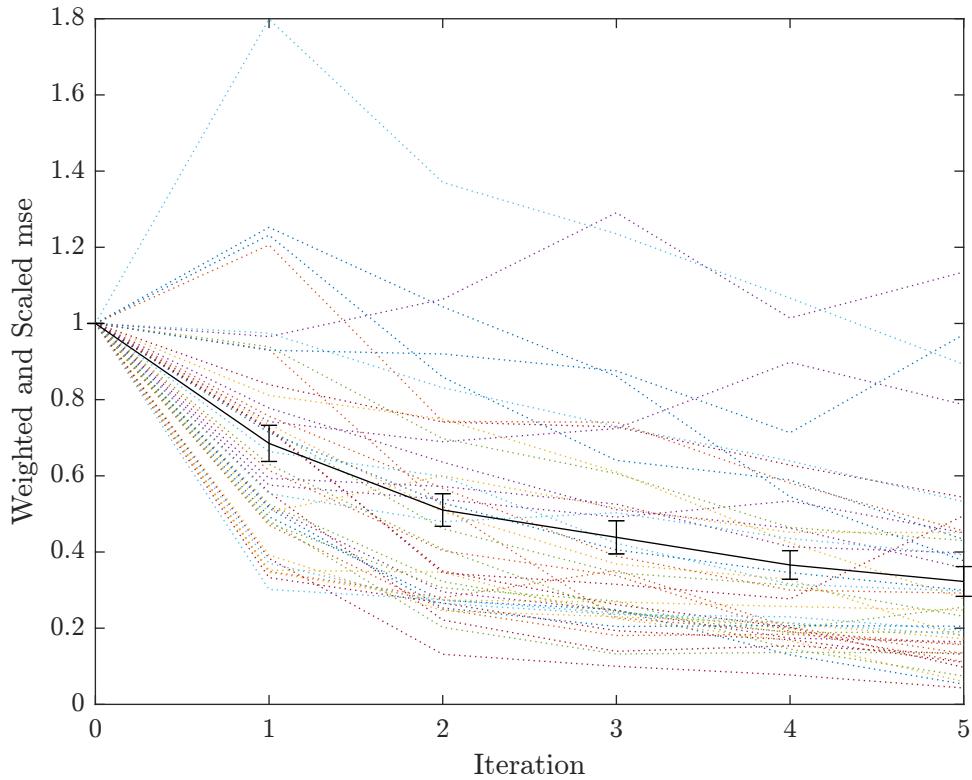


Fig. 4.2 Results of greedy sampling on the test datasets. Dotted lines represent the individual scoring for the data sets and the solid line shows the mean results at each iteration with error bars of 1 standard deviation.

tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### 4.2.2 ROD with Greed

When testing the ROD with greed sampling method, it was found that despite the weighting towards higher value targets, no improvement was seen over ROD with  $\alpha = 0$ , with  $\alpha$  defined in 3.2. However, the tolerance at small  $\alpha$  is low, as shown with the errors attached during parameter fitting.

