Bayesian Optimisation for Nice Colours

Hannah Kockelbergh

Hertford College

Supervised by

Michael Osbourne

Department of Engineering Science

University of Oxford

May 3, 2018

UNIVERSITY OF OXFORD

FINAL HONOUR SCHOOL OF
ENG / EEM (delete as appropriate)

DECLARATION OF AUTHORSHIP

You should complete this certificate. It should be bound into your fourth year project report, immediately after your title page. Three copies of the report should be submitted to the Chairman of examiners for your Honour School, c/o Clerk of the Schools, examination Schools, High Street, Oxford.

Name (in capitals): ……………………………………………………………………………………………

College (in capitals): ………………………………… **Supervisor:** …………………………………………

Title of project (in capitals): ………………………………...………………………………………………

Page count (excluding risk and COSHH assessments): ……………………………………………………

Please tick to confirm the following:

I have read and understood the University's disciplinary regulations concerning conduct in examinations and, in particular, the regulations on plagiarism (*Essential Information for Students. The Proctors' and Assessor's Memorandum*, Section 9.6; also available at www.admin.ox.ac.uk/proctors/info/pam/section9.shtml). □

I have read and understood the Education Committee's information and guidance on academic good practice and plagiarism at www.admin.ox.ac.uk/edc/goodpractice. □

The project report I am submitting is entirely my own work except where otherwise indicated. □

It has not been submitted, either partially or in full, for another Honour School or qualification of this University (except where the Special Regulations for the subject permit this), or for a qualification at any other institution. □

I have clearly indicated the presence of all material I have quoted from other sources, including any diagrams, charts, tables or graphs. □

I have clearly indicated the presence of all paraphrased material with appropriate references. □

I have acknowledged appropriately any assistance I have received in addition to that provided by my supervisor. □

I have not copied from the work of any other candidate. □

I have not used the services of any agency providing specimen, model or ghostwritten work in the preparation of this thesis/dissertation/extended essay/assignment/project/other submitted work. (See also section 2.4 of Statute XI on University Discipline under which members of the University are prohibited from providing material of this nature for candidates in examinations at this University or elsewhere: http://www.admin.ox.ac.uk/statutes/352-051a.shtml#_Toc28142348.) □

The project report does not exceed 50 pages (including all diagrams, photographs, references and appendices). □

I agree to retain an electronic copy of this work until the publication of my final examination result, except where submission in hand-written format is permitted. □

I agree to make any such electronic copy available to the examiners should it be necessary to confirm my word count or to check for plagiarism. □

Candidate's signature: …………………………………………….. **Date:** ………………………..

Abstract

The focus of this work is to determine whether a person's ultimate colour preference or colour combination preference is best found using Bayesian optimisation. With the history of attempts of their quantification and modelling in mind, colour and colour combination preferences are considered as an optimisation problem. The suitability of Bayesian optimisation to solve this problem is assessed through testing. The first of these tests evaluates the performance of Bayesian optimisation against other derivative-free methods for minimisation of test functions. Bayesian optimisation is found to perform the best for a small number of observations, and hence is the most suitable for interactive optimisation. The second test was intended to simulate a single colour preference optimisation problem. The prevalence of each colour in a set of paintings by an artist is observed compared to the average for a larger set of paintings. The most used colour is used as a surrogate for colour preference, and is found using Bayesian optimisation. For the four artists chosen as subjects, colours close to this most-used colour, as well as local maxima, were found. Both tests inspire confidence in the Bayesian optimisation algorithm for finding colour preferences or colour combination preferences. Finally, a method for interactive colour preference learning using Bayesian optimisation is put forward, and demonstrated. This is extended to a colour palette extension technique, which is also demonstrated. The resulting palettes are used throughout this report.

Acknowledgements

I would first like to thank my supervisor, Michael Osbourne, for his time and enthusiasm for my project title, as well as considerable guidance on the implementation of Bayesian optimisation. I am also grateful for his project proposal that resulted in this report; the unique application of Bayesian optimisation to colour preferences captured my interest and kept it.

Also, I would like to thank my colleages at Hertford college for their curiosity in the end results of my project, and the stimulating conversations we shared in the run-up to its completion.

Finally, I would like to acknowledge the support provided by my family throughout my years of study, and particularly during this last year. Thank you.

Contents

Chapter 1

Introduction

Colour perception has been exhaustively studied within the fields of Opthalmology, Psychology and Design since the discoveries of Newton in the 17th century. Eysenck (1941) suggested that colour preference may not be entirely subjective, as some agreement in ranking of basic, fully saturated colours was found. Influenced by this study, Granger (1955) produced results that supported this theory of objectivity, albeit using a small sample.

More recently, models for single colour and colour combination preference were developed by Ou et al. (2004) using both colour emotions and appearance attributes, which had reasonable success. Palmer and Schloss (2010) proposed their Ecological Valence Theory, which suggests a relationship between colour preference, and response to colour-associated objects. This was more sucessful that the previous modelling in explaining experimental data; the WAVE model accounted for 80% of variance in the colour preference data. The proposed link between evolved response to stimuli and affinity for its colour could explain the apparent element of objectivity in colour preferences.

While models may provide generic colour or colour combination preferences, a subjective element to colour aesthetics remains likely. For example, Ou et al. (2012) found some differences in colour perception for two age groups: the first including ages 20-30 and the second including ages over 60. In particular, affinity for all colours was lower for the older group, with a larger affinity discrepancy for a handful of colours. It is unknown whether this effect is caused by the physical impact of aging or some psychological effect. Hanafy and Sanad (2015) explored the effect of educational background on colour preference for both clothing and a living room. The two groups, information technology students and graphic design students, only significantly differed in preference for the living room situation. However, the groups were not particularly different in educational background; the subjects were all attending the same university. Also, Schloss and Palmer (2011) investigated the effects of university affiliation on colour preference, based on the Ecological Valence theory mentioned earlier.

Affinity for a colour was linked to feelings associated with universities which used the specified colour. Regardless of how differences in colour preferences arise, the assumption that some element of subjectivity in colour preference seems valid. This validates the pursuit of an optimisation method to obtain an ultimate colour preference for an individual.

1.1 Colour Preference as an Optimisation Problem

The RGB colour model is a common method for digital representation of colours. It is based on additive light: red, blue and green light add together in varying amounts to produce each colour. Their union produces white light, demonstrated in figure 1.1. The RGB representation of white is therefore $[255, 255, 255]$: the maximum values of each channel. The ability to represent colour as continuous space inside a cube is very useful for computation.

Figure 1.1. The union of red, green and blue light, which produces secondary colours magenta, yellow, and cyan, as well as white light. The RGB colour model is based on this addition of red, green and blue light. An RGB therefore has intensity values for red, green and blue light.

An individual's affinity for each colour or combination thereof can be treated as a black-box function, $g(\mathbf{x})$. Theoretically, the ultimate preference is at $\mathbf{x}_{\text{opt}}$, which is defined as the maximiser of $g(\mathbf{x})$:

$$\mathbf{x}_{\text{opt}} = \underset{\mathbf{x}\epsilon X}{\arg\max}\, g(\mathbf{x}). \tag{1.1}$$

The search space, X, is a hypercube with bounds $[0, 255]$ and $3c$ dimensions, where c is the number of colours in each combination. Furthermore, X is a set of concatenated RGB vectors; in the case where a favourite pair of colours is to be found, X is a set of 6-dimensional vectors.

A black-box function is a function whose analytic form is unknown. This definition means that gradients are unknown, and that the function could be non-convex, non-smooth or even discontinuous. The function of colour or colour combination affinity is almost certainly highly multimodal; a person generally likes a few different colours. The likely abundance of local maxima means that g is non-convex.

1.2 Global Optimisation

Global optimisation for convex functions, or local optimiation, is a well-understood field, with a vast amount of literature. Global optimisation of non-convex functions is quite different. An approximate global optimisation method is use of a local optimiser with multi-start initialisation (see section 2.3). However, the lack of knowledge of gradients associated with black-box functions is problematic. For this reason, the class of derivative-free global optimisers will be considered.

Bayesian optimisation belongs to this class. Its apparent efficiency with respect to number of function evaluations compared with other methods (Osborne et al., 2009) provided a rationale for its study in this project. The theory of Bayesian optimisation is detailed in chapter 2.

Two other derivative-free methods, Particle Swarm optimisation (PSO) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), are considered in chapter 3, where they are tested against Bayesian Optimisation.

1.3 Motivation

The colour preference problem is an excellent candidate to demonstrate the capabilities of Bayesian optimisation. An individual's affinity for colours or combinations thereof is both unknown and expensive to evaluate; affinity scoring is likely to become tedious quickly, hence the high cost.

The end goal of this work is to produce software capable of learning colour and colour combination preference. This software could be used extensively to aid design. For example, taking inspiration from Kita and Miyata (2016), it could be used to extend colour palettes. Instead of maintaining colour harmony, however, a bespoke set of preferred colours could be obtained. For professionals such as graphic or fashion designers, this could improve efficiency by reducing the time spent adjusting colours manually.

1.4 Overview

In this chapter, the colour preference was introduced as a global optimisation problem. Several methods, including Bayesian optimisation, were suggested to solve the problem. In addition, the value of software capable of colour or colour combination affinity optimisation was put forward.

The theory of Gaussian processes and Bayesian optimisation is presented in chapter 2, as well as details of their implementation in practice. A software package was created based on this work.

In chapter 3, the Bayesian optimisation software mentioned above has been tested alongside Particle Swarm optimisation and the Covariance Matrix Adaptation Evolution strategy using an assortment of multi-dimensional test functions. The results can be used to confirm that the software works, and determine whether Bayesian optimisation is the most suitable method for finding colour preferences based on a user's input.

Chapter 4 includes a non-interactive demonstration of the Bayesian optimisation algorithm when applied to a model for single colour preferences. Using a combined histogram of paintings, an artist's most used colour is sought. Accurate results reinforce confidence in the application of Bayesian optimisation to the colour preference problem.

A demonstration of interactive colour preference learning using Bayesian optimisation is presented in chapter 5. A technique for creating and extending colour palettes using this interactive method is also explained and demonstrated. Optimal colour palettes were created and applied to figures throughout this report.

This project is summarised and its success is detailed in chapter 6. Limitations and shortcomings of this work are discussed, alongside suggestions for further improvements to the colour preference learning software.

Chapter 2

Background

This chapter's purpose is primarily to explain the theory of Bayesian optimisation and its implementation in practice. This includes techniques used in each step of the Bayesian optimisation algorithm:

Algorithm 1 Bayesian optimisation

```
initialise $\mathbf{x}_0$
for $i = 0$ to $i = N$ do
    observe $\mathbf{y}_i$ at $\mathbf{x}_i$
    update Gaussian process $f$
    compute maximiser of the expected improvement, $\mathbf{x}_{i+1}$
end for
return $\mathbf{y}^{best}, \mathbf{x}^{best}$.
```

The objective function is evaluated at N points chosen using a predictive distribution. This intelligent choosing allows for a tight budget on function evaluations, which can be a very efficient use of time in the case of an expensive objective function.

Bayesian optimisation is named for its use of Bayes' rule; prior beliefs and evidence are used to obtain predictions, which are used to determine the location of search points. This name is attributed to Močkus (1975), who minimised a function's expected deviation from the extremum to find the next search point. More recently, the use of a Gaussian process to obtain an a posteriori distribution has become popular. A comprehensive theory of Gaussian processes was written by Rasmussen and Williams (2006). Lizotte (2008) and Osborne et al. (2009) implemented and tested Gaussian process global optimisation using multi-dimensional test functions, and found that it performed well when the number of allowed function evaluations is limited. Osborne (2010) went on to apply this work to weather sensor networks. Also, the review of Bayesian optimisation using Gaussian processes by Shahriari et al. (2015) provides an excellent big-picture of the technique. The implementation of Gaussian processes will be discussed in section 2.1.

Interactive Bayesian optimisation, where a user supplies the data, is of particular interest. Brochu

(2010) used Bayesian optimisation to learn user preferences for graphics and animation. The indication of preferences rather than scoring, and the learning of hyperparameters over all user sessions were found to be successful. Also, Bayesian optimisation was applied to a cookie recipe by Solnik et al. (2017), where each different cookie was scored on a scale from 1 to 7 by a group of test subjects. This application to a complex real world problem demonstrates the capabilities of Bayesian optimisation.

2.1 Gaussian Processes

As defined by Rasmussen and Williams (2006), "a Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution". One can be fitted to a number of observations, resulting in a non-parametric predictive model. A Gaussian process regression is shown in figure 2.1. It is important to understand that for each input value, the Gaussian process is a univariate Gaussian probability density function. The predictions refer to the mean of the Gaussian at every input value. The shaded area represents the 95% confidence interval of the Gaussian at every input value.

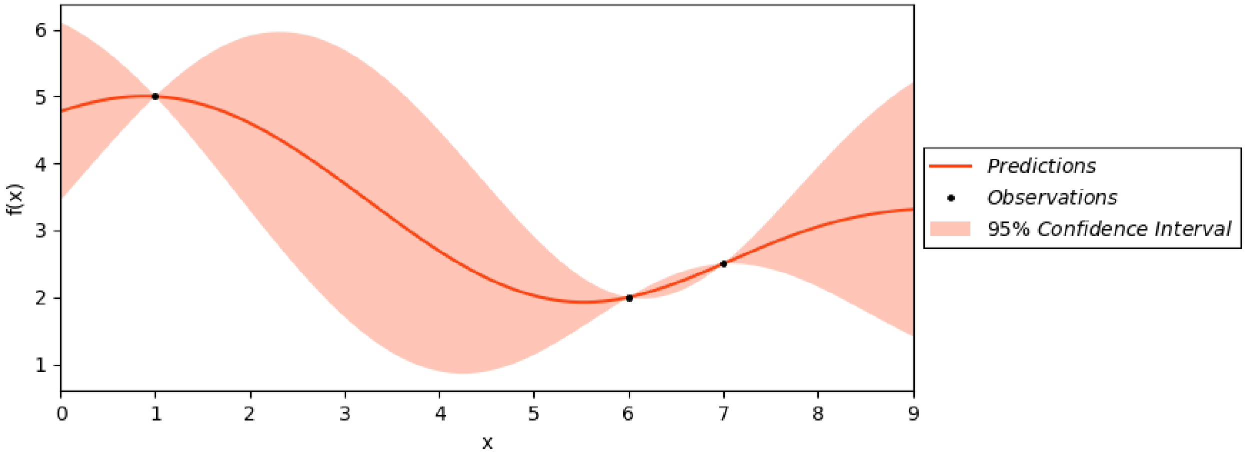


Figure 2.1. A Gaussian process fitted for three observations, where its mean gives predictions for the input space. Uncertainty in predictions are represented using the 95% confidence interval of the Gaussian process

The lack of parameters required by a Gaussian process is advantageous, as it negates the chance that a predictive model of an unsuitable form could be chosen, for example, the choice of a linear model to predict a relationship which is actually square.

A Gaussian process can be fully defined for a single input $\mathbf{x}_i$ by a covariance function and a mean function:

$$f(\mathbf{x}_i) \sim \mathcal{GP}(m(\mathbf{x}_i), k(\mathbf{x}_i, \mathbf{x}_i')). \tag{2.1}$$

The full covariance matrix of $\mathbf{x}$ has the form

$$k(\mathbf{x},\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & \dots & k(\mathbf{x}_1,\mathbf{x}_N) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & \dots & k(\mathbf{x}_2,\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N,\mathbf{x}_1) & k(\mathbf{x}_N,\mathbf{x}_2) & \dots & k(\mathbf{x}_N,\mathbf{x}_N) \end{bmatrix} \text{(Roberts et al., 2013).} \tag{2.2}$$

These functions should be chosen to reflect the prior beliefs about the objective function. This will be explored later in sections 2.1.1 and 2.1.2.

