People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

MOHAMED KHIDER UNIVERSITY, BISKRA

FACULTY of EXACT SCIENCES, SIENCE of NATURE and LIFE

DEPARTMENT of MATHEMATICS

A thesis submitted for the fulfillment of the requirements of :

The Doctorate Degree in Mathematics

Option : **Numerical Analysis and Optimization**

By

HASSOUNA Houda

Title :

Gaussian Process for Image Classification

Members of the jury :

MELKEMI Khaled	Pr.	University of Batna	President
MOKHTARI Zouhir	Dr.	University of Biskra	Supervisor
BELLAGOUN Abdelghani	Dr.	University of Biskra	Examiner
KHELIL Nacer	Dr.	University of Biskra	Examiner
ZERROUG Abdelhamid	Dr.	University of Biskra	Examiner

February 2016.

People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

MOHAMED KHIDER UNIVERSITY, BISKRA

FACULTY of EXACT SCIENCES, SIENCE of NATURE and LIFE

DEPARTMENT of MATHEMATICS

A thesis submitted for the fulfillment of the requirements of :

The Doctorate Degree in Mathematics

Option : **Numerical Analysis and Optimization**

By

HASSOUNA Houda

Title :

Gaussian Process for Image Classification

Members of the jury :

MELKEMI Khaled	Pr.	University of Batna	President
MOKHTARI Zouhir	Dr.	University of Biskra	Supervisor
BELLAGOUN Abdelghani	Dr.	University of Biskra	Examiner
KHELIL Nacer	Dr.	University of Biskra	Examiner
ZERROUG Abdelhamid	Dr.	University of Biskra	Examiner

February 2016.

To my family.

Abstract

Compared to state-of-the-art classifiers, the Gaussian process classifier (GPC) offers several attractive properties. For instance, their Bayesian nature gives the possibility to integrate any kind of prior information in the classification process. They allow a full automatic estimation of the hyperparameters. Feature selection may be part of the learning process by using appropriate kernels. Moreover, in addition to the class posterior probability estimate used to perform the decision, they yield a variance estimate that can be exploited as a confidence value on the provided decision. In order to improve the GPC capabilities, in this thesis, we propose to reformulate the GPC learning model so as to integrate spatial contextual information. Though it has been shown for numerous other classification approaches that the exploitation of such information can be potentially attractive to increase the classification accuracy, little attention has been given to do so for GPC. All the mathematical developments leading to the proposed Spatial GPC (SGPC) are described. Experimental results show that the SGPC can help in improving the classification accuracy compared to the baseline GPC.

Keywords : **Gaussian processes, image classification, Laplace approximation, spatial contextual information**.

Résumé

Par rapport à l'état des classificateurs d'art, le processus Gaussien de classification offre plusieurs propriétés intéressantes. Par exemple, sa nature Bayésienne donne la possibilité d'intégrer tout type d'information préalable dans le processus de classification. Ils permettent une estimation automatique complète des hyperparamètres. La sélection de fonction peut faire une partie du processus d'apprentissage à l'aide de noyaux appropriés. En plus de classe d'estimation de probabilité posterior utilisé pour effectuer la décision, ils cédent une estimation de la variance qui peuvent être exploités comme une valeur de confiance sur la décision prévue. Afin d'améliorer les capacités de GPC, dans notre thèse, nous proposons de reformuler le modèle d'apprentissage GPC afin d'intégrer l'information contextuelle spatiale. Cependant il a été démontré pour de nombreuses autres méthodes de classification que l'exploitation de ces informations peut être potentiellement attrayant pour augmenter la précision de la classification, peu d'attention a été donnée de GPC. Tous les développements mathématiques menant à GPC spatiale proposé (SGPC) sont décrit. Les résultats expérimentaux montrent que SGPC peut aider à améliorer la précision de la classification par rapport à la ligne de base GPC.

Mots clés : **processus Gaussien, classification d'image, approximation de Laplace, information contextuelle spatiale**.

Acknowledgement

So many believed in me and in what this work is really worth and for whom thanks can never be enough to express my deep appreciation. This section is my wish to acknowledge those many who have lent a hand of assistance and whispered a word of encouragement and have made this work possible.

I would like to thank God for his blessing and guidance in the process of completing this research work.

My deepest gratitude is to my advisor, MOKHTARI Zouhir, Doctor of University of Biskra, Algeria. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. Don taught me how to question thoughts and express ideas. His patience and support helped me overcome many crisis situations and finish this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D. study.

I would like to thank MELGANI Farid, Doctor of University of Trente, Italy, who shared me his ideas and offered his time and helpful comments. His insightful tips have been my stars and compass that guided me safely ashore.

I would especially like to thank MELKEMI Khaled, Professor of University of Batna, Algeria, who was the director of my Master thesis (2013) for his insightful comments and encouragement.

My sincere thanks also goes to the rest of my thesis committee: Dr. BELLAGOUN Abdelghani, Dr. KHELIL Nacer, and Dr. ZERROUG Abdelhamid, for accepting to evaluate this thesis.

I wish to thank Dr. CHALA Abdelouahed and Dr. YAHIA Djebrane, who have encouraged me all along.