## 4.3 Special Case: COVID-19

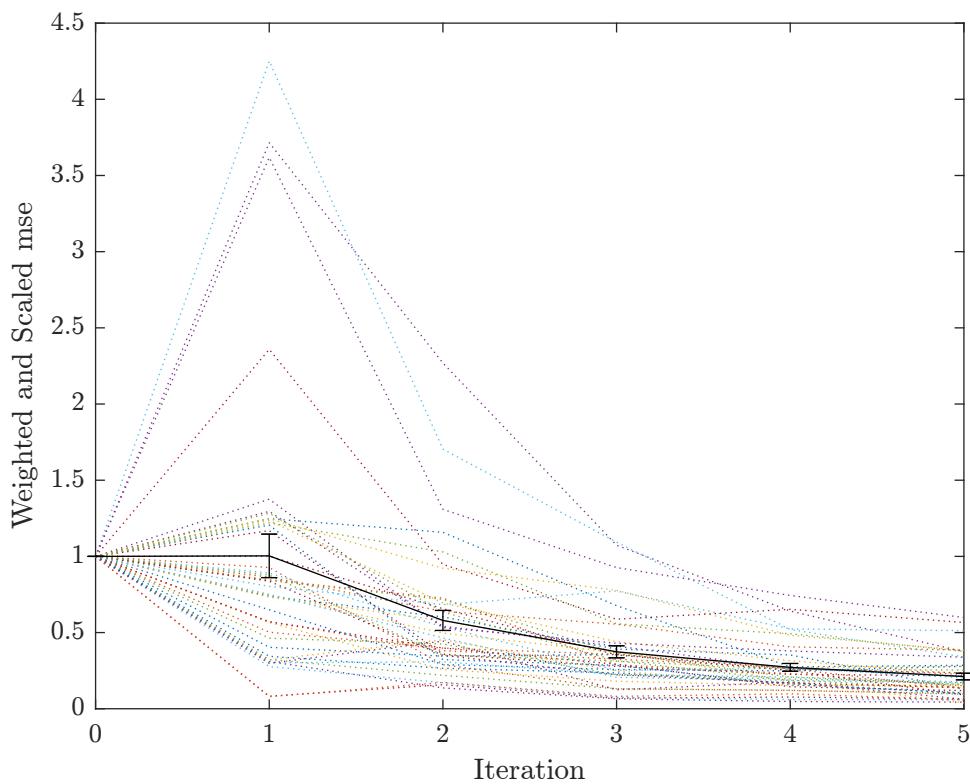
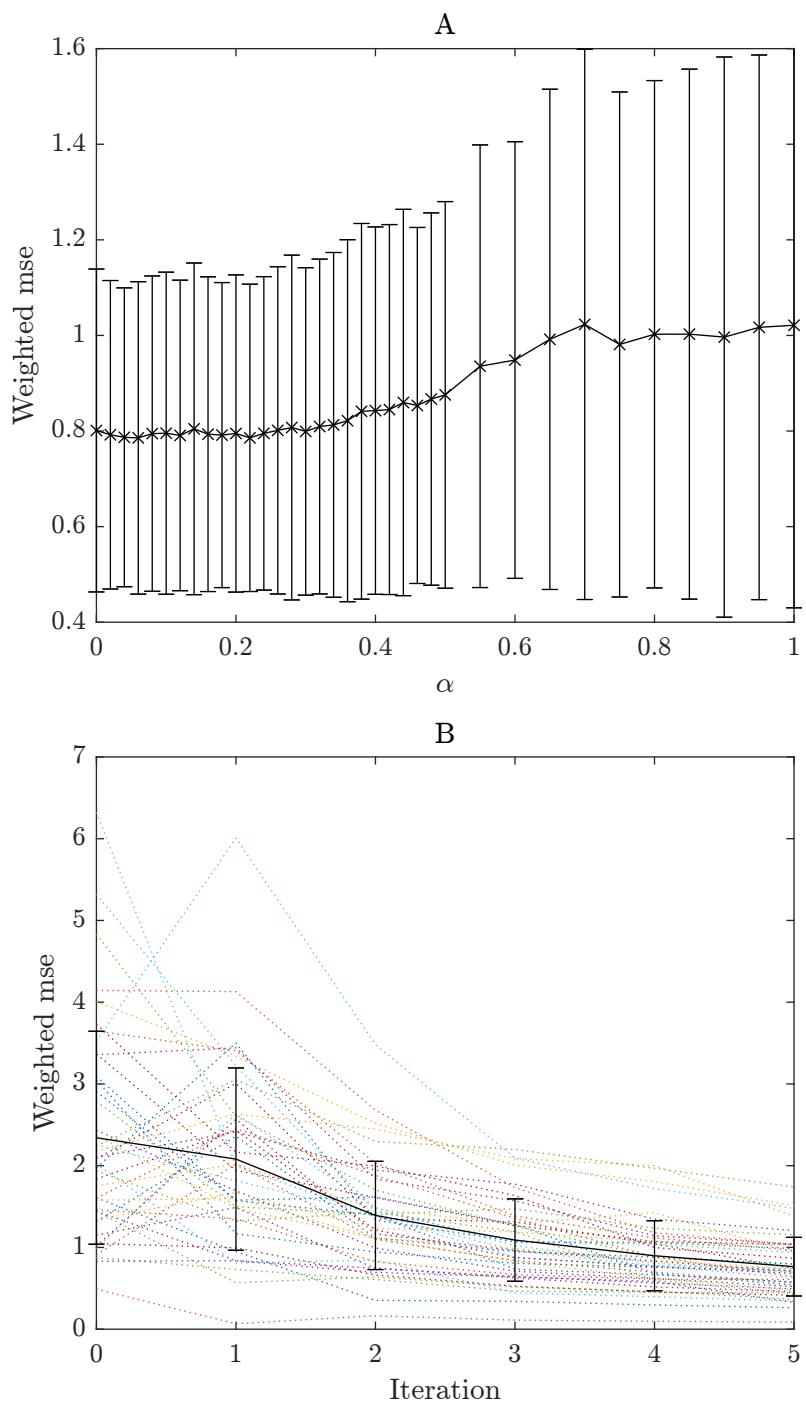