When allowing for noisy observations, the Gaussian process evaluated at each point plus some noise, ϵ_i, with variance σ_n^2 is equal to each observation, $\mathbf{y}_i$. The expression for a single observation is

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i. \tag{2.3}$$

This is equivalent to expressing the distribution of N observations as a Gaussian with mean $f(\mathbf{x})$ and diagonal covariance of magnitude equal to the noise variance:

$$P(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 I_N). \tag{2.4}$$

Using the definition of Gaussian process $f(\mathbf{x}_i)$ in equation 2.1, the expression in equation 2.4 can be written as

$$P(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N). \tag{2.5}$$

The joint distribution of observations and predictions is expressed as:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}) \\ m(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N & k(\mathbf{x},\mathbf{x}^*) \\ k(\mathbf{x},\mathbf{x}^*)^T & k(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix} \right). \tag{2.6}$$

Superscript $*$ denotes a test variable. Following the procedure by Rasmussen and Williams (2006), it can be proven that the predictive distribution is

$$P(\mathbf{f}^*|\mathbf{x},\mathbf{y},\mathbf{x}^*) \sim \mathcal{N}(\mu(\mathbf{x}^*), \mathbf{\Sigma}(\mathbf{x}^*)), \tag{2.7}$$

where the predictive mean and covariance are expressed as

$$\mu(\mathbf{x}^*) = m(\mathbf{x}^*) + k(\mathbf{x}^*,\mathbf{x})[k(\mathbf{x},\mathbf{x}) + \sigma_n^2 I_N]^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{2.8}$$

and

$$\mathbf{\Sigma}(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{x})[k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N]^{-1} k[\mathbf{x}, \mathbf{x}^*] \tag{2.9}$$

respectively.

The result is a predictive distribution for all test inputs. Both a predicted mean output and a measure of certainty of this mean, the covariance, have been obtained. This means that for each valid test input, a predictive distribution of each output can be evaluated using just the observations, prior mean and covariance functions, and speculative value of noise variance. Note that the standard deviation of the marginals of $\mathbf{f}^*$ can be found by taking the square root of the diagonal elements of the covariance matrix:

$$\sigma(\mathbf{x}^*) = \sqrt{\mathsf{diag}(\mathbf{\Sigma}(\mathbf{x}^*))}. \tag{2.10}$$

In practice, the inverse kernel plus noise should be calculated using the Cholesky factoristion.

2.1.1 The Mean Function

The prior mean function $m(\mathbf{x})$ is selected with the consideration that, far from the observations, the predictive mean will return to its value. Prior beliefs about features of the objective function can influence this choice, which is commonly zero or a constant (Osborne, 2010). In the case of a constant mean function, the constant would be considered a hyperparameter. To achieve maximal generality, the mean function chosen for this project is

$$m(\mathbf{x}_i) = \frac{1}{N} \sum_{i=0}^{N} \mathbf{y}_i, \tag{2.11}$$

the mean of the observed function values.

2.1.2 The Covariance Function

The covariance or kernel function may be chosen to reflect prior beliefs about the shape of an objective function. For example, a squared exponential kernel work well in prediction of a function known to be smooth, and a periodic kernel would be suitable for prediction of a function known to be periodic, such as tide height. Kernels can be combined using multiplication or addition to achieve multiple properties (Duvenaud, 2014).

A popular covariance function, and the choice for this project, is the squared exponential kernel. For a person's colour preferences and most other applications, the assumption of smoothness seems reasonable. The squared exponential kernel is

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sigma_o^2 \exp\left(\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2l^2}\right) \tag{2.12}$$

for $\mathbf{x}_i$ with only one dimension.

Two hyperparameters are introduced here: lengthscale l and outer standard deviation, σ_o. These are both squared to ensure that the resulting covariance matrix is positive semi-definite. The values of these hyperparameters can affect the shape of the resulting Gaussian process greatly. In figure 2.2, regressions based on noisy observations of a known function, $f(x) = \sin(x) + \frac{1}{2}\cos(\frac{x}{2})$, are plotted with varying hyperparameters.

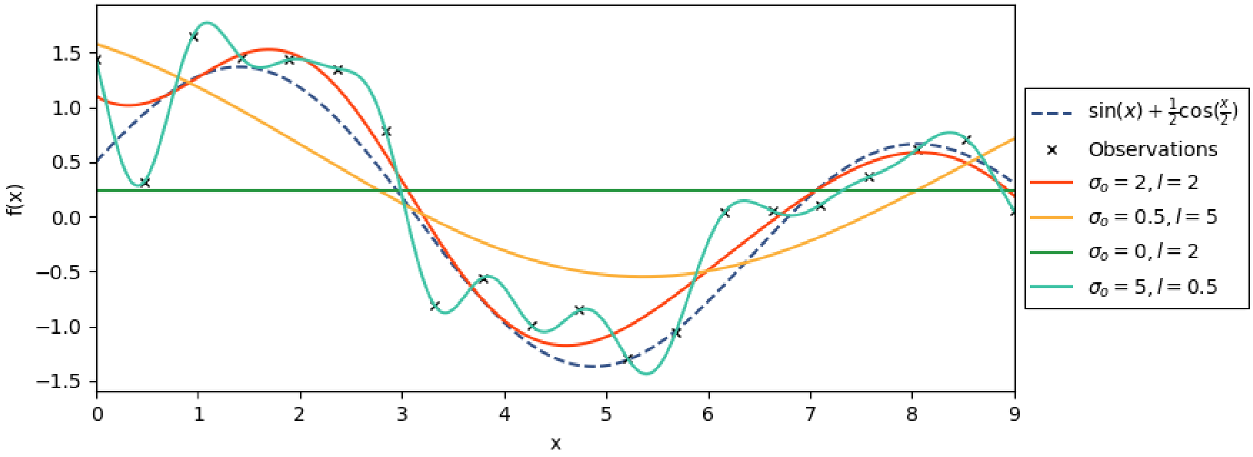


Figure 2.2. Gaussian process regressions, using a squared exponential kernel with different outside variance and lengthscale values, based on noisy observations of an known function. The additive noise is zero-mean Gaussian with $\sigma_n = 0.5$. The colour palette used in this graph was selected using interactive colour preference learning (figure 5.3).

A zero outside variance reduces the kernel function to zero. The resulting predictive distribution at each point is a dirac delta function centred at the mean function value, which in this case is the mean of the observations. A small outside variance paired with a longer lengthscale appears to cause underfitting, whilst a larger outside variance paired with a shorter lengthscale seems to cause overrfitting. Out of the 4 combinations tried, the best predictions are produced when $\sigma_o = 2$ and $l = 2$. These were chosen using prior knowledge of the function. However, when the analytic form of a function is unknown, hand tuning like this becomes more of a trial and error process. Instead, it is common practice to automatically tune hyperparameters using optimisation. This is explained in section 2.1.3.

The kernel in equation 2.12 is only suitable for scalar inputs, and is therefore only applicable to objective functions with one input dimension. However, the Gaussian process must be suitable for multi-dimensional inputs, given that $\mathbf{x}$ will represent colour using the RGB model. The squared exponential kernel can be adapted a D-dimensional input by multiplying the squared exponential part for each dimension together to find the combined squared exponential:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sigma_o^2 \exp\left(\sum_{d=1}^{D} -\frac{(\mathbf{x}_{id} - \mathbf{x}_{jd})^2}{2l_d^2}\right). \tag{2.13}$$

This approach introduces an extra lengthscale for each dimension. This is necessary to allow the Gaussian process enough freedom to provide the best possible predictions.

2.1.3 Optimal Hyperparameters

As seen in subsection 2.1.2, the squared exponential kernel function introduces an outside variance and D lengthscales, which can be hand-tuned. Especially when D is more than 1, hand-tuning becomes difficult or impossible, as well as tedious. An alternative is to choose the hyperparameters by optimisation. A simple way to do this is by maximisation of marginal likelihood of the predictions. This is equivalent to minimisation of the negative log-likelihood,

$$-\log(\mathbf{y}|\mathbf{x}) = \frac{1}{2}(\mathbf{y} - m(\mathbf{x}))^T (k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N)^{-1} (\mathbf{y} - m(\mathbf{x})) + \frac{1}{2}\log|k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N| + \frac{N}{2}\log 2\pi. \tag{2.14}$$

In order to simplify notation, two new matrices are introduced. K_y can replace the kernel plus noise:

$$K_y = k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N \tag{2.15}$$

and α can replace the product of its inverse with observation deviation:

$$\alpha = (k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I_N)^{-1} (\mathbf{y} - m(\mathbf{x})). \tag{2.16}$$

A gradient based-method, could be used to minimise the negative log-likelihood, due to the ease of obtaining gradients through differentiation of the explicit function. The optimisation method used in practice is multi-start L-BFGS-B. Further details can be found in section 2.3.

The partial derivative of negative log-likelihood with respect to a general hyperparameter can be expressed as

$$\frac{\partial(-\log(\mathbf{y}|\mathbf{x}))}{\partial\theta_\zeta} = \frac{1}{2}\text{trace}(K_y^{-1} - \alpha\alpha^T)\frac{\partial K_y}{\partial\theta_\zeta}, \tag{2.17}$$

(Rasmussen and Williams, 2006), where θ is a vector of hyperparameters of form

$$\theta = \begin{bmatrix} \sigma_o & l_1 & \dots & l_D & \sigma_n \end{bmatrix}. \tag{2.18}$$

The noise standard deviation σ_n can be treated as a hyperparameter here, which allows the most suitable value to be chosen automatically in the case where the function evaluations are noisy. The derivation of partial derivatives $\frac{\partial K_y}{\partial \theta_\varsigma}$ follows below. The outer variance derivative,

$$\frac{\partial K_y}{\partial \sigma_o} = 2\sigma_o \exp\left(\sum_{d=1}^{D} -\frac{(\mathbf{x}_{id} - \mathbf{x}_{jd})^2}{2l_d^2}\right), \tag{2.19}$$

is easy to calculate, as the exponential part is considered a constant.

The derivation of lengthscale gradients are more involved. First, express the derivative as a constant multiplied by the partial derivative of the dimension in question, $\bar{d}$:

$$\frac{\partial K_y}{\partial l_{\bar{d}}} = \sigma_o^2 \exp\left(\sum_{d=1,d\neq\bar{d}}^{D} -\frac{(\mathbf{x}_{id} - \mathbf{x}_{jd})^2}{2l_d^2}\right) \frac{\partial}{\partial l_{\bar{d}}} \exp\left(-\frac{(\mathbf{x}_{i\bar{d}} - \mathbf{x}_{j\bar{d}})^2}{2l_{\bar{d}}^2}\right). \tag{2.20}$$

Next, calculate the partial derivative with respect to $l_{\bar{d}}$:

$$\frac{\partial}{\partial l_{\bar{d}}} \exp\left(-\frac{(\mathbf{x}_{i\bar{d}} - \mathbf{x}_{j\bar{d}})^2}{2l_{\bar{d}}^2}\right) = \frac{(\mathbf{x}_{i\bar{d}} - \mathbf{x}_{j\bar{d}})^2}{l_{\bar{d}}^3} \exp\left(-\frac{(\mathbf{x}_{i\bar{d}} - \mathbf{x}_{j\bar{d}})^2}{2l_{\bar{d}}^2}\right). \tag{2.21}$$

The end result is

$$\frac{\partial K_y}{\partial l_{\bar{d}}} = \frac{(\mathbf{x}_{i\bar{d}} - \mathbf{x}_{j\bar{d}})^2}{l_{\bar{d}}^3} k(\mathbf{x}_i, \mathbf{x}_j), \tag{2.22}$$

which is found by substituting 2.21 into 2.20.

The derivative with respect to noise standard deviation is straightforward.:

$$\frac{\partial K_y}{\partial \sigma_n} = 2\sigma_n I_N. \tag{2.23}$$

These partial derivatives of K_y can be substituted back into equation 2.17 to find the partial derivatives of negative log-likelihood with respect to the hyperparameters and noise standard deviation. The likelihood function can be evaluated without the Gaussian process being calculated explicitly, so the hyperparameters can be optimised in advance.

In figure 2.3, a Gaussian process regression is plotted for noisy observations of the same functions as in figure 2.2. The optimal values are: $\sigma_o = 0.838$; $l = 1.062$; $\sigma_n = 0.411$. The predictions are quite close to the true function, despite the high noise level.

It should be noted that difficulties may arise during hyperparameter optimisation as a result of the likely multimodal negative log-likelihood as a function of outer variance and lengthscales. Each different interpretation of the observations may correspond to a local minimum. For example, the data in figure 2.3 might be interpreted as a highly multimodal periodic function where resulting predictions

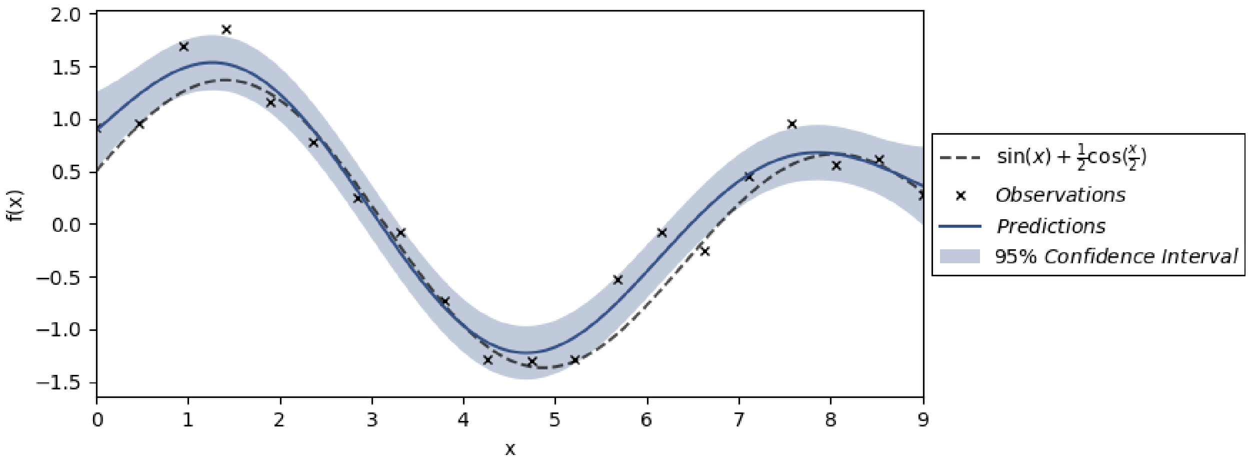


Figure 2.3. A predictive distribution based on noisy observations of an known function, using a squared exponential kernel with optimal hyperparameter values. The additive noise is zero-mean Gaussian with $\sigma_n = 0.5$.

are highly noise dependent, or the opposite. The multistart local optimiser in section 2.3 will likely succeed in finding the most appropriate hyperparameters, conditional on there being enough coverage of the function by the initialisations. The chosen number of initialisation per hyperparameter is 10.

2.2 Bayesian Optimisation

It has been established how a Gaussian process regression can provide a predictive model. Now, this model can be used to choose search points. To gain the best possible information from an evaluation, an acquisition function can be maximised. This acquisition function can be defined based on the desired behaviour of the algorithm.

The success of Bayesian optimisation lies in a balance between exploration and exploitation. In minimisation, using a good acquisition function, the objective function would be evalutated where both the predicted function value is low and the predictied variance is high.