I am much obliged to several friends : Mohamed, Rim, Bochra and Abdallah, students of University of Trente, Italy, who helped to steer my work in different ways : supporting me morally by keeping up my spirits when they wilted.

Symbols and Abbreviations

The different symbols and abbreviations used in this thesis.

$q_{\setminus i}(.)$	:	approximate cavity function.
$J_{D/.}$	:	bessel function of order $D/$.
G	:	binarized image.
$P_{\setminus i}(.)$	:	cavity distribution.
$\mathbf{c}$	:	centre of the basis function Φ.
l	:	characteristic length scale.
$\varphi_B(I)$	:	closing of an image I by structuring element B.
f^*	:	complex conjugation of f.
$\mathbf{C}$	:	complex numbers.
$\mathbf{f}\|\mathbf{X}$ and $P(\mathbf{f}\|\mathbf{X})$	:	conditional random variable $\mathbf{f}$ given $\mathbf{X}$ and its probability (density).
C	:	contrast.
$\mathbf{K}$, $\grave{\mathbf{K}}$	:	covariance matrix.
$k,\ k(x,\ \grave{x})$	:	covariance matrix, covariance matrix between x and $\grave{x}$.
$L_1,\ L_2$	:	degrees of brightness.
$\mathbf{f}^{\setminus i}$	:	denotes $\mathbf{f}$ without f_i.
d	:	depth of the image.
$\mid \mathbf{K} \mid$	:	determinant of $\mathbf{K}$ matrix.

ς	:	diffusion coefficient.
$\delta_B(O)$	:	dilation of a set O by structuring element B.
D	:	dimension of input space.
$\sim$	:	distributed according.
$P(.)$	:	distribution.
$\triangleq$	:	equality which acts as a definition.
$\varepsilon_B(O)$	:	erosion of a set O by structuring element B.
$E[x]$	:	expectation at x.
$\Phi_C(x_i)$	:	feature map of input x_i.
J	:	function of light intensity and color.
$\Gamma(.)$	:	gamma function.
$q(.\|\bullet)$	:	Gaussian approximation.
ϕ	:	Gaussian cumulative distribution function.
X	:	Gaussian process.
$\overline{\mathbf{f}}_*$	:	Gaussian process posterior mean.
$\mathbf{f}_*$	:	Gaussian process (posterior) prediction (random variable).
$f(\mathbf{x})$ or $\mathbf{f}$	:	Gaussian process (or vector of) latent function value.
$\mathcal{F}(x, y)$	:	grayscale of the pixel at the coordinates $(x,\ y)$.
$\nabla\nabla$	:	(Hissian) matrix of second derivatives.
I	:	image.
χ	:	input space.
η	:	interest rate.

$\mathbf{K}^{-1}$	:	inverse of matrix $\mathbf{K}$.
y_*	:	label of test sample x_*.
$\grave{\mathbf{y}}$, $\mathbf{y}$	:	labels.
f_*	:	latent function value.
LUT	:	look up table.
$\mathbf{X}$, $\grave{\mathbf{X}}$	:	matrix of training data.
$\mu,\ \mu(x)$	:	mean function, mean function at x.
$\hat{\mathbf{f}}'$ and $\hat{\mathbf{f}}$	:	means.
$\boldsymbol{\mu}$	:	mean vector.
θ	:	measure.
$min,\ max$	:	minimum and maximum respectively.
k_ν	:	modified bessel function.
$\mathbf{N}$	:	natural numbers.
A_1 and A_2	:	neighboring regions of an image.
$\mathcal{N}$	:	normal distribution.
σ_i^2	:	noise variance.
C	:	number of classes.
N and $\grave{N}$	:	numbers of samples.
O	:	object.
$\gamma_B(I)$	:	opening of an image I by structuring element B.
p	:	order of polynomial.
∇	:	partial derivatives.

$\mathbf{p}$	:	pixel (point) of an image.
Σ_p	:	prior covariance matrix.
$W_t,\ U_t,\ S_t,\ B^H$	:	processes.
$\mathbf{R}$, $\mathbf{R}_+$, $\mathbf{R}^D$	:	real numbers, positif real numbers, real numbers of dimension D.
Θ	:	set of hyperparameters.
$\mathcal{A}$	:	set of measurables.
$\mathbf{y}_n^*$	:	spatial neighbors of y_*.
$S(s)$	:	spectral density at frequency s.
ρ	:	spectral mesure.
$B,\ B_{\mathbf{p}}$	:	structuring elements.
$\breve{B}$	:	symmetrical structuring element of B.
x_*	:	test sample.
$\mathbf{S}$	:	threshold.
$\boldsymbol{D}, \grave{\boldsymbol{D}}$	:	training set.
$\mathbf{y}^T$	:	transpose of vector $\mathbf{y}$.
σ_f^2	:	variance of the signal.
$var(x),\ Cov(x,\ \grave{x})$	:	variance of x and covariance between x and $\grave{x}$.
Ω	:	vector of hyperparameters of the covariance function.
$\grave{\mathbf{f}}$	:	vector of latent function value.
w	:	weight.