Fig. 4.3 Results of ROD sampling on the test datasets. Dotted lines represent the individual scoring for the data sets and the solid line shows the mean results at each iteration with error bars of 1 standard deviation.



Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

# Chapter 5

## Discussion

### 5.1 Non-Parametric

Three algorithms tested of non-parametric variety producing several noticeable results. Firstly, the baseline result did not produce the worst results with respect to accuracy, although precision was consistently worse. This is demonstrated convincingly through Figure 5.1 where results from the greedy results suggest the worst accuracy.

Despite the greedy algorithm demonstrating the worst accuracy, interesting results were shown with ROD sampling. Indeed, the progression from the first to the second and third iteration demonstrates a faster average learning rate than the other algorithms. This is expected as the ROD algorithm specifically targets regions of the model which are challenging causing the largest changes towards proper fitting.

Both ROD and greedy sampling are suspected to suffer from clusterisation whereby data points similar to each other in the feature space are sampled within the same batch, thus reducing the total information conveyed per batch operation. The random nature of Monte Carlo reduces this prospect, hence the apparent promising performance of a random sampling methodology.

### 5.2 Parametric

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien.

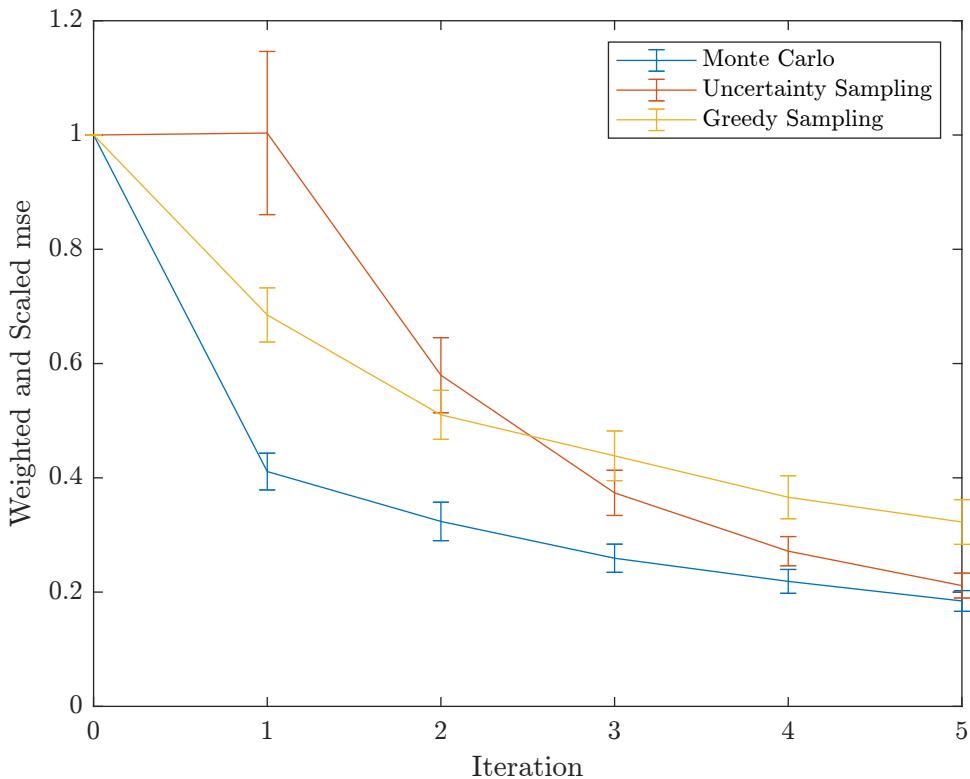


Fig. 5.1 Comparison of different non-parametric algorithms with standard deviations represented as error bars.