2.2.1 Probability of Improvement

One candidate is the probability of improvement acquisition function, which prioritises areas where there is a higher chance of improvement. During minimisation, when the smallest observation so far, y^-, is predicted to be improved upon at point $\mathbf{x}^*{}_i$, a reward of 1 unit is assigned. Otherwise, there is no reward. This incentivises the program to seek an improvement in the best observation. This reward is represented by the utility function:

$$u(\mathbf{x}^*{}_i) = \begin{cases} 1 & \text{if } f(\mathbf{x}^*{}_i) < y^- \\ 0 & \text{if } f(\mathbf{x}^*{}_i) \geq y^- \end{cases} . \tag{2.24}$$

The expected value of the reward at each value of x can be found by integration of the expected reward over all predicted function values. This is a simple task when predictions are based on a Gaussian process regression, because the normal distribution can be integrated easily like so:

$$a_{PI}(\mathbf{x_i^*}) = \int_{-\infty}^{y^-} \mathcal{N}(\mathbf{f}^*; \mu(\mathbf{x}^*{}_i), \sigma(\mathbf{x}^*{}_i))d\mathbf{f}^* = \Phi(\frac{y^- - \mu(\mathbf{x}^*{}_i)}{\sigma(\mathbf{x}^*{}_i)}). \tag{2.25}$$

However, this approach is not ideal. Every improvement, large or small, is treated equally. This could mean that the program is mainly exploitative: the rewards for negligibly improving the current minimum locally and finding a new, much better minimum are the same.

2.2.2 Expected Improvement

Instead, the reward could be equal to the magnitude of improvement:

$$u(\mathbf{x}^*{}_i) = \begin{cases} y^- - f(\mathbf{x}^*{}_i) & \text{if } f(\mathbf{x}^*{}_i) < y^- \\ 0 & \text{if } f(\mathbf{x}^*{}_i) \geq y^- \end{cases} . \tag{2.26}$$

When this reward is integrated over all possible values, the expected improvement acquisition function is derived, which is

$$a_{EI}(\mathbf{x}^*) = \int_{-\infty}^{y^-} (y^- - f^*)N(f^*; \mu(\mathbf{x}^*{}_i), \sigma(\mathbf{x}^*{}_i)df^*. \tag{2.27}$$

The properties of a Gaussian allow for relatively easy integration, resulting in

$$a_{EI}(\mathbf{x}^*) = (y^- - \mu(\mathbf{x}^*{}_i))\Phi(y^-; \mu(\mathbf{x}^*{}_i), \sigma(\mathbf{x}^*{}_i)) + \sigma(\mathbf{x}^*{}_i)\mathcal{N}(y^-; \mu(\mathbf{x}^*{}_i), \sigma(\mathbf{x}^*{}_i)). \tag{2.28}$$

An equivalent expression,

$$a_{EI}(\mathbf{x}^*{}_i) = \sigma(\mathbf{x}^*{}_i)(z\Phi(z) + \mathcal{N}(z; 0, 1)), \tag{2.29}$$

is obtained by substituting z,

$$z = \frac{y^- - \mu(\mathbf{x}^*{}_i)}{\sigma(\mathbf{x}^*{}_i)}. \tag{2.30}$$

This step allows the expected improvement acquisition function to be expressed in terms of a standard normal distribution.

The decision to use expected improvement rather than probability of improvement as the acquisition function is supported by tests conducted by Shahriari et al. (2015). The Bayesian optimisation algorithm using expected improvement is clearly superior in comparison of absolute error over a number of evaluations.

2.2.3 Maximising the Acquisition Function

The multi-start L-BFGS-B algorithm detailed in subsection 2.3 could be used again to maximise the expected improvement. As with the hyperparameter optimisation, gradients of the acquisiton function should be derived. The procedure by Rana et al. (2017) can be followed, starting with the expected improvement derivative as a function of the predictive standard deviation derivative and z derivative:

$$\frac{\partial a_{EI}}{\partial \mathbf{x^*}_{\bar{d}}} = (z\Phi(z) + \mathcal{N}(z;0,1))\frac{\partial \sigma(\mathbf{x}^*)}{\partial \mathbf{x^*}_{\bar{d}}} + \sigma(\mathbf{x}^*)\Phi(z)\frac{\partial z}{\partial \mathbf{x^*}_{\bar{d}}}. \tag{2.31}$$

The derivative of predictive standard deviation with respect to $\mathbf{x}^*$ can be evaluated by differentiating equation 2.9:

$$\frac{\partial \sigma(\mathbf{x}^*)}{\partial \mathbf{x^*}_{\bar{d}}} = -\frac{1}{\sigma(\mathbf{x}^*)}(\frac{\partial k(\mathbf{x},\mathbf{x}^*)^T}{\partial \mathbf{x^*}_{\bar{d}}}K_y^{-1}k(\mathbf{x},\mathbf{x}^*)). \tag{2.32}$$

The partial of z with respect to $\mathbf{x}^*$ can be found in a similar way:

$$\frac{\partial z}{\partial \mathbf{x^*}_{\bar{d}}} = \frac{1}{\sigma(\mathbf{x}^*)}(\frac{\partial k(\mathbf{x},\mathbf{x}^*)^T}{\partial \mathbf{x^*}_{\bar{d}}}K_y^{-1}\mathbf{y} - z\frac{\partial \sigma(\mathbf{x}^*)}{\partial \mathbf{x^*}_{\bar{d}}})\text{(Rana et al., 2017)}. \tag{2.33}$$

Substitute equations 2.32 and 2.33 back into equation 2.31 to obtain the gradient in terms of the derivative of k with respect to the $\bar{d}$-th dimension of $\mathbf{x}$:

$$\frac{\partial a_{EI}}{\partial \mathbf{x^*}_{\bar{d}}} = -\frac{\mathcal{N}(z;0,1)}{\sigma(\mathbf{x}^*)}(\frac{\partial k(\mathbf{x},\mathbf{x}^*)^T}{\partial \mathbf{x^*}_{\bar{d}}}K_y^{-1}k(\mathbf{x},\mathbf{x}^*)) + \Phi(z)\frac{\partial k(\mathbf{x},\mathbf{x}^*)^T}{\partial \mathbf{x^*}_{\bar{d}}}K_y^{-1}\mathbf{y}. \tag{2.34}$$

The remaining step is to find the partial derivative of $k(\mathbf{x},\mathbf{x}^*)^T$ with respect to each dimension of $\mathbf{x}^*$. Recall that the structure of the kernel in equation 2.2, which can be reproduced for $k(\mathbf{x},\mathbf{x}^*)^T$:

$$k(\mathbf{x},\mathbf{x}^*)^T = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}^*) & k(\mathbf{x}_2,\mathbf{x}^*) & . & . & . & k(\mathbf{x}_N,\mathbf{x}^*) \end{bmatrix}. \tag{2.35}$$

Each element of this kernel is the multi-dimensional squared exponential:

$$k(\mathbf{x}_i,\mathbf{x}^*)^T = k(\mathbf{x}^*,\mathbf{x}_i) = \sigma_o^2 \exp\left(\sum_{d=1}^{D} -\frac{(\mathbf{x}_d^* - \mathbf{x}_{id})^2}{2l_d^2}\right). \tag{2.36}$$

Now, each element of the partial derivative with respect to a generic dimension of $\mathbf{x}$, $\bar{d}$, can be evaluated:

$$\frac{\partial k(\mathbf{x}^*, \mathbf{x}_i)}{\partial \mathbf{x}^*_{\bar{d}}} = \sigma_o^2 \frac{-(\mathbf{x}^*_{\bar{d}} - \mathbf{x}_{i\bar{d}})}{l^2_{\bar{d}}} \exp\left(\sum_{d=1}^{D} -\frac{(\mathbf{x}^*_d - \mathbf{x}_{id})^2}{2l^2_d}\right) \tag{2.37}$$

Note the similarity to the derivation of lengthscale gradients in section 2.1.3.

A single element of the simplified gradient can be expressed in terms of the kernel as

$$\frac{\partial k(\mathbf{x}^*, \mathbf{x}_i)}{\partial \mathbf{x}^*_{\bar{d}}} = k(\mathbf{x}^*, \mathbf{x}_i) \frac{\mathbf{x}_{i\bar{d}} - \mathbf{x}^*_{\bar{d}}}{l^2_{\bar{d}}}. \tag{2.38}$$

The full gradient can be represented by a matrix product, where the quotient $\frac{\mathbf{x}_{i\bar{d}} - \mathbf{x}^*_{\bar{d}}}{l^2_{\bar{d}}}$ is assembled into a diagonalised matrix, Q:

$$\frac{\partial k(\mathbf{x}^*, \mathbf{x}_i)}{\partial \mathbf{x}^*_{\bar{d}}} = k(\mathbf{x}^*, \mathbf{x}) \begin{bmatrix} \frac{\mathbf{x}_{1\bar{d}} - \mathbf{x}^*_{\bar{d}}}{l^2_{\bar{d}}} & 0 & \dots & 0 \\ 0 & \frac{\mathbf{x}_{2\bar{d}} - \mathbf{x}^*_{\bar{d}}}{l^2_{\bar{d}}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\mathbf{x}_{N\bar{d}} - \mathbf{x}^*_{\bar{d}}}{l^2_{\bar{d}}} \end{bmatrix} = k(\mathbf{x}^*, \mathbf{x}) \ Q. \tag{2.39}$$

The final expected improvement gradient,

$$\frac{\partial a_{EI}}{\partial {\mathbf{x}^*}_{\bar{d}}} = k(\mathbf{x}^*, \mathbf{x}) \ Q \ K_y^{-1} \left(\Phi(z)\mathbf{y} - \frac{\mathcal{N}(z; 0, 1)}{\sigma(\mathbf{x}^*)} k(\mathbf{x}, \mathbf{x}^*)\right), \tag{2.40}$$

can be found by substituting 2.39 into 2.34. Note that the result is a scalar value.

There are often many good places to evaluate a function, the best of which can be difficult to find. The expected improvement is usually higher where predictive variance is high. This is the case between far-apart observations. Often, this results in a highly multimodal expected improvement. The multistart local optimiser in section 2.3 works fairly well as is with 10 initialisations, usually finding the global maximum, but sometimes finding local maxima which are nearly as good as the former. Occasionally, a situation arises where there is a sharp peak in expected improvement very close to the current best value. In order to ascend these peaks, exploitation points can be added to the multi-start initialisation set, very close to the current best.

2.3 Global Optimisation using Multi-Start BFGS

As mentioned in subsection 2.1.3, the negative log-likelihood should be minimised with respect to hyperparameters and noise standard deviation to fit a Gaussian process. This presents the challenge of finding the global optimum of a potentially non-convex function. A relatively good approximation to a global optimiser is a local optimiser with multiple initialisations. Many local extrema are found, out of which the best result is chosen. With enough points evenly spread out, the global optimum is usually found. A Latin Hypercube is suitable for providing a spread out set of initialisation points (McKay et al., 1979).

Also in sections 2.1.3 and 2.2.3 respectively, it was established that gradients for the negative log-likelihood and expected improvement could be explicitly calculated. For this reason, a gradient-based optimiser should be selected.

The Broyden-Fletcher-Goldfarb-Shanno algorithm is an excellent candidate (Nocedal and Wright, 2000), and has been considered based on reports of good performance in practice. BFGS is viewed to be the standard quasi-Newton method; it outperforms similar algorithms with noisier function evaluations (Fletcher, 1987). In practice, minimisation of negative log-likelihood will be undertaken by the L-BFGS-B algorithm, which is BFGS with limited memory and the ability to bound the optimisation space.

Chapter 3

Optimisation of Test Functions

The aim of this chapter is to test Bayesian optimisation against two other global optimisation methods: Particle Swarm Optimisation and the Covariance Matrix Adaptation Evolution Strategy. Brief overviews of these methods are given in sections 3.1 and 3.2 respectively. There exist many functions intended for testing of global optimisation. These functions have a global minima, and often, local minima. The performance of an optimisation can be measured by comparing the discovered minimum to the true global minimum.

Success of an optimiser in the context of this project can be assessed using several criteria. Efficient use of function evaluations is very important, due to the high cost of user input. Also, good performance for multimodal objective functions is essential: there are likely to be many peaks at favourable colours or combinations. Software to find a colour pair preference would involve a six-dimensional input, so good performance for functions with up to six dimensions is desirable. Testing for higher dimensional inputs would be sensible if preference for a larger combination of colours was to be found.

3.1 Particle Swarm Optimisation

In 1995, Kennedy and Eberhart (1995) intoduced the particle swarm method for optimisation. It was based partly on behaviour such as bird flocking and fish schooling. A swarm of particles is defined by positions and velocities. At each iteration, each particle's velocity is updated based on the global best function value, and the its personal best. Each position is updated using the equation of motion (Marini and Walczak, 2015).

3.2 Covariance Matrix Adaptation Evolution Strategy

Hansen (2006) has been one of the main contributing forces behind the Covariance Matrix Adaptation Evolution Strategy. During optimisation, search points are chosen by sampling a multivariate normal distribution. Based on the function values at these search points, the mean and covariance of the search distribution are updated. New search points are found by sampling, and the process is repeated (Hansen, 2016).

3.3 Test Functions

In order to adequately test the optimisation algorithms, the test function set should include multimodal and high dimensional functions. An example of a highly multimodal test function is the Ackley function. In figure 3.1 it is shown for 2 input dimensions, but there is no limit on its dimensionality in truth. The full set of functions is displayed in table 3.1.

Name	Dimensions	Search Space	Global Minimum
Branin	2	$[-5, 10] \times [0, 15]$	0.397887
Six-Hump Camel	2	$[-5, 5]^2$	-1.0316
Goldstein-Price	2	$[-5, 5]^2$	3
Hartmann	3	$[0, 1]^3$	-3.86278
Hartmann	6	$[0, 1]^6$	-3.32237
Shekel 5	4	$[0, 10]^4$	-10.1532
Shekel 7	4	$[0, 10]^4$	-10.4029
Shekel 10	4	$[0, 10]^4$	-10.5364
Shubert	2	$[-10, 10]^2$	-186.7309
Rastrigrin	2	$[0, 10]^2$	0
Ackley	2	$[-32.8, 32.8]^2$	0
Ackley	5	$[-32.8, 32.8]^5$	0

Table 3.1. The set of functions used to test the three optimisation methods, based on information from the simulation library by Surjanovic and Bingham.

3.4 Methodology

Each optimisation method has been applied to each test function 5 times, with a budget of 20 function evaluations per dimension. For CMA-ES, the population size has been changed from a default of 6 to 2 per input dimensions, while for PSO, the swarm size has been reduced from 100 to 5 per dimension. Otherwise, too many of the allowed function evaluations would be used in their initialisations. In the case of PSO, the default number of function evaluations for initialisation greatly exceeds the budget.

CMA-ES requires an additional input: initial step size or standard deviation. The optimum should

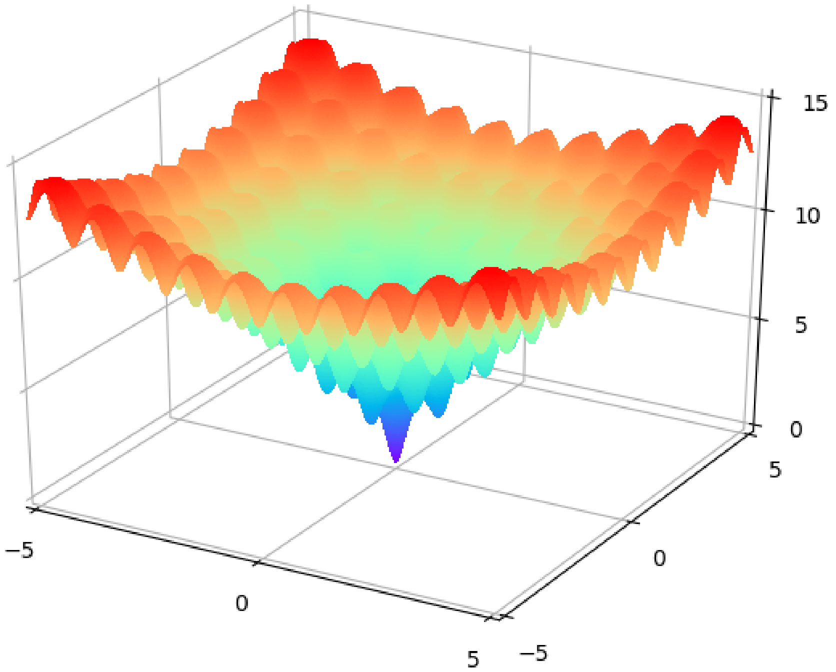


Figure 3.1. The Ackley function evaluated on the square $[-5, 5]^2$.

be expected to lie within 3 standard deviations of the mean, so it was set to one sixth of the test boundary width.