Contents

List of Figures

List of Tables

General introduction

Recently, a new machine learning approach that is based on the Gaussian process (GP) theory has been introduced. It represents a powerful and interesting theoretical framework for Bayesian regression and classification. Despite it has gained prominence in recent years, it remains an approach whose potentialities are not yet sufficiently exploited in particular in the remote sensing field.

According to this approach, the learning of a machine (regressor or classifier) is formulated in terms of a Bayesian estimation problem, where the parameters of the machine are assumed to be random variables which are a priori jointly drawn from a Gaussian distribution. Compared to other regression methods, GP regression has several advantages:

1) Its prediction equation is much simpler and it is given in an analytical form.

2) To each predicted output it associates a confidence measure.

3) Thanks to its Bayesian formulation, the model selection issue is handled in an automatic way, as demonstrated by different works recently published in the literature, including Pasolli et al [56], Bazi et al [6] and Hultquist et al [32].

As very well described in Rasmussen and Williams [62], the main idea of GP classification (GPC) is to assume that the probability of belonging to a class label for an input sample is monotonically related to the value of some latent function (logistic or probit functions) at that sample. Such monotonic relationship is defined according to a so-called squashing function.

A Gaussian process prior characterized by a covariance matrix embedding a set of hyperparameters is placed on this latent function. The inference is made by integrating over the latent function. Since such integral is analytically intractable, solutions based on Monte Carlo sampling or analytical approximation methods can be adopted. The multiclass implementation of GPCs is obtained through an intrinsic multiclass formulation, which can be complex, or simply by decomposition into binary classification problem. Among the few works reported in the remote sensing literature dealing with GPC, on can find Bazi and Melgani [8], and Sun et al [73].

Up to now, to the best of our knowledge, the integration of spatial contextual information in a GPC model has not yet been envisioned for classifying remote sensing imagery. An apparently close work can be found in [34], where GP regression is used to exploit spatial coordinates of the training samples for predicting mean vectors. The classification task is performed by means of a maximum likelihood (ML) classifier.

In our case, we exploit spatial contextual information, which is different from spatial coordinates, and embed it in a GPC model. It is well-known that spatial contextual information can be useful, if well exploited, to improve the classification accuracy by opportunely capturing local spatial correlation conveyed in the image under analysis.

Our thesis is organized in 4 chapters that allow us to present the different aspects of our work.

In the first chapter we introduce the general notions of images processing, giving the definition of the image, the characteristics of digital image and the most filters used for improving the quality of images.

The second chapter gives the general overview on the Gaussian process and the covariance functions.

The third chapter will be dedicated to the presentation of the classification by the Gaussian process using the Laplace approximation for binary GP classifier or Expectation propagation.

And the last chapter contains the fruit of our work, the obtened results of the classification by Spatial Contextual Gaussian Process Classification (SGPC) enriched by the results and the interpretation of these.

Chapter 1

Generalities on images processing

Today, the image constitutes one of the most important tools used by the people in order to communicate with each other. It is universal tool of communication of which the richness of content allows people of different ages and all cultures to understand each other.

Images processing is a set of methods and operating technical on those latter it on order to ameliorate the visual aspect of the image and to extract relevant judged information that we will use in different applications for instance : the recognition, the classification, ... etc.

In this chapter, we present some principal concepts of images processing which are matched to our subject of study.

1.1 Definition of the image

The image is a representation of a person or object through painting, sculpture, drawing, photography, film, ... etc. It is also a set of structured information that, become after the display on the screen, meaningful to the human eye.

It can be described as an analog brightness continuous function $J(x, y)$ defined in a bounded domain, where x and y are the spatial coordinates of a point of the image and J is a function of light intensity and color. In this aspect, the image is unusable by the machine, which requires its digitization.

1.2 Definition of the digital image

The term of digital image refers, in its most general sense, to any image that has been acquired, processed and stored in encoded form, represented in numbers (numerical values).

The digitization is the process that allows passing the state of physical image (optical image, for example) that is characterized by the continuous appearance of the signal, that it represents (infinite value of the light intensity, for example), to the state of a digital image that is characterized by the appearance discrete (light intensity can take only values quantized into a finite number of distinct points). It is this digital form which allows further exploitation by software tools on computer.

1.3 Types of images

1.3.1 Raster image (bitmap)

A raster image (bitmap) (see Fig 1.1) is an image in point model. The most universal coding system consists in decomposing the graphic representation, the image, a some number of elementary points characterized by their spatial coordinates and color.