1 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed  
 2 interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit  
 3 amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet  
 4 aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna  
 5 dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.  
 6 Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet  
 7 mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a  
 8 dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in,  
 9 velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing  
 10 elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam  
 11 rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit  
 12 mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia  
 13 lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor  
 14 sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque  
 15 pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales

commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc  
nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum  
dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum  
libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante  
lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam,  
luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis  
accumsan semper.

1  
2  
3  
4  
5  
6  
7

Draft - v1.1

Monday 9<sup>th</sup> May, 2022 – 05:01

# Chapter 6

1

## Conclusion

2

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam,

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

<sup>1</sup> luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis  
<sup>2</sup> accumsan semper.

# References

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- Capecci, Alice, Daniel Probst, and Jean-Louis Reymond (June 12, 2020). “One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome”. In: *Journal of Cheminformatics* 12.1, p. 43. ISSN: 1758-2946. DOI: 10.1186/s13321-020-00445-4. URL: <https://doi.org/10.1186/s13321-020-00445-4> (visited on 05/06/2022).
- Center for Drug Evaluation and Research (Apr. 25, 2022). “Coronavirus Treatment Acceleration Program (CTAP)”. In: FDA. Publisher: FDA. URL: <https://www.fda.gov/drugs/coronavirus-covid-19-drugs/coronavirus-treatment-acceleration-program-ctap> (visited on 05/05/2022).
- EMBL-EBI (2009). *ChEMBL Database*. URL: <https://www.ebi.ac.uk/chembl/> (visited on 05/06/2022).
- McKerns, Michael and Michael Aivazis (2010). *pathos: a framework for heterogeneous computing*. URL: <http://uqfoundation.github.io/project/pathos>.
- McKerns, Michael M. et al. (Feb. 6, 2012). “Building a Framework for Predictive Science”. In: *arXiv:1202.1056 [cs]*. arXiv: 1202.1056. URL: <http://arxiv.org/abs/1202.1056> (visited on 05/07/2022).
- Pardo, Joe et al. (May 22, 2020). “The journey of remdesivir: from Ebola to COVID-19”. In: *Drugs in Context* 9, pp. 2020–4–14. ISSN: 1745-1981. DOI: 10.7573/dic.2020-4-14. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7250494/> (visited on 05/05/2022).
- Pedregosa, Fabian et al. (n.d.). “Scikit-learn: Machine Learning in Python”. In: *MACHINE LEARNING IN PYTHON* (), p. 6.
- Rogers, David and Mathew Hahn (Feb. 4, 2010). “Extended-Connectivity Fingerprints | Journal of Chemical Information and Modeling”. In: *Journal of Chemical Information and Modeling* 50.5. DOI: 10.1021/ci100050t. URL: <https://pubs.acs.org/doi/10.1021/ci100050t> (visited on 11/01/2021).
- Scikit Learn (2022). 2.3. *Clustering*. scikit-learn. URL: <https://scikit-learn.org/stable/modules/clustering.html> (visited on 05/05/2022).
- Settles, Burr (2009). *Active Learning Literature Survey*. Technical Report. Accepted: 2012-03-15T17:23:56Z. University of Wisconsin-Madison Department of Computer Sciences. URL: <https://minds.wisconsin.edu/handle/1793/60660> (visited on 11/01/2021).
- Settles, Burr and Mark Craven (Oct. 25, 2008). “An analysis of active learning strategies for sequence labeling tasks”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '08. USA: Association for Computational Linguistics, pp. 1070–1079. (Visited on 05/01/2022).
- University of Cambridge (2022). *Research Computing Services*. URL: <https://www.hpc.cam.ac.uk/> (visited on 05/07/2022).
- Wang, Haidong et al. (Apr. 16, 2022). “Estimating excess mortality due to the COVID-19 pandemic: a systematic analysis of COVID-19-related mortality, 2020–21”. In: *The Lancet* 399.10334. Publisher: Elsevier, pp. 1513–1536. ISSN: 0140-6736, 1474-547X.

- <sup>1</sup> DOI: 10.1016/S0140-6736(21)02796-3. URL: [https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(21\)02796-3/fulltext](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(21)02796-3/fulltext) (visited on 05/06/2022).
- <sup>3</sup> World Health Organization (May 6, 2022). *WHO Coronavirus (COVID-19) Dashboard*. URL: <https://covid19.who.int> (visited on 05/06/2022).