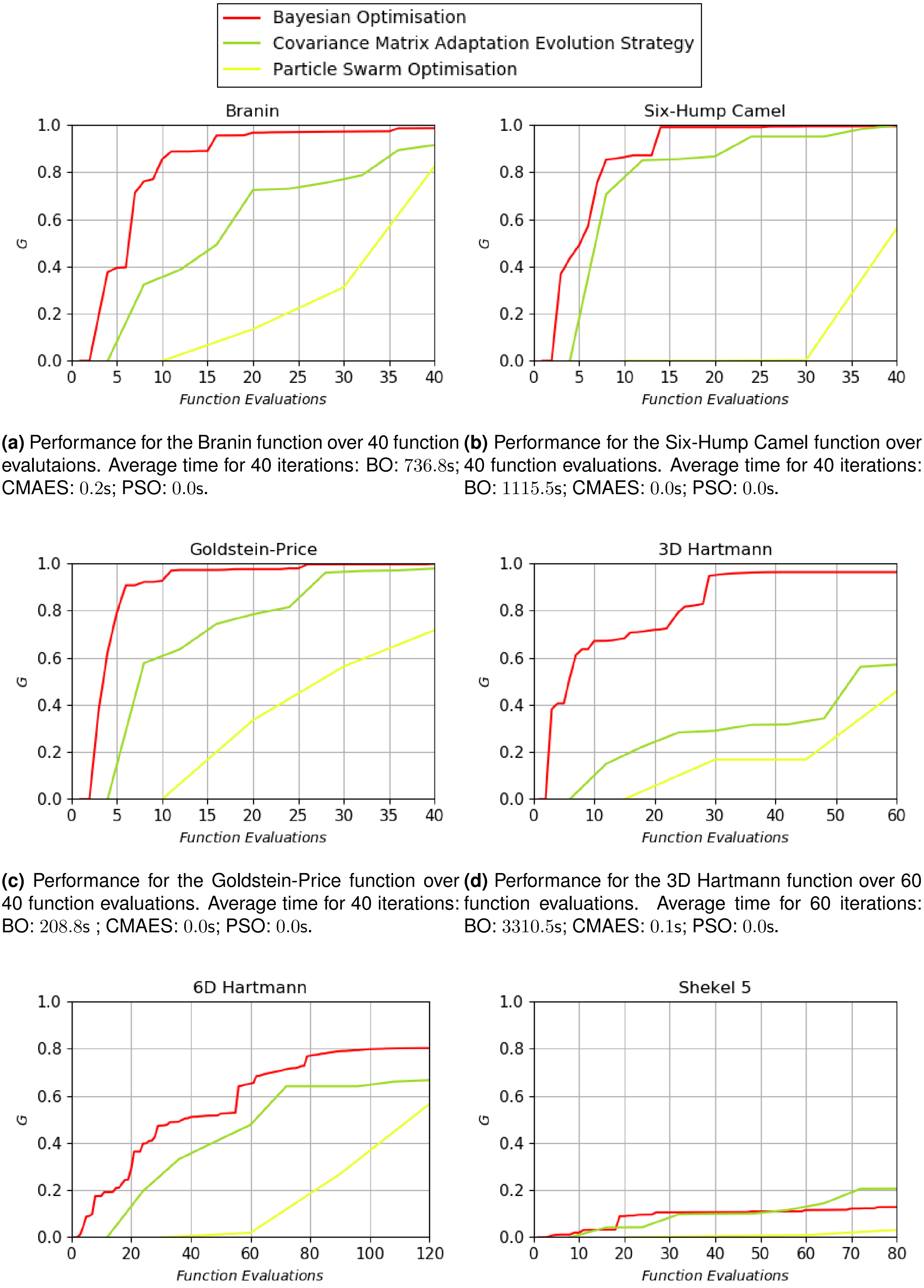
In the case of both Bayesian optimisation and CMA-ES, the initial input is a random point inside the test boundary. For PSO, the starting points are randomly initialised by the program. Of course, a 'lucky' initial value close to the minimum could result in an easier optimisation. For this reason, the metric used to compare methods is not proximity to the global minimum, but

$$G = \frac{f_{\textit{init}} - f_{\textit{best}}}{f_{\textit{init}} - f_{\textit{opt}}}, \tag{3.1}$$

which gives weight to the improvement of the obtained minimum over the course of the optimisation. As the best value decreases and tends towards the true minimum, G approaches its maximum value of 1. The mean of G has been observed as each optimisation progresses for each test function, alongside the mean time for completion.

3.5 Results

Overall, Bayesian optimisation has performed well compared to the other methods, especially in the first few iterations. For the three easiest to optimise 2-dimensional functions: Branin 3.2a, Six-Hump Camel 3.2b, and Goldstein-Price 3.2c, Bayesian optimisation has been very succesful. In each plot,



(a) Performance for the Branin function over 40 function evalutaions. Average time for 40 iterations: BO: 736.8s; CMAES: 0.2s; PSO: 0.0s.

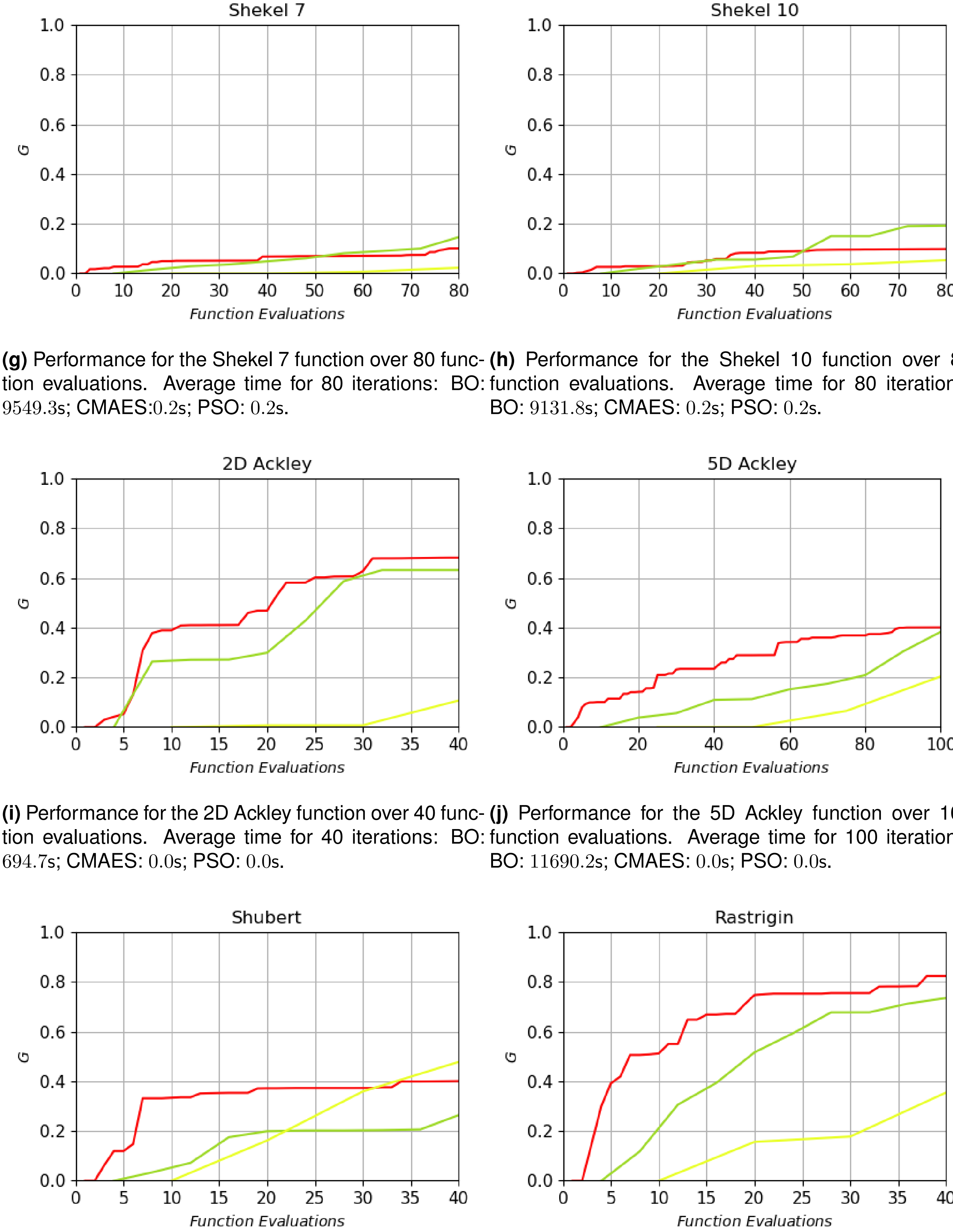
(b) Performance for the Six-Hump Camel function over 40 function evaluations. Average time for 40 iterations: BO: 1115.5s; CMAES: 0.0s; PSO: 0.0s.

(c) Performance for the Goldstein-Price function over 40 function evaluations. Average time for 40 iterations: BO: 208.8s ; CMAES: 0.0s; PSO: 0.0s.

(d) Performance for the 3D Hartmann function over 60 function evaluations. Average time for 60 iterations: BO: 3310.5s; CMAES: 0.1s; PSO: 0.0s.

(e) Performance for the 6D Hartmann function over 120 function evaluations. Average time for 60 iterations: BO: 18463.4s; CMAES: 0.1s; PSO: 0.1s.

(f) Performance for the Shekel 5 function over 80 function evaluations. Average time for 80 iterations: BO: 10846.6s; CMAES: 0.0s; PSO: 0.0s.



(g) Performance for the Shekel 7 function over 80 function evaluations. Average time for 80 iterations: BO: 9549.3s; CMAES:0.2s; PSO: 0.2s.

(h) Performance for the Shekel 10 function over 80 function evaluations. Average time for 80 iterations: BO: 9131.8s; CMAES: 0.2s; PSO: 0.2s.

(i) Performance for the 2D Ackley function over 40 function evaluations. Average time for 40 iterations: BO: 694.7s; CMAES: 0.0s; PSO: 0.0s.

(j) Performance for the 5D Ackley function over 100 function evaluations. Average time for 100 iterations: BO: 11690.2s; CMAES: 0.0s; PSO: 0.0s.

(k) Performance for the Shubert function over 40 function evaluations. Average time for 40 iterations: BO: 443.0s; CMAES: 0.0s; PSO: 0.0s.

(l) Performance for the Rastrigin function over 40 function evaluations. Average time for 40 iterations: BO: 545.3s; CMAES: 0.0s; PSO: 0.0s.

Figure 3.2. Plots showing the mean evolution of G over the course of optimisation by three different methods: Bayesian optimisation, Particle Swarm optimisation, and the Covariance Matrix Adaptation Evolution strategy. Colours used to represent each optimisation method were based on a palette created using interactive Bayesian optimisation. This palette can be viewed in figure 5.2.

it can be seen that each average final G value is close to 1. CMA-ES has also performed well, especially for the Six-Hump Camel function.

Bayesian optimisation performed particularly well for the 3D Hartmann 3.2d function within the first 30 function evaluations. Results for both the 3D Hartmann and 6D Hartmann 3.2e functions show that for Bayesian optimisation, iteration-wise improvement tails off as the optimisation progresses, despite there still being a non-negligible error. It might be found that CMA-ES and PSO outperform Bayesian optimisation, given a larger function evaluation budget.

The Shekel functions are notoriously difficult to minimise. Brochu (2010) found similarly low values for G using Bayesian optimisation to those in figures 3.2f, 3.2g and 3.2h. CMA-ES seems to perform better than Bayesian optimisation when the number of function evaluations is high, especially for the Shekel function with 10 local minima. However, PSO does not seem to work well for any of the Shekel functions.

CMA-ES and Bayesian optimistion seem to perform similarly well for the 2-dimensional Ackley function in figure 3.2i, while PSO does not appear to be very effective within the function evaluation restriction. Bayesian optimisation looks as if it works the best for the 5-dimensional Ackley function in figure 3.2j for small function evaluation budgets, although the trend of G for CMA-ES suggests that it might be the most effective beyond 100 evaluations.

In figures 3.2k and 3.2l respectively, results for two highly multimodal test functions: Shubert and Rastrigin, are displayed. Within the stopping criterion for the Shubert test, PSO seems to have the best overall average improvement, implying that it might suit optimisation of the Shubert function particularly well. However, within 10 function evaluations, Bayesian optimisation is by far the most effective. The Rastrigin function appears to be minimised fairly well using Bayesian optimisation for a small function evaluation budget. As the optimisation continues, CMA-ES seems to catch up on average.

3.6 Discussion

Overall, it seems fair to conclude that Bayesian optimisation is a powerful method that in general outperforms the others for small function evaluation budgets. Due to the cost of user input, the function evaluation budget should be minimised for the colour or colour combination affinity optimisation problem considered in this project. If an individual must score many colours or combinations thereof, they might lose interest, causing a failure of the optimisation. For this reason, Bayesian optimisation is the best of the 3 methods for interactive optimisation, and therefore, the problem at hand.

That being said, it is clear from the vastly different run times that for optimisation of a cheap objective function, such as the set of test functions used in this chapter, CMA-ES and PSO are far more suitable than Bayesian optimisation. A few CMA-ES and PSO tests were so quick that their run time registered as 0.0 seconds. In contrast, many of the Bayesian optimisation tests were run overnight to comply with time constraints.

It should be noted that these tests were based on a very small number of runs: only five per function per method. This was due to the time constraints, mostly imposed by the long run times for Bayesian optimisation. The results presented may not be fully representative of the performance of each method.

Also, the apparent poor performance of particle swarm optimisation might be due to the unsuitability of the improvement measure, G. In equation 3.1, G is shown to depend on the initial function value. For Bayesian optimisation with one evaluation per iteration, this is unambiguous. However, for an initialisation involving more than one evaluation, the best of them was taken as the initial function value. When 10 or more evaluations are carried out in the first iteration, as with PSO, the initial function value is often close to the minimum already. Therefore, improvement may be more difficult. This effect might also apply to CMA-ES to a lesser extent, with only 2 evaluations per dimension for each iteration.

Chapter 4

Painting Experiment

It has been established that Bayesian optimisation is the best method for minimisation of the majority of test functions in chapter 3. Testing can be expanded to include optimisation of colour preference, using an individual's artwork as data. In the following chapter, Bayesian optimisation will be demonstrated for an artist's colour affinity, assuming that their most-used colour compared to other artists is their preference. However, there could be many reasons other than preference for colour usage within paintings, such as subject matter or an evoked feeling intended by the artist. With this in mind, an artist's colour usage can be considered as a surrogate for their colour affinity.

Colour usage is similar to colour affinity in that it is highly multi-modal with respect to 3-dimensional colour space; an individual usually likes many colours from different parts of the spectrum and not necessarily those in-between, and an artist uses a colour palette containing a selection of colours. This similarity allows Bayesian optimisation to be tested for a problem resembling the colour preference problem, without the need for a potentially time-consuming user study for interactive colour preference learning.

In practice, this problem would be not necessarily be a good candidate for a Bayesian optimisation approach. Scores for each colour are based on the combined histogram of their paintings compared to the combined histogram of the remaining dataset of paintings, so it would make sense to calculate the full histograms prior to the optimisation, essentially making it easy to evaluate the whole objective function prior to the optimisation process. This makes redundant the efficiency of Bayesian optimisation with respect to function evaluations.