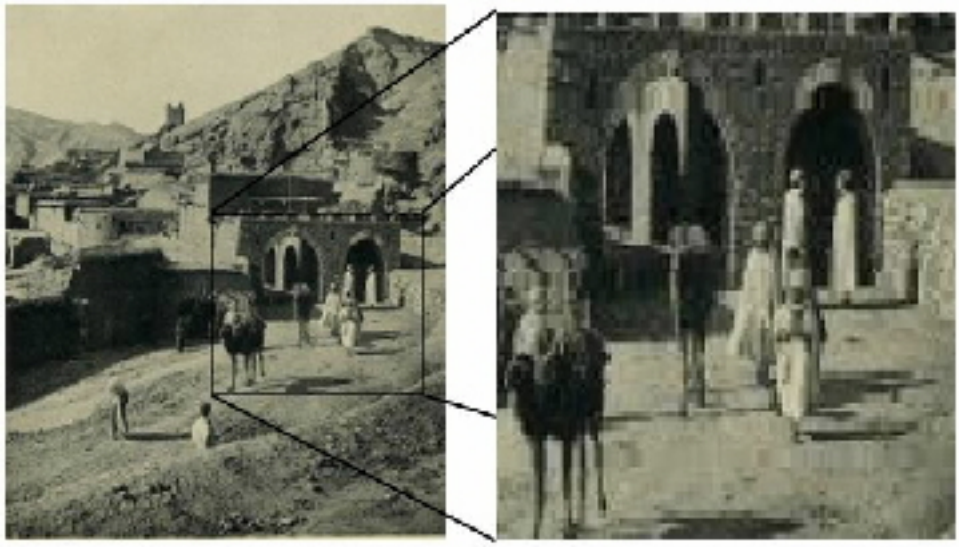


Figure 1.1: Raster image

1.3.2 Vector image

In a vector image, the data are represented by simple geometrical forms which are described at a mathematical point of view. It comes to represent the images data by geometric formulas that will able to be in described a mathematical way. In other words, storing the operations sequence leading to the trak is stored in the case of a vector image, then it stores a mosaic of point elementary in the case of raster image.

These images present two advantages : they occupy little memory space and they can be resized without lossing of an information.

1.4 Tagged Image File Format (TIFF)

1.4.1 Concepts and definitions

Pixel

The pixel represents the smallest component of raster image. The word pixel comes from an abbreviation of British expression PICture Element. The numerical value of pixel represents a luminous intensity.

The pixel coding

Almost, the value of pixel is a binary word of length d bits, therefore a pixel can be taken one of the values of the interval $[0 \dots 2^{d-1}]$. The value d is called the depth of the image. All these pixels are contained in a two-dimensional table (a matrix) constituting the obtained image.

The size of an image

The size of an image is the number of pixels of the image, the size of an image which is represented by (328×456) where 328 is the number of rows, and 456 is the number of columns is equal at : $328 \times \ 456 = 149568$ pixels.

The resolution of an image

In the field of digital imaging, the resolution is a measure of the sharpness of the display or capture of an image, expressed in number of pixels per unit of area, that means (the density) in pixels.

The resolution of a digital image is expressed in PPI (Pixels Per Inch). When the resolution of an image is big, its quality is better (see Fig 1.2).

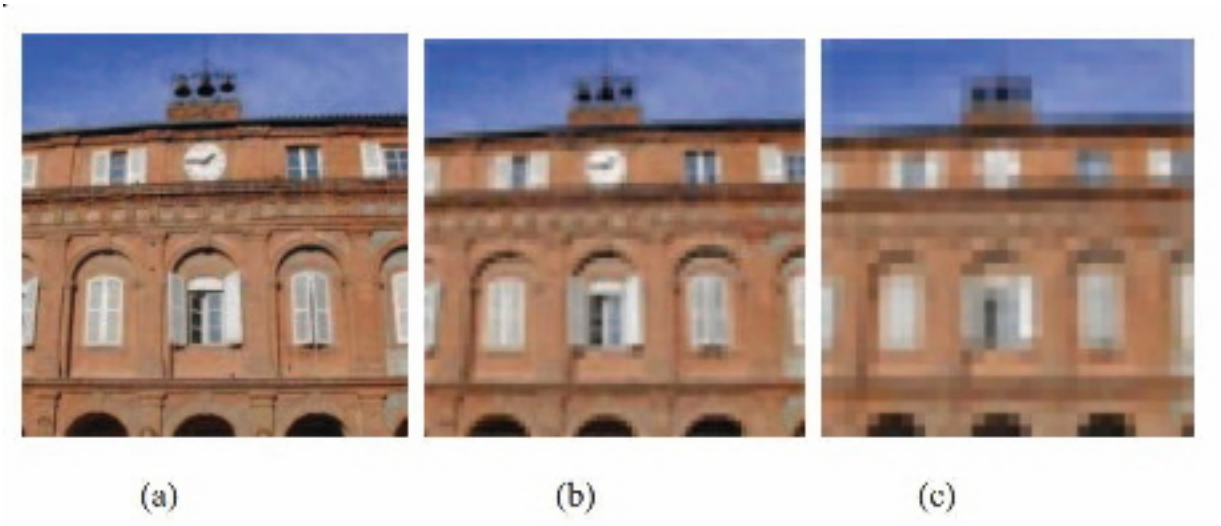


Figure 1.2: The resolution of an image : (a) an image acquires at 256 dpi, (b) an image acquires at 64 dpi, (c) an image acquires at 32 dpi. [4]

The luminance

The word luminance is substituted for the word brilliance, matched to the eclat of an object. The luminance is the brightness of the pixels of an image. It is also defined as the intensity of the extended source in a given direction, divided by the apparent area from this source in the same direction.

The contrast

It is the marked opposition between two regions of an image, more precisely between the dark region and clear region of this image. The contrast is defined as a function of the luminances of two regions of an image. If L_1 and L_2 are the degrees of brightness of two neighboring regions respectively A_1 and A_2 of an image, the contrast C is defined by :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \tag{1.1}$$

The noise

A registration system of an image does not restore the image perfectly. Actually parasites information are added by the random manner to details of the original scene which we call it : noise.