However, in section 4.2, the problem will be explored in more detail. There may be a justification for the use of Bayesian optimisation when a kernel density estimate is used to negate the inaccuracy of the histogram binning process. Regardless of the suitability of Bayesian optimisation to the problem, this experiment is a good demonstration of the algorithm when applied to colour affinity.

4.1 Data Set

The paintings used in this experiment are taken from the Modern and Contemporary Art department within the collection of The Metropolitan Museum of Art. The selection criteria for paintings is as follows: the image is available online; the artwork is in the public domain; the image is not monochromatic; the image does not include a large border or part of the artwork's frame; the image is within the first five images of the search results suitable, given the previous criteria, for a single artist.

The acquired set of artwork consists of 95 paintings. A set of up to five paintings by a specific artist can be taken as the subject of the experiment. For the purposes of demonstration, the art of Florine Stettheimer has been chosen, pictured in figure 4.1. Florine Stettheimer (1871-1944) was a New York socialite and fairly unknown painter. Her style was "characterised by a vivid palette and stylized figures." (Benezit Dictionary of Artists)

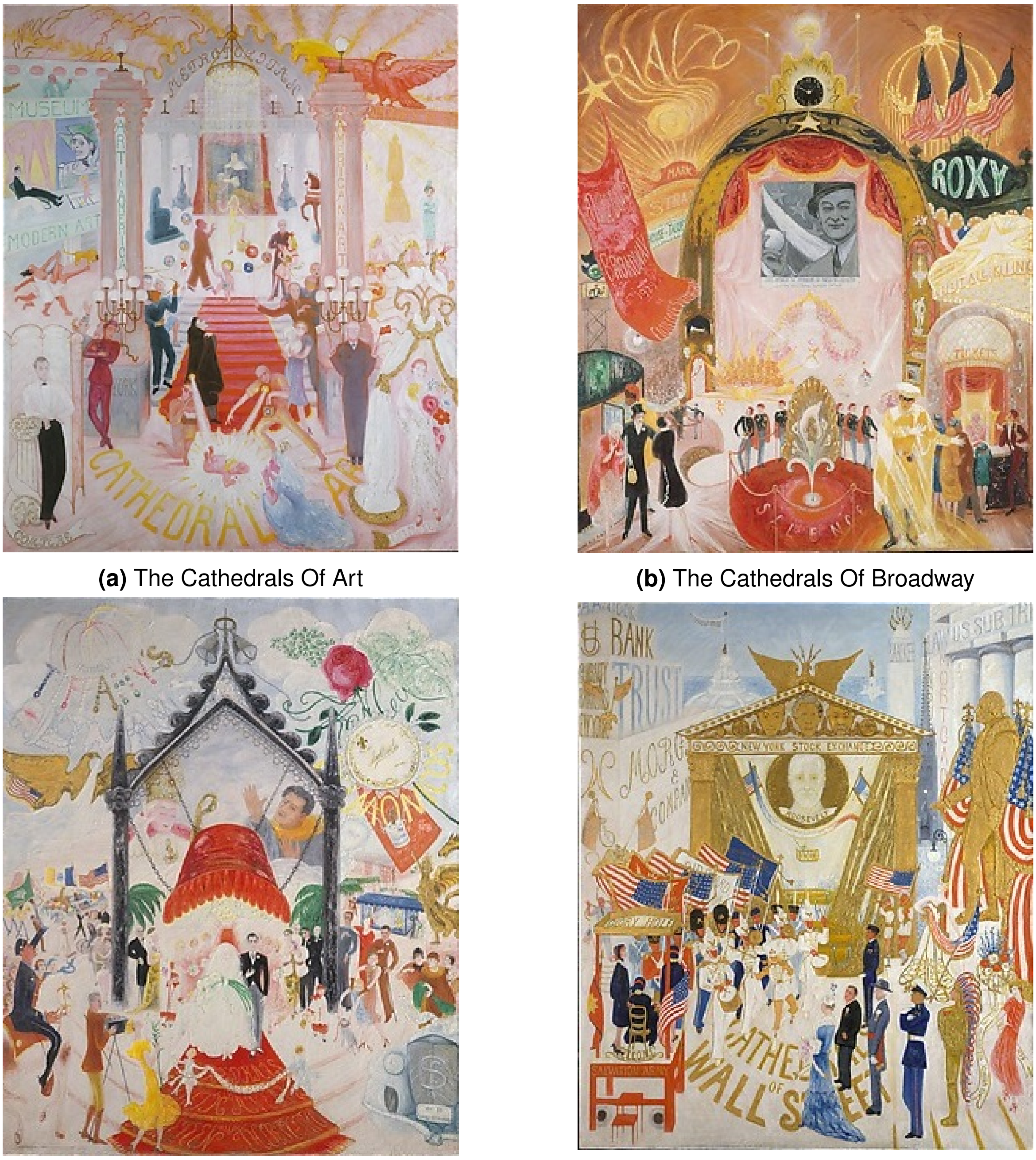
4.2 Background

An image can be represented as a 2-dimensional array of pixels, each of which can be represented by a 3-dimensional RGB vector. Only the colour of each pixel, and not its location in space, is relevant here. Therefore, it is simpler and just as informative to represent each image as a list of RGB vectors, or pixel list. A pixel list can be produced for each painting. Under the reasonable approximation that the paintings are the same size, pixel lists can be concatenated to represent the body of work by the selected artist and remaining artists.

A histogram operation collects the number of pixels in a each colour range, specified by bins. For example, for a colour space using 8 bins per channel, the space would be discretised by 512 single-coloured cubes. Normalised histograms can be calculated using both pixel lists, by dividing the resulting pixel counts by the number of pixels in each list.

This experiment relies on the idea that an artist's colour affinity can be defined as their use of a colour compared to the average use for other artists. Therefore, their colour preference can be found by maximisation of the chosen artist's histogram minus the other artists' histogram. The resulting objective function is very sharp and multimodal.

In figure 4.2, the normalised red channel histogram difference for Stettheimer's work with 256 bins is plotted. Note the small deviations from what seems to be a multimodal Gaussian. This could be explained by the discrete nature of paints- an artist often pre-mixes a colour palette and uses those colours in a painting. Colours slightly different to those pre-mixed colours are likely to be liked a similar amount, but they aren't in the painting and so aren't categorised as such. For this reason,



(a) The Cathedrals Of Art

(b) The Cathedrals Of Broadway

(c) The Cathedrals Of Fifth Avenue

(d) The Cathedrals Of Wall Street

Figure 4.1. Works by Florine Stettheimer (The Metropolitan Museum of Art)

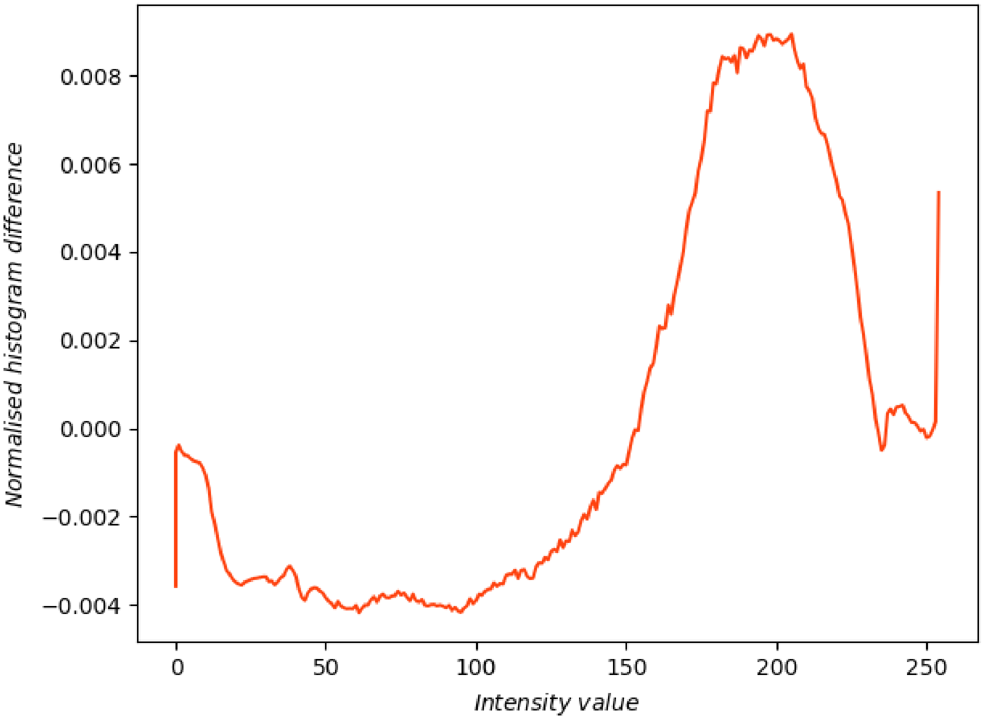


Figure 4.2. The difference between normalised red channel histograms of Florine Stettheimer's paintings and the remaining 91 paintings in the data set.

simply finding the maximum bin count when bins are very small might not yield the desired result of the optimisation.

4.2.1 The Kernel Density Estimate

It has been expressed before that the function of colour affinity over the 3-dimensional colour space is likely to be smooth. A small change in red, blue or green intensity value of a colour is not likely to adversely affect how much a person likes the colour, due to the change being barely detectable. The application of this idea to the entire colour space provides a justification for the assumption of smoothness.

If a person's colour affinity is smooth, smoothing might improve the suitability of the histogram difference function to its modelling. At the least, it would make the histogram difference easier to maximise using a continuous optimisation method. A smooth histogram can be produced through use of a Gaussian kernel density estimate (Duong, 2001). A kernel density estimate is a generalisation of a histogram where the need for binning is eliminated. In the case of colours, instead of each pixel being counted as an instance of the single-coloured cubic bin to which is belongs, it is counted as its exact RGB value. In one-dimensional input space, this is equivalent to stacking tophats centred at their exact values (Duong, 2001), resulting in a histogram like in figure 4.3.

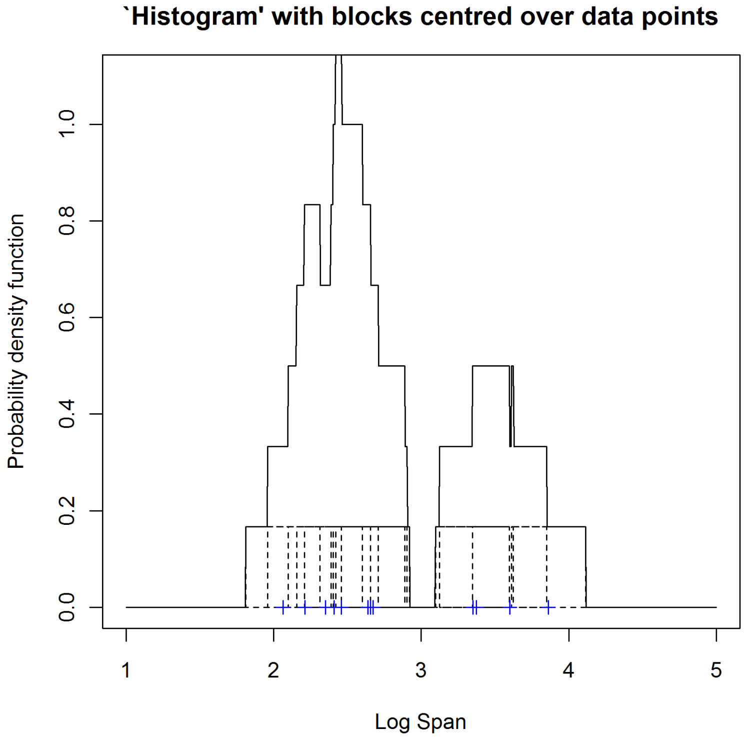


Figure 4.3. A Histogram estimate using addition of tophats centred on true values, from Duong (2001).

This method has the advantage of being unbiased by any choice of bin width. However, the function is not smooth. Instead of using a top hat to represent each data point, a Gaussian could be used. These blocks are the kernels referred to in the technique's name: kernel density estimation. A Gaussian kernel density estimate has been applied to the smoothed red channel difference histogram for Stettheimer's paintings in figure 4.4.

4.3 Methodology

First, Bayesian optimisation is applied to the smoothed red channel difference histogram, of which Florine Stettheimer's work is the subject.

Next, Bayesian optimisation is applied to the full colour difference histograms, with several artists' work chosen as the subjects, including Stettheimer's. Each unique histogram will be maximised 5 times, with a stopping criterion of 120 function evaluations.The initial colour for each trial is chosen randomly.

The best colour and evaluated colour are recorded at each iteration for each case. These results can be compared to the maximum of each small-binned conventional difference histogram. However, this shouldn't necessarily be considered to be a true preference, as explained in section 4.2. The smooth histogram difference functions will be considered to be unknown. For this reason, no

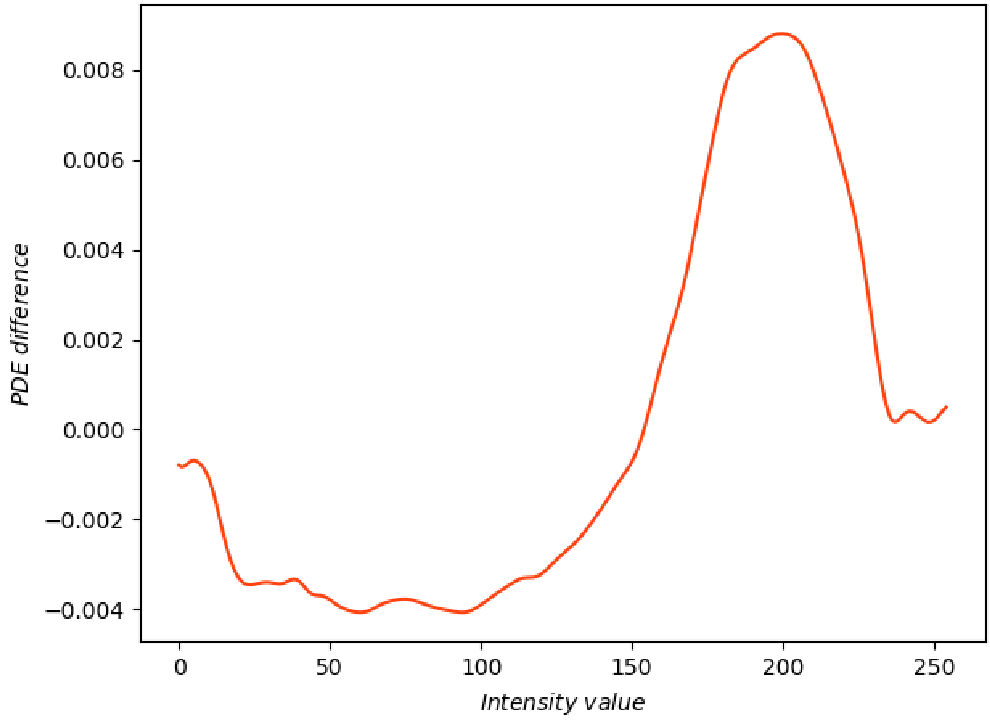


Figure 4.4. The difference between Gaussian kernel density estimates of the red channel histograms of Florine Stettheimer's paintings and the remaining 91 paintings in the data set.