The noise has different origins but it causes the similar effects as the loss of sharpness on the detail or the appearance of grains.

The histogram

The histogram of grayscale or color images is a function that associates each intensity value of the number of image pixels with that value (see Fig 1.3).

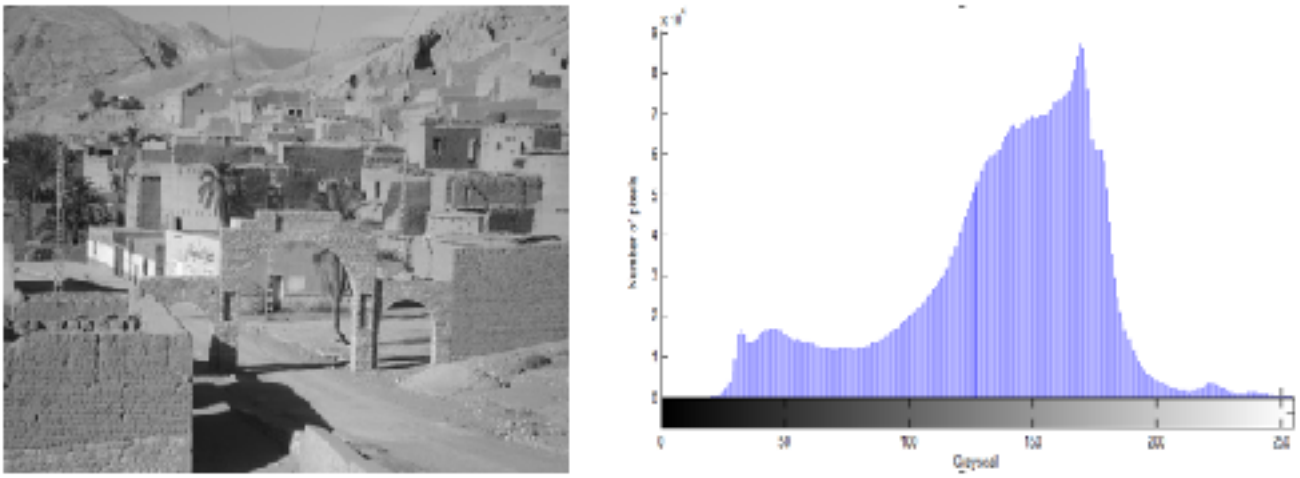


Figure 1.3: Example of histogram of an image.

The colors palette

We call the color palette, the list of colors that can contain an image. The value of each pixel represents the rank of a color in this list. It is frequent to see images never use certain colors, this makes it interesting to limit the color palette by selecting only color or colors actually used by the image (see Fig 1.4).

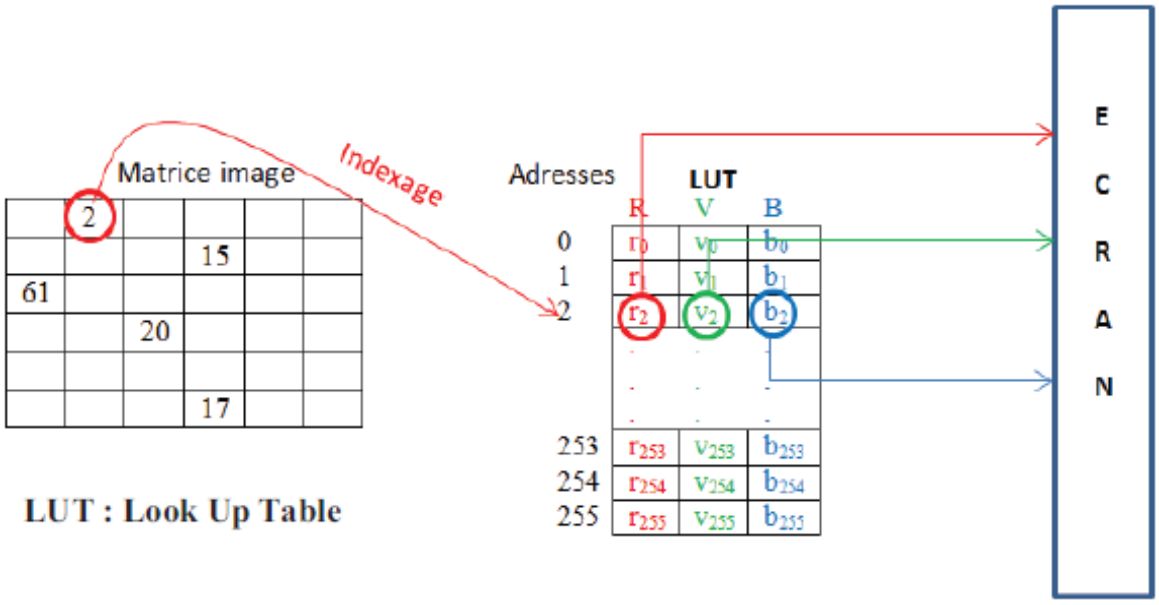


Figure 1.4: The ponctual transformation of digital image [4]

1.5 The colors coding

1.5.1 The binary images (black and white)

The binary images are images of depth $d = 1$ bit, so, a pixel can be taken one of the values: black or white (0 or 1).

It is typically, the type of the image that we use for scaning the text when it is composed of one color.

1.5.2 The grayscale images

In general, the grayscale images are images of depth $d = 8$ bits, so, each pixel can be taken one of the values of the interval [0...255], where the value 0 represents the minimum brightness (the black) and 255 the maximum brightness (the white). This type of image is frequently used to reproduce texts or photos on black and white.

In many professional applications of photography and printing as well as medicine and astronomy, 8 bits per pixel is not enough, for that there are other types of grayscale images of depth $d = 12$, $d = 14$ or $d = 16$ bits.

1.5.3 The colors images

The color space is based on the synthesis colors, i.e, the mixture between different colors (three, four, ...) gives a color .

Most colors images are based on three primary colors : Red, Green and Blue (RGB), and typically they use 8 bits for each component color, so each pixel requires $3 \times 8 = 24$ bits for coding the three components, and each color component can take one of the values of interval [0... 255].

1.6 Format of images files

An image format is a computer representation of the image, including information on the way the image is encoded and possibly providing guidance on the way to decoce and to manipulate.

The most formats are composed of a header containing attributes (image size, coding type, LUT, ...etc.), followed by data (the actual image). The structuring of attributes and data varies from one format to another. There are many images formats, we will mention a some of them.

1.6.1 Windows BitMaP (BMP)

BMP format is one of the simplest formats. It was jointly developed by Microsoft and IBM. This Technology has the main advantage the quality of images doesn't supply

compression (no quality loss). This makes it a very heavy format of image, no or little used on the internet.

1.6.2 Tagged Image File Format (TIFF)

This format is oriented to the professionals (printers, advertisars, ...) because it has the advantage of being recognized on all types of Operating System : Windows, Mac, Linux, Unix, ldots, ... etc.

It provides a image of a very a good quality, but its size remains large, although it is lower than the BMP file.

1.6.3 Joint Photographic Expert Group (JPEG)

It is the most frequent format, it is found in the Internet. It takes up little disk space. It's the developed format by photographers to transmit images of a professional photographic quality . It supports millions of colors but it has not associated the colors palette and therefore the colors can be different on machines and different systems.

1.6.4 Graphics Interchange Format (GIF)

The filles in GIF format are highly compressed while keeping a very decent quality. They have a palette of associated colors (limited to 256 colors) and they occupy little of disk space.

1.6.5 Portable Network Graphic (PNG)

PNG format uses the principle of GIF format encoding but it is not limited to 256 colors, and generally it provides a more effective compression. It allows, unlike GIF to record photographs without lossing the of quality, but with a gain of less storage space comparing to JPEG.

1.7 Some processing of images

There is a variety of images processing, we will present some examples :

1.7.1 Binarization

The binarization (and thresholding) (see Fig 1.5) is the simplest technique of classification, where the pixels of the image are shared by a single threshold **S** into two classes: those which is belonging to background and to the stage (the object). The image is then separated into two classes in a way that the information between 0 and **S** is successful and the other one not, or vice versa..

We have the image $I(M,\ N)$, supposing that $F(x,\ y)$ represents the grayscale of the pixel at the coordinates (x, y), $0 \leq x \leq M$, $0 \leq y \leq N$ and **S** is the chosen threshold , the pixels of the object are those having inferior the grayscale at **S** and the others having the level of superior gray than **S** are background pixels. So, the binarized image G is determined by the pixels $(x,\ y)$ which its value is :

$$g(x,y) \quad = \quad \begin{cases} 0 & \text{if} \quad g(x,y) < \mathbf{S} \quad , \\ 255 & \text{if} \quad g(x,y) \geq \mathbf{S} \quad . \end{cases} \tag{1.2}$$

Figure 1.5: Binarization of an image (a) original image , (b) binarized image.

1.7.2 Segmentation

The segmentation of images is an operation that is intended togather pixels with each according to predefined criteria, and it may be accomplished according to several methods. The pixels are grouped into regions that constitute a paving or partition of the image. The segmentation is an important step in image processing.

1.7.3 Skeletonization

The skeletonization procedure is performed on a binary image, and it aims to reduce the thickness track of a pixel only, while maintaining the continuity thereof (see Fig 1.6).