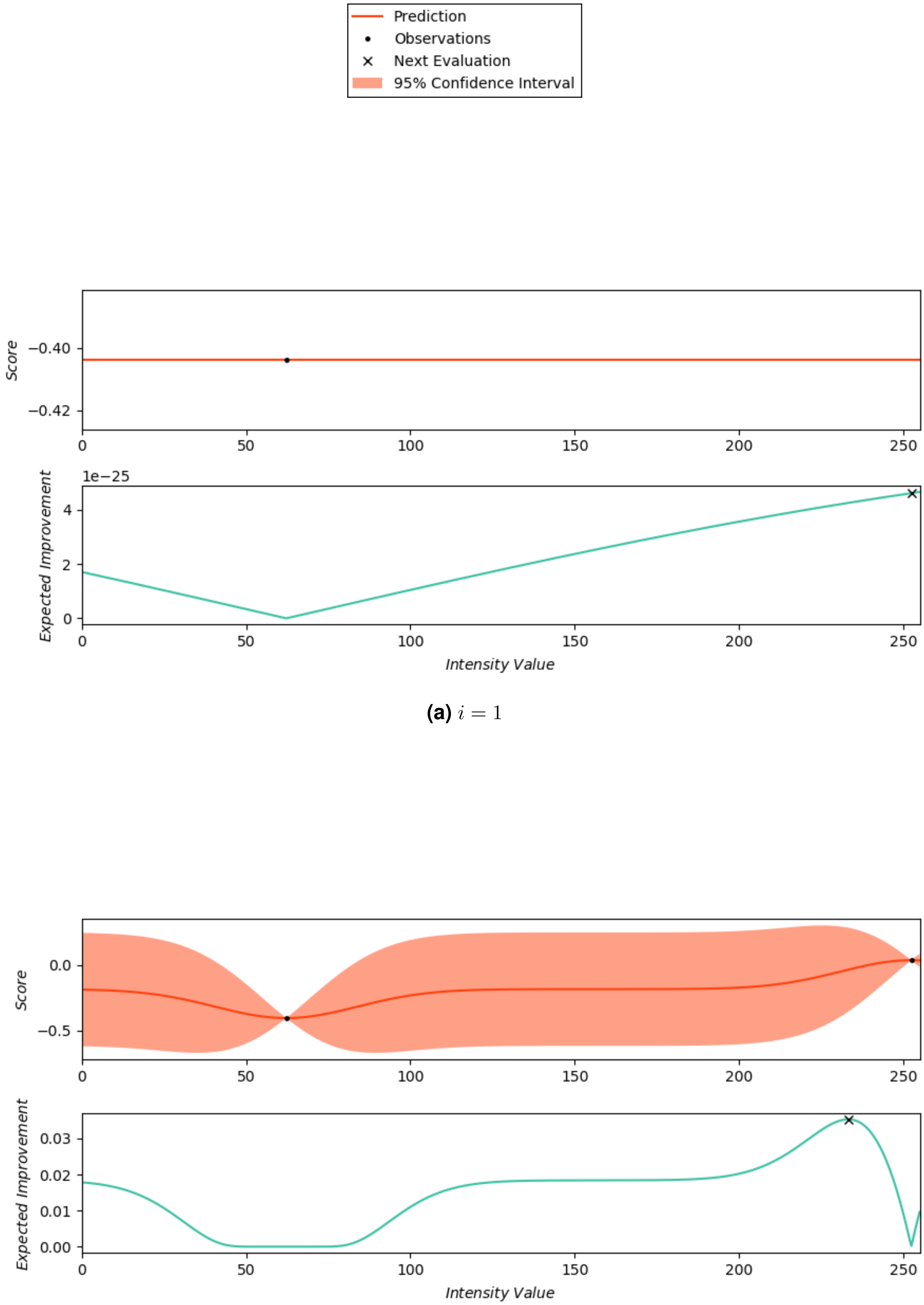
measures of performance have been applied.

4.4 Red Channel Results

To allow for visual representation of the Bayesian optimisation process, only the intensity values of the red channel will be considered at first. The smoothed histogram difference, to which Bayesian optimisation has been applied, is shown in figure 4.4. figure 4.5.

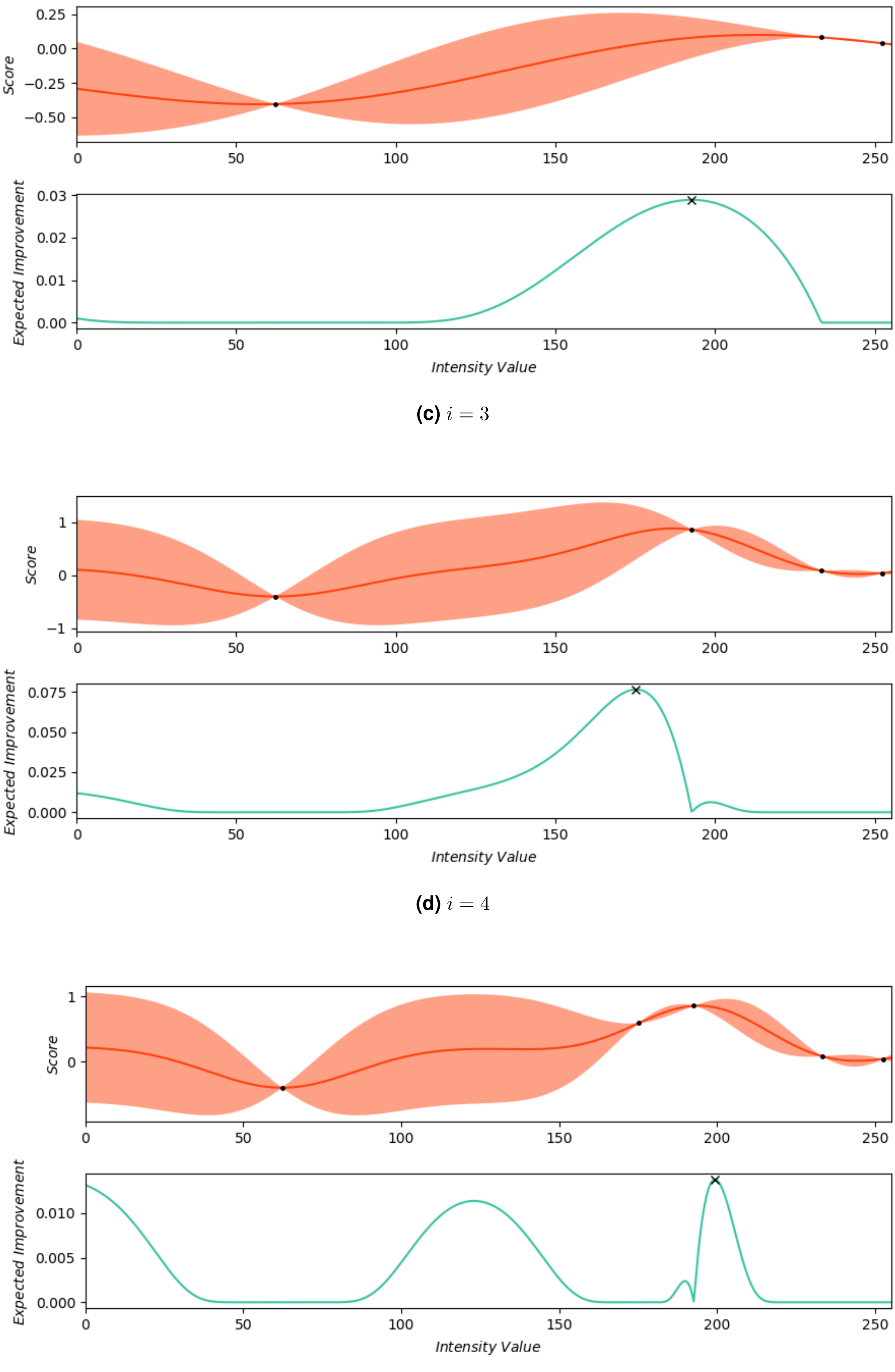
In figure 4.5h, the maximum of the expected improvement is not found perfectly. This can be attributed to the very small magnitude of its gradient. However, this evaluation is still better than most according to the expected improvement, so perhaps this shortcoming in its maximisation can be overlooked.

After 6 function evaluations, the main peak in the smoothed histogram difference (4.4) has been discovered. The true maximum is at an intensity value of about 205, which is very close to the maximum of this peak in the Gaussian process, at just over 200.


Prediction
Observations
Next Evaluation
95% Confidence Interval
Score
−0.40
−0.42
1e−25
Expected Improvement
4
2
0
0
50
100
150
200
250
Intensity Value
Score
0.0
−0.5
Expected Improvement
0.03
0.02
0.01
0.00
0
50
100
150
200
250
Intensity Value

(a) $i = 1$

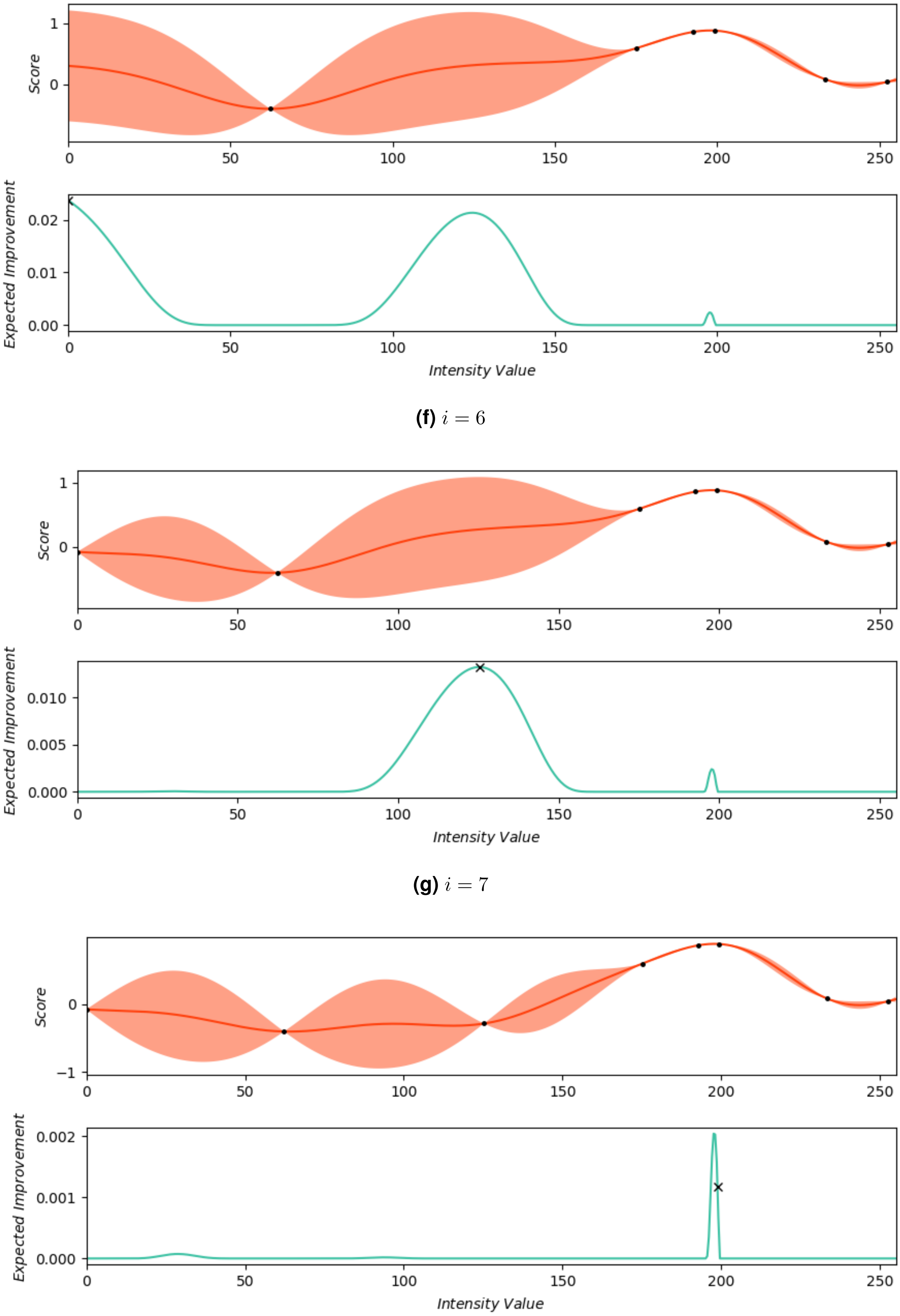
(b) $i = 2$


Score
0.25
0.00
−0.25
−0.50
0
50
100
150
200
250
Expected Improvement
0.03
0.02
0.01
0.00
Intensity Value
Score
1
0
−1
Expected Improvement
0.075
0.050
0.025
0.000
Intensity Value
Score
1
0
Expected Improvement
0.010
0.005
0.000
Intensity Value

(c) $i = 3$

(d) $i = 4$

(e) $i = 5$



(f) $i = 6$

(g) $i = 7$

(h) $i = 8$

Figure 4.5. Plots showing the Gaussian process and expected improvement at each stage of the Bayesian optimisation process. Colours used to represent the predictive distribution and expected improvement were taken from a palette created using interactive Bayesian optimisation (figure 5.3).

4.5 Full Colour Spectrum Results

Given the success of Bayesian optimisation for intensity of a single channel, it can now be applied to the full colour spectrum. In section 4.2, it was recognised that the maxmimum argument of the conventional histogram difference could be found easily. Those colours for several different artists are displayed in table 4.1, next to some very small-scale images of their artwork samples. These colours can be compared to the results for Bayesian optimisation of the smoothed histogram difference, which are displayed in table 4.2. The final colours for each run of the optimisation can be identified in within the painting samples for each artist. Some of these colours can be thought of as local maxima, therefore.

The full run history of the optimisation with Florine Stettheimer's work as the subject is presented in figure 4.6. Clearly presented are the periods during which exploration and exploitation occur. The highest scoring colours after 120 iterations can all be matched to colours within Stettheimer's paintings in figure 4.1, which is encouraging. It is likely that most of these are local maxima. However, the final results of runs 4 and 5 are close to the highest scoring colour in table 4.1, perhaps suggesting proximity to the global minimum.

4.6 Discussion

This chapter has explored the use of a painting histogram difference to model an artist's colour affinity, and its maximisation using Bayesian optimisation to infer colour preference. For a large number of function evaluations, the discovered colour preference was often close to the maximum of the true histogram difference, and almost always recognisable within the painting sample.

However, the abundance of a particular colour in comparison to its use in general might not be very reflective of a painter's true colour preference. For at least two of the cases: Stettheimer and Dove, the discovered colour preference was used predominantly or wholly in the background. Therefore, the choice to use a large amount of those particular colours may have been decided by colour combination preference, the feeling they were trying to convey in the piece, or a real object they were trying to represent.

The application of Bayesian optimisation was, in some cases, quite successful. However, in some other cases, improvement was hard to come by. Over much of the colour space, the smoothed histogram difference function was of zero or very small magnitude . Low-magnitude-scoring initial and subsequent colours produced a very flat predictive distribution, which in turn meant that the expected improvement was small for the whole colour space, and quite flat far from the observations. This

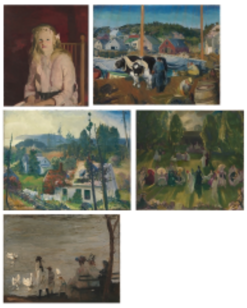
Artist's Name	Sample of paintings	Highest scoring colour
Florine Stettheimer		⬜
Arthur Dove		🟨
George Bellows		🟫
Horace Pippin		⬛

Table 4.1. A table showing a sample of paintings for each artist, and the highest scoring colour in the difference between normalised colour histograms of the painting sample and the remaining paintings in the data set.