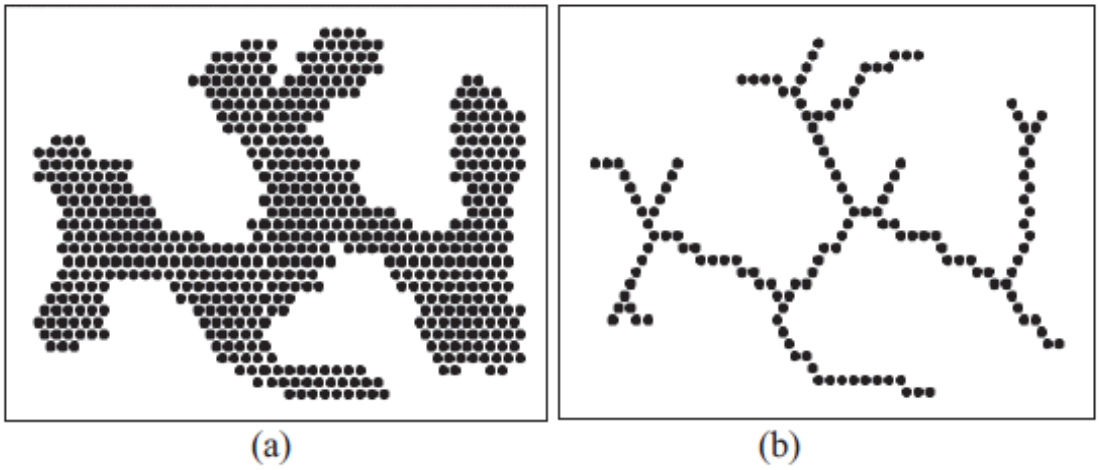


Figure 1.6: Skeletonization of an image (a) original image , (b) skeletonized image.[4]

1.7.4 Convolution

The convolution is the replacement of the value of a pixel by a combining its surroundings. It consists to scan an analysis window (mask) on all the pixels of the image.

The convolution operation is calculated at every point of the source image in 3 steps :

1) The mask is centered (for example, a square mask 3×3) on the current pixel.

2) 9 products are calculated between the value of the image and the value of superimpsed mask.

3) Then we sum the 9 products to get the pixel value of the filtered image.

1.7.5 Filtering

The notion of filtering concept is borrowed from physics and signal processing techniques. If a signal (electrical, radio, image, ... etc.) has a very different frequency components, it may be advantageous to remove some, in this case we talk about filtering.

The image filtering is a local treatment used mainly to perform a spatial analysis of an image. Its aim is to accentuate the image intensity variances, or to detect contours and

to reduce existing noises. There are a large number of filters, we can be classified into two broad categories : linear filters and nonlinear filters.

1.7.6 Mathematical morphology

Mathematical morphology (MP) is a mathematical theory and technique and structural analysis computer, it is linked to algebra and it is performed on a binary image. One of the basic ideas of mathematical morphology is to study or to treat a set with another set, called structuring element (binary mask consisting of black and white pixels), which serves as a probe.

At each position of the structuring element, we remark if it touches or it is included in the initial set. Depending on the response function, we construct a set of output. We are obtaining basic operators which are relatively intuitive.

Among the most important tools of mathematical morphology are : erosion, dilation, opening and closing.

Structuring element

The structuring element (SE) is an assembly applied to image of study. SE "plans" include a set of points without any value unlike the SE volume or points are considered. SE "plans" are so named because they have only two dimensions in the case of $2D$ images. The basic morphological operators require the definition of an origin for each structuring element.

This origin allows the positioning of the structuring element on a point or a given pixel a SE is a point $\mathbf{p}$ which means that its origin coincides with $\mathbf{p}$. A structuring element is identified by its origin. The basic structuring elements planes and isotopes for -hexagonale and square grids are represented by (Fig 1.7).

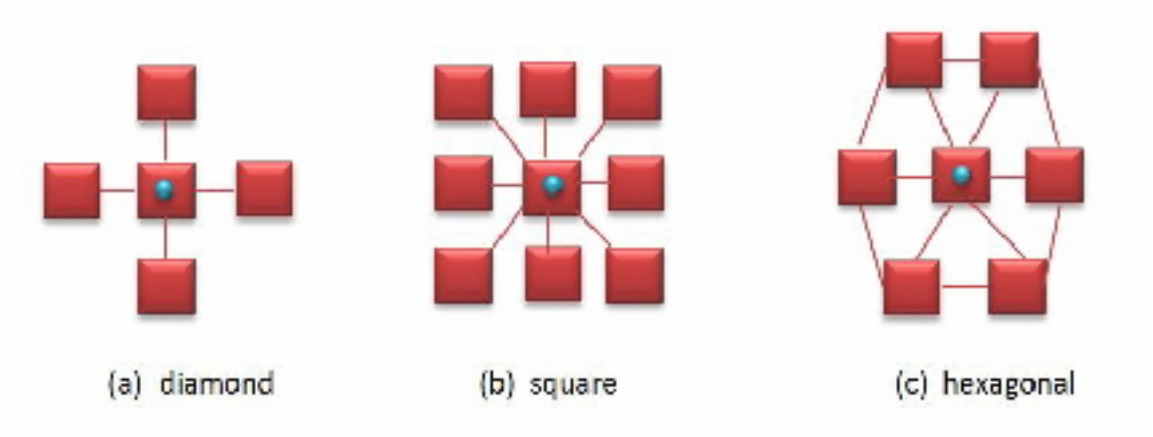


Figure 1.7: Elementary structuring elements planes and isotopes. The origin of each structuring element and its center.[48]

A structuring element defines a relation of neighbourhoud and of connectivity in an image. The relation is the center to the neighbors as shown in (Fig 1.8) which shows the neighborhood relation of a square structuring element 8− connexities.

The form and size of the structuring element must be adapted to the geometric properties of the image objects. For example, the linears SE is suitable for the extraction of linear objects. We will detail in the following sections the basic operators morphology.