Artist's name	Run no.	Highest scoring colour				
		$i = 1$	$i = 31$	$i = 61$	$i = 91$	$i = 121$
Florine Stettheimer	1	🟥	🟨	🟨	🟫	🟫
	2	🟦	🟪	🟪	🟪	🟪
	3	🟪	🟪	🟪	🟧	🟫
	4	🟪	🟪	🟪	⬜	⬜
	5	🟦	🟨	⬜	⬜	⬜
Arthur Dove	1	🟧	🟩	⬛	⬛	⬛
	2	🟪	🟩	🟩	🟩	🟩
	3	🟩	🟩	🟩	⬜	⬜
	4	🟥	🟩	🟩	⬜	⬜
	5	🟩	🟩	⬜	⬜	⬜
George Bellows	1	🟥	🟩	🟩	🟩	🟩
	2	🟩	🟦	🟦	🟦	🟦
	3	🟨	🟫	🟫	🟫	🟫
	4	🟩	🟩	🟩	🟦	🟦
	5	🟦	🟪	🟪	🟪	⬛
Horace Pippin	1	🟫	⬜	⬜	⬜	⬜
	2	🟥	🟪	🟫	🟫	🟫
	3	🟩	🟩	⬛	⬛	⬛
	4	🟥	🟩	⬜	⬜	⬜
	5	🟦	🟧	⬛	⬛	⬛

Table 4.2. The best approximation of the maximum of the colour affinity model at different iterations of the Bayesian optimisation process. The smoothed histogram difference, for a sample of paintings of several artists, was the objective function.

either resulted in a random-esque search or attempted repetition of exploitation of a very low-scoring maximum. This was the probably down to inaccuracies in maximisation of the expected improvement as a result of very small gradients.

If this experiment was repeated for a practical reason other than for demonstration, initialisations wouldn't be randomised. Best practice in this situation would be to initialise the optimisation based on prior beliefs. In this situation, looking at the set of paintings and picking out one of the most obvious colours would be a good strategy.

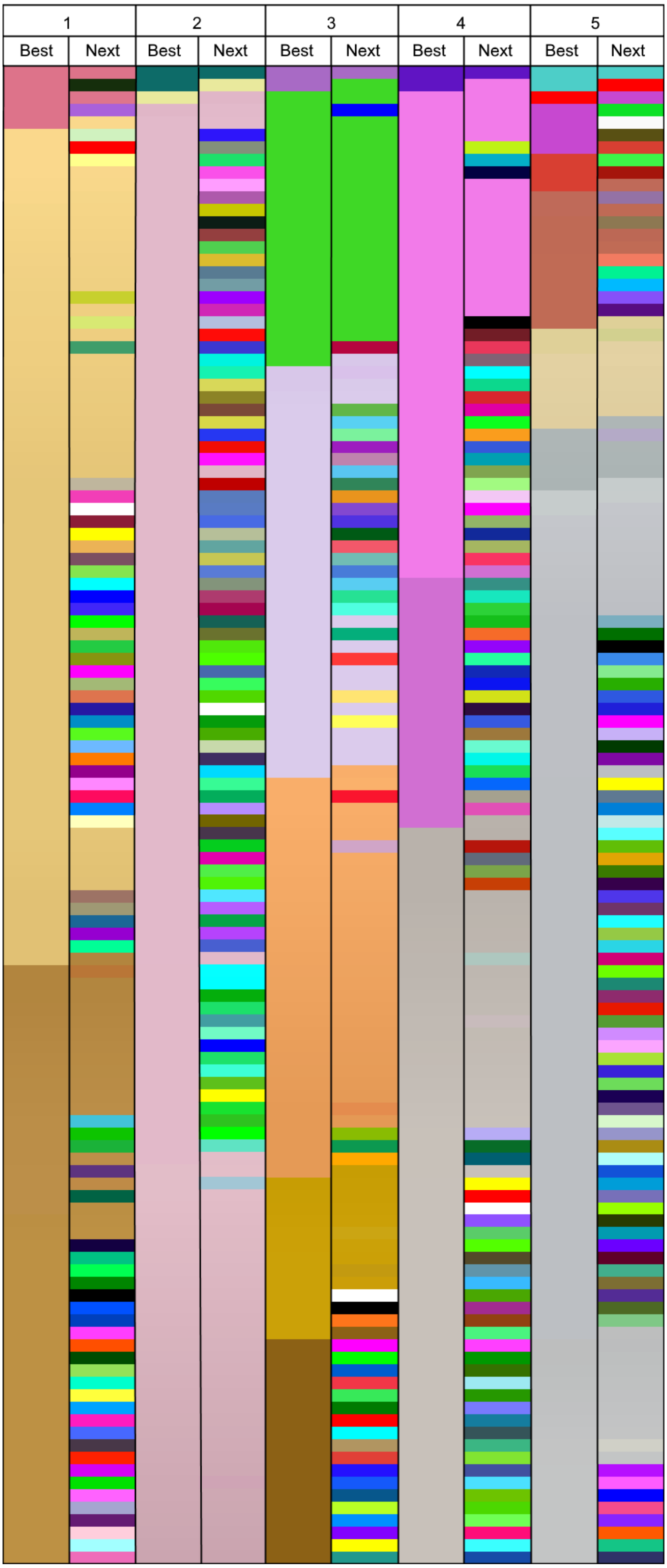


Figure 4.6. A visual representation of 5 runs of the colour histogram difference optimisation for Stettheimer's work. For each run, the next colour to be evaluated and the current best colour are shown at each iteration. There are 121 rows of colour in total, representing 120 iterations and the initialisation.

Chapter 5

Interactive Colour Preference Learning

In chapters 3 and 4, it was established that Bayesian optimisation is an appropriate method to find colour or colour combinations favoured, if not preferred, by an individual based on their input. The following chapter is concerned with the implementation of this interactive colour affinity optimisation, and its application to colour palette creation.

5.1 Bayesian Optimisation of Colour Preference

First, the user is presented with a randomly generated colour or colour combination. They are asked to score the colour or combination thereof on a numerical scale. Taking inspiration from Solnik et al. (2017), this scale extends continuously from 0, the worst colour or combination, to 7, the best, in hopes that limiting the choice of score will make one easier to assign. The first observation is recorded, and the optimisation begins: a Gaussian process is fitted to the observation; the maximiser of the expected improvement is obtained. This maximiser is a colour or colour combination which requires scoring as before, and these steps are repeated. An example of this process for a single colour is shown in table 5.1.

i	Colour to Evaluate	Score	Current Best Colour
1	🟨	2	🟨
2	🟪	3	🟪
3	🟦	3	🟪
⋮	⋮	⋮	⋮
28	🟩	3.5	🟥
29	🟩	3	🟥
30	🟥	6	🟥

Table 5.1. A table showing the next colour to be scored and the current best scoring colour at each iteration of Bayesian optimisation. I was the user providing colour affinity scores.

The stopping criterion for interactive colour preference learning is best defined by the number of iterations a user can bear to provide, because the objective function is unknown. From experience, about 30 individual colours can be evaluated before the process becomes tedious. For colour pairs, this may be doubled to 60, given that the final result yields double the information. However, due to the additional data gained at each iteration, successive computations of evaluation points take longer. As a result, an individual intending to find an optimal colour pair may need more patience.

The scale of possible scores is defined between the integers 0 and 7. This use of integers can guide the user to pick integer scores. This potential dicretisation of scores might cause some inaccuracy, introducing noise to the observations. To aid understanding of this concept, discrete scores for an arbitrary input are shown in figure 5.1, to which a Gaussian process is fitted. The area around its peak is problematic, as noise created by rigidity of the scoring interferes with predictions.

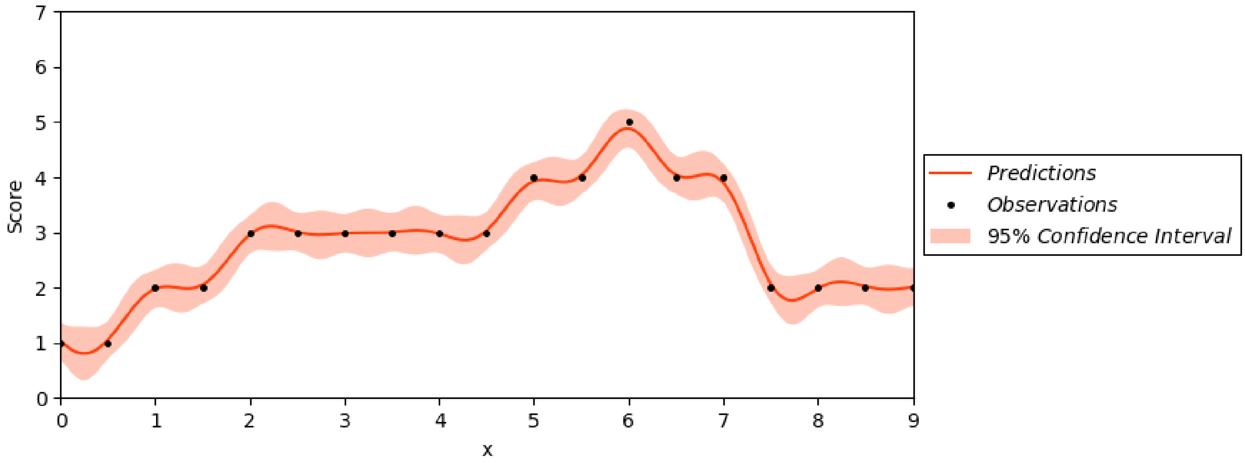


Figure 5.1. A plot showing observations which are restricted to discrete values from 0 to 7, and the Gaussian process fitted to them.

However, the scale is not restricted to integers; it is continuous. This allows for instances like the 28th colour score of 3.5 in table 5.1, where this certain shade of green is judged to be nicer than a colour already assigned a score of 3, but worse than one with a score of 4. This flexibility might partially reduce the effect of noise arising from inaccuracies in scoring.

5.2 Optimal Colour Palette Selection

The colour palette selection method proposed relies on the idea that a colour can be scored given that it is combined with a set of other colours. The method in section 5.1 can be used to find a single optimal colour and an individual, and then iteratively, the method can be re-applied, with the optimal colours found so far given in combination with the colour to be scored. The result is a colour palette containing as many colours as the user desires.

This colour palette extension method is a cheap approximation of the difficult task of optimising affinity for a combination of colours. The procedure can be described as greedy, because the colour affinity optimisation problem is solved for each colour individually.

In table 5.2, the optimal colour found in table 5.1 is extended to a palette of two colours using Bayesian optimisation, as mentioned previously.

i	Colour Combination to Evaluate	Score	Current Best Colour Combination
1	🟥🟦	3	🟥🟦
2	🟥🟧	4	🟥🟧
3	🟥🟫	1	🟥🟧
⋮	⋮	⋮	⋮
28	🟥🟫	2	🟥🟩
29	🟥🟩	5	🟥🟩
30	🟥🟩	5	🟥🟩

Table 5.2. A table showing the next colour pair to be scored and the current best scoring colour pair at each iteration of Bayesian optimisation. I was the user providing colour affinity scores.

The two colour palette can be extended again to a three colour palette, using the same method. The resulting palette is shown in figure 5.2. This palette has been used to distinguish between performance measures for three different optimisation methods in figure 3.2.

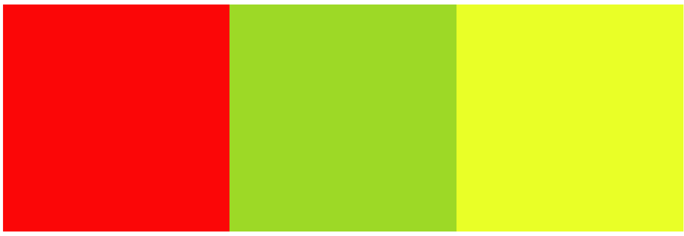


Figure 5.2. A palette of three colours obtained through interactive Bayesian optimisation, where results of this process are shown in tables 5.1 and 5.2. These colours were optimised in line with my own preferences.

The colour palette extension method can be applied to a colour that is already chosen. An example of a 5-colour palette, where the first colour was pre-defined as orange red (xkcd) and the subsequent colours were optimised, is shown in figure 5.3. This palette is used in figure 2.2, where the different colours reflect variation of Gaussian process hyperparameters.

The colour palette extension technique detailed in this section also could be applied to colour pairs, or even larger combinations, where they replace the single colours in the process shown in tables 5.1 and 5.2. As indicated in section 5.1, this might be significantly more time-consuming.

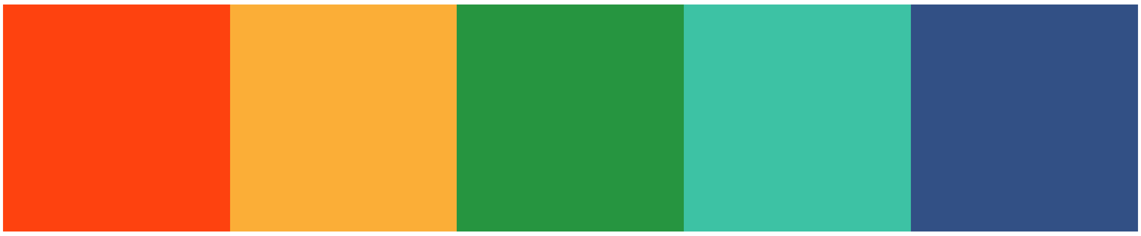


Figure 5.3. A palette of five colours, the first chosen to be orange red (xkcd), and the rest chosen using interactive Bayesian optimisation. These colours were optimised in line with my own preferences.

5.3 Discussion

The process for obtaining an optimal colour or colour combination using interactive Bayesian optimisation was presented in section 5.1. A colour palette expansion technique employing this colour affinity optimisation method was put forward. Using this technique, two colour palettes were created and applied to figures in previous chapters, for the purposes of demonstration.

For optimisation of colour preference in section 5.1, the choice of scale on which to score colours or combinations thereof was considered carefully, resulting in the selection of a continuous scale from 0 to 7. However, with experience of scoring in this way, it is difficult to maintain consistency over a session. A solution to this issue is proposed by Brochu (2010): order of preference can be inferred by choosing preferences from sets containing two choices each. Results obtained in this user study suggest that this technique is particularly effective.

Finding an optimal colour, given that it is to be combined with a set of pre-determined colours, could be particularly useful in practice. In instances such as web design, a company logo might already exist, which is to be incorporated into the background. A background colour which compliments said logo could easily be found using the colour palette extension technique in section 5.2. The creation of a palette by finding one colour at a time involves a series of three-dimensional global optimisation problems, instead of one $3c$-dimensional problem. This greedy process is more efficient due to shorter computation times at each iteration of the Bayesian optimisation.

As a user of the colour palette creation software, I prefer the palettes in figures 5.2 and 5.3 for use in graphs in previous chapters to the colours I had previously chosen. I also appreciate that the five-colour palette in figure 5.3 included bright yet muted colours that I wouldn't have thought to use otherwise.

Chapter 6

Conclusions

The intention behind this work was to determine if a person's ultimate colour or colour combination preference is best found using Bayesian optimisation. Several questions needed answering: Can colour affinity be modelled as a function of colour arrtibutes?; How is Bayesian optimisation implemented, and what makes it a candidate for this application?; How well does Bayesian optimisation perform compared to other suitable methods?; Can colour affinity be optimised efficiently using Bayesian optimisation?; Can a user's colour affinity be optimised efficiently using interactive Bayesian optimisation? The answers to these quesions, which were addressed in previous chapters, are to be discussed further.

In chapter 1, colour or colour combination preference was intoduced as an optimisation problem. Previous attempts to model colour affinity were explored and reviewed, and it seemed to be the consensus that colour affinity has a relationship to colour attributes.

The theory behind Bayesian optimisation was discussed in detail in chapter 2. The idea that intelligently chosen function evalutions give rise to good performance under a tight evaluation budget was emphasised. This idea is particularly applicable to user-defined objective functions such as colour affinity, when evaluated interactively.

In chapter 3, the performance of Bayesian optimisation was assessed alongside Particle Swarm optimisation and the Covariance Matrix Adaptation Evolution strategy using a set of test functions. Overall, Bayesian optimisation performed the best for a fixed number of function evaluations, but performed much worse than the other two methods with respect to time. This suggests that, for the colour affinity optimisation problem, Bayesian optimisation is the best choice. Concern lies mostly with the number of evaluations needed when objective function cost is high. To enhance confidence in the Bayesian optimisation algorithm for colour affinity, these tests could be repeated where some noise is added to the test functions.

The painting experiment within chapter 4 was intended for demonstration purposes. It was shown that Bayesian optimisation could be applied to an objective function with a 3-dimensional continuous search space representing colour. The results support that some function of colour can be optimised using Bayesian optimisation, though not always efficiently. Also, the histogram difference may not be a good model for colour affinity.

In chapter 5, a method for interactive Bayesian optimisation of colour affinity was put forward. A technique employing this method to create and extend colour palettes was proposed and demonstrated. The resulting palettes were used in figures throughout this report. The user's colour affinity was evaluated by requesting a score on a numerical scale. This method could be improved upon by use of preference galleries (Brochu, 2010), because remaining consistent whilst scoring colours on a numerical scale is difficult for humans. Also, a user study to test the Bayesian optimisation algorithm in an interactive context could provide evidence to support an answer to the question proposed in this project: is Bayesian optimisation the best method for finding a user's colour or colour combination preference?

The final conclusion is that, although Bayesian optimisation is not efficient enough to find a person's ultimate colour or colour combination preference through interaction, it can produce colours that are subjectively nice.

Bibliography

Benezit Dictionary of Artists. Stettheimer, Florine. URL http://www.oxfordartonline.com/view/10.1093/benz/9780199773787.001.0001/acref-9780199773787-e-00175693.

Eric Brochu. *Interactive Bayesian Optimization: Learning Parameters for Graphics and Animation*. PhD thesis, University of British Columbia, Vancouver, Canada, December 2010.

Tarn Duong. An introduction to kernel density estimation. *Weatherburn Lecture Series, Department of Mathematics and Statistics, University of Western Australia*, 24, 2001. URL http://www.mvstat.net/tduong/research/seminars/seminar-2001-05/.

David Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.

H. J. Eysenck. A Critical and Experimental Study of Colour Preferences. *The American Journal of Psychology*, 54(3):385–394, 1941. doi: 10.2307/1417683.

R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, 1987.

G.W. Granger. An Experimental Study of Colour Preferences. *The Journal of General Psychology*, 52(1):3–20, 1955. doi: 10.1080/00221309.1955.9918340.

Ihab Mahmoud Hanafy and Reham Sanad. Colour Preferences According to Educational Background. *Procedia - Social and Behavioral Sciences*, 205:437 – 444, 2015. ISSN 1877-0428. doi: https://doi.org/10.1016/j.sbspro.2015.09.034. URL http://www.sciencedirect.com/science/article/pii/S1877042815050521. 6th World Conference on Psychology, Counseling and Guidance (WCPCG-2015).

Nikolaus Hansen. *The CMA Evolution Strategy: A Comparing Review*, pages 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-32494-2. doi: 10.1007/3-540-32494-1_4. URL https://doi.org/10.1007/3-540-32494-1_4.

Nikolaus Hansen. The CMA Evolution Strategy: A Tutorial. *CoRR*, abs/1604.00772, 2016. URL http://arxiv.org/abs/1604.00772.

J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995. doi: 10.1109/ICNN.1995.488968.

N. Kita and K. Miyata. Aesthetic Rating and Color Suggestion for Color Palettes. *Computer Graphics Forum*, 35(7):127–136, 2016. doi: 10.1111/cgf.13010.

Daniel James Lizotte. *Practical Bayesian Optimization*. PhD thesis, Edmonton, Alta., Canada, 2008. AAINR46365.

Federico Marini and Beata Walczak. Particle Swarm Optimization (PSO). A Tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149:153 – 165, 2015. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2015.08.020. URL http://www.sciencedirect.com/science/article/pii/S0169743915002117.

M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2): 239–245, 1979. ISSN 00401706. URL http://www.jstor.org/stable/1268522.

J. Močkus. On Bayesian Methods for Seeking the Extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg. ISBN 978-3-540-37497-8.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2000. ISBN 9780387987934. URL https://books.google.co.uk/books?id=epc5fX0lqRIC.

Michael Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, PhD thesis, University of Oxford, 2010.

Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15, 2009.

Li-Chen Ou, M. Ronnier Luo, Andrée Woodcock, and Angela Wright. A study of colour emotion and colour preference. part iii: Colour preference modeling. *Color Research and Application*, 29(5): 381–389, 10 2004. doi: 10.1002/col.20047.

Li-Chen Ou, M. Ronnier Luo, Pe-Li Sun, Neng-Chung Hu, and Hung-Shing Chen. Age Effects on Colour Emotion, Preference, and Harmony. *Color Research and Application*, 37(2):92–105, 4 2012. ISSN 1520-6378. doi: 10.1002/col.20672. URL `http:https://doi.org/10.1002/col.20672`.

Stephen E. Palmer and Karen B. Schloss. An Ecological Valence Theory of Human Color Preference. *Proceedings of the National Academy of Sciences*, 107(19):8877–8882, 2010. ISSN 0027-8424. doi: 10.1073/pnas.0906172107. URL `http://www.pnas.org/content/107/19/8877`.

Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. High Dimensional Bayesian Optimization with Elastic Gaussian Process. In *Proceedings of Machine Learning Research*, volume 70, pages 2883–2891, 2017.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Stephen J. Roberts, Michael A. Osborne, Mark Ebden, Steve Reece, Neale P. Gibson, and Suzanne Aigrain. Gaussian Processes for Time-series Modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013. doi: 10.1098/rsta.2011.0550.

R.M. Schloss, K.B.and Poggesi and S.E. Palmer. Effects of University Affiliation and "School Spirit" on Color Preferences: Berkeley Versus Stanford. *Psychonomic bulletin and review*, 18(3):498–504, 2011.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Technical report, Universities of Harvard, Oxford, Toronto, and Google DeepMind, 2015. URL `http://www.cs.ox.ac.uk/people/nando.defreitas/publications/BayesOptLoop.pdf`.

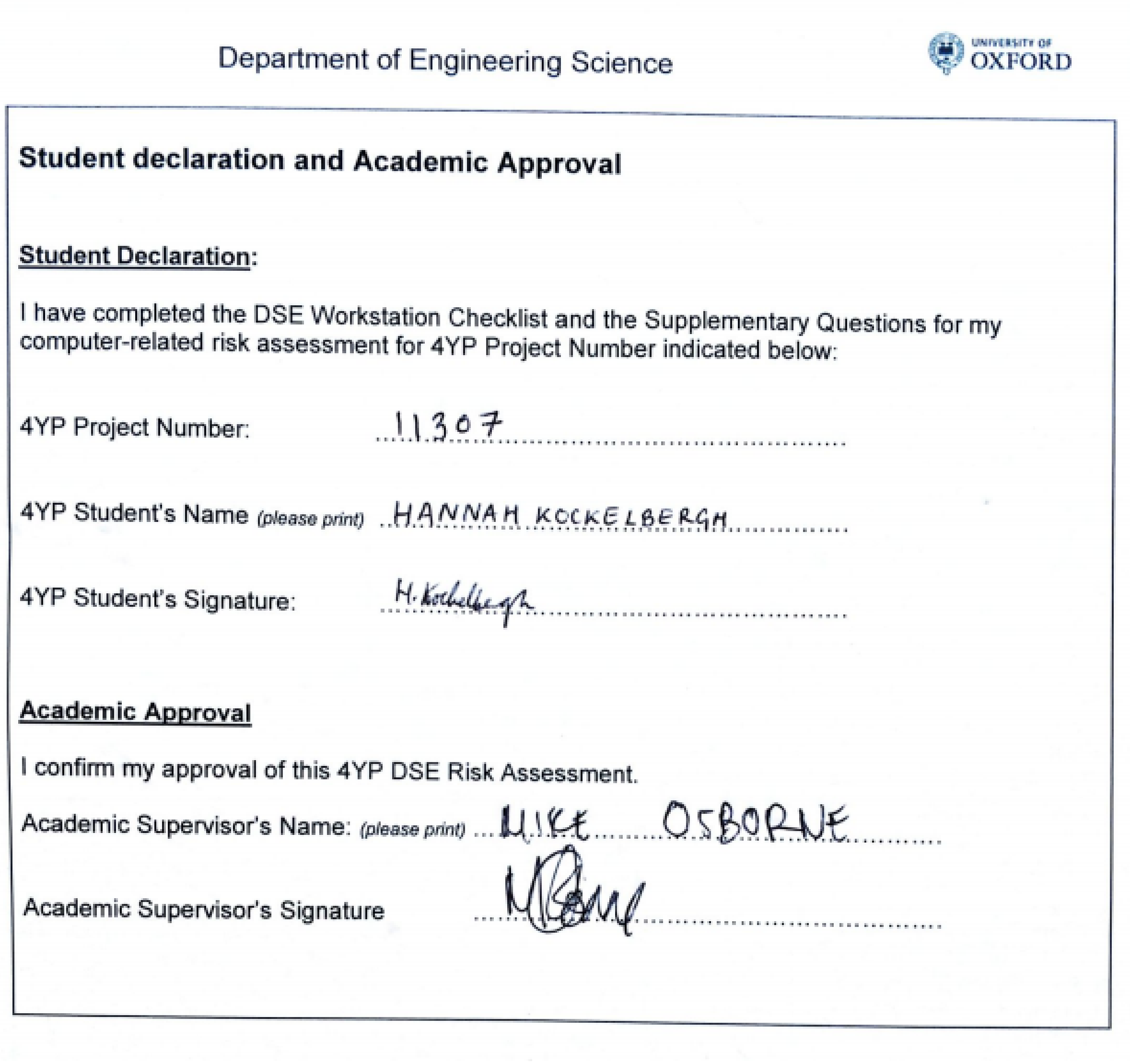
Benjamin Solnik, Daniel Golovin, Greg Kochanski, John Elliot Karro, Subhodeep Moitra, and D. Sculley. Bayesian Optimization for a Better Dessert. In *Proceedings of the 2017 NIPS Workshop on Bayesian Optimization*, December 9, 2017, Long Beach, USA, 2017.

S. Surjanovic and D. Bingham. Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved April 20, 2018, from `http://www.sfu.ca/~ssurjano`.

The Metropolitan Museum of Art. Collection. URL `https://www.metmuseum.org/art/collection`. [Online; accessed March 23, 2018].

xkcd. Common RGB Colours. URL `https://xkcd.com/color/rgb/`.

Risk Assessment



Department of Engineering Science

UNIVERSITY OF OXFORD

Student declaration and Academic Approval

Student Declaration:

I have completed the DSE Workstation Checklist and the Supplementary Questions for my computer-related risk assessment for 4YP Project Number indicated below:

4YP Project Number: 11307

4YP Student's Name *(please print)* HANNAH KOCKELBERGH

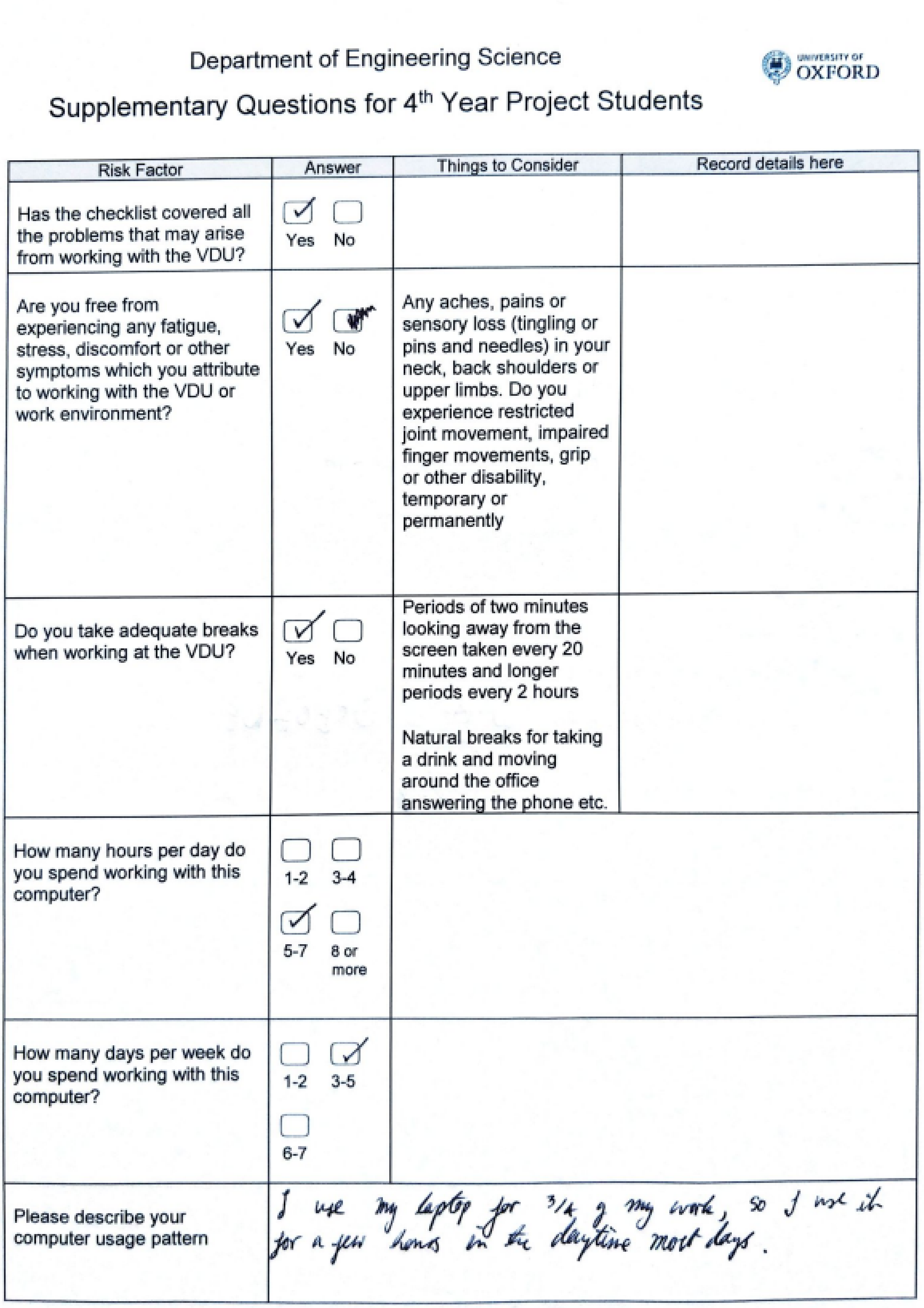
4YP Student's Signature: H. Kockelbergh

Academic Approval

I confirm my approval of this 4YP DSE Risk Assessment.

Academic Supervisor's Name: *(please print)* MIKE OSBORNE

Academic Supervisor's Signature



Department of Engineering Science

UNIVERSITY OF OXFORD

Supplementary Questions for 4th Year Project Students

Risk Factor	Answer	Things to Consider	Record details here
Has the checklist covered all the problems that may arise from working with the VDU?	☑ Yes □ No		
Are you free from experiencing any fatigue, stress, discomfort or other symptoms which you attribute to working with the VDU or work environment?	☑ Yes ☒ No	Any aches, pains or sensory loss (tingling or pins and needles) in your neck, back shoulders or upper limbs. Do you experience restricted joint movement, impaired finger movements, grip or other disability, temporary or permanently	
Do you take adequate breaks when working at the VDU?	☑ Yes □ No	Periods of two minutes looking away from the screen taken every 20 minutes and longer periods every 2 hours Natural breaks for taking a drink and moving around the office answering the phone etc.	
How many hours per day do you spend working with this computer?	□ 1-2 □ 3-4 ☑ 5-7 □ 8 or more		
How many days per week do you spend working with this computer?	□ 1-2 ☑ 3-5 □ 6-7		
Please describe your computer usage pattern	I use my laptop for 3/4 of my work, so I use it for a few hours in the daytime most days.		