Erosion

The erosion of set is to ask at each pixel $\mathbf{p}$ of an object O, the question : "is the structuring element $B_{\mathbf{p}}$ be entirely contained in O ?". The eroded set consists of the points where the answer of this question is affirmative. The treated set represents either the objects of a binary image or a subgraph of a grayscale image. The erosion of a set O by a structuring element B is denoted by $\varepsilon_B(O)$ and it is defined by the set of points $\mathbf{p}$, such as B is included in O when its origin is placed on $\mathbf{p}$.

$$\varepsilon_B(O) \quad = \quad \{\mathbf{p} \ / \ B_{\mathbf{p}} \subset O\} \quad , \tag{1.3}$$

The (Eq 1.3) can also be written as intersections from a set of translations. These

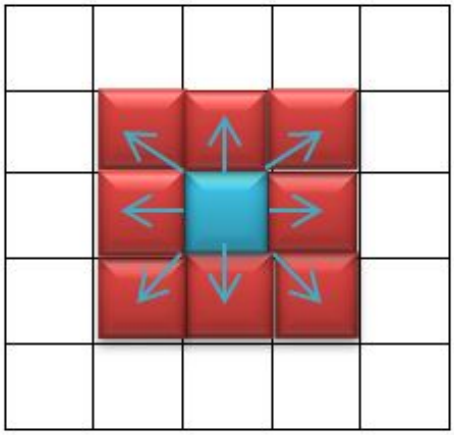


Figure 1.8: Relation of neighborhood for a square structuring element at 8−connexties. [48]

translations are defined by the structuring element.

$$\varepsilon_B(O) \quad = \quad \bigcap_{b \in B} O_{-b} \quad , \tag{1.4}$$

The previous definition can be applied on binary images and on grayscale image : the erosion of an image I by a structuring element B is denoted $\varepsilon_B(I)$ and it is defined as the minimum of translations of I by the vectors b of B.

$$\varepsilon_B(I) \quad = \quad \bigwedge_{b \in B} I_{-b} \quad , \tag{1.5}$$

Hence the eroded value of a given pixel is the minimum value of the image in the window defined by the structuring element when its origin is placed on $\mathbf{p}$:

$$\left[\varepsilon_B(I)\right](\mathbf{p}) \quad = \quad \min_{b \in B} I(\mathbf{p} + b) \quad , \tag{1.6}$$

The erosion reduces the « pics » of the grayscale and enlarge the « valleys » : it tens so, to homogenize the image, to darken and to spead the edge of the most dark objects (Fig 1.9).

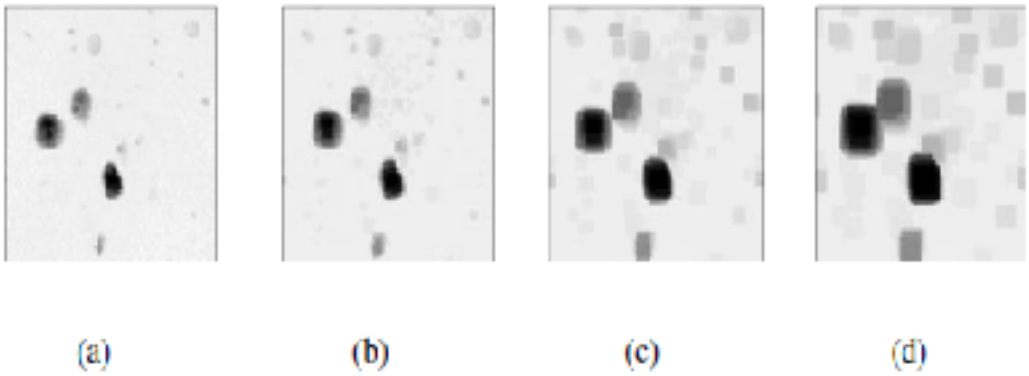


Figure 1.9: Erosion : (a) image on original grayscale, (b) erosion with SE square at size 3, (c) erosion with SE square at size 6, (d) erosion with SE square at size 10. [48]

Dilation

The dilation is the adjoint operator of erosion. It consists to ask at each pixel $\mathbf{p}$ of an object O, the question : " does the structuring element $B_{\mathbf{p}}$ intersect the set O ?" The dilated set is constitued of the pixels where the answer is affirmative.The dilation of a set O by a structuring element B is denoted as $\delta_B(O)$, and it is defined by the set of points $\mathbf{p}$ where B intersects the set O when its origin coincides with $\mathbf{p}$:

$$\delta_B(O) \quad = \quad \left\{ \mathbf{p} \ / \ \breve{B}_{\mathbf{p}} \cap O \neq 0 \right\} \qquad , \tag{1.7}$$

In the (Eq 1.7), $\breve{B}$ designates the symmetrical structuring element of B. Noting that the dilatation of a single point gives as output the centered structuring element on its origin. The (Eq 1.7) can also be written in the form of unions of translations. These translations are defined by the structuring element.

$$\delta_B(O) \quad = \quad \bigcup_{b \in B} O_b \quad , \tag{1.8}$$

The previous definition can be also applied on the binary images also on the grayscale images : the dilation of an image I by a structuring element B is denoted by $\delta_B(I)$ and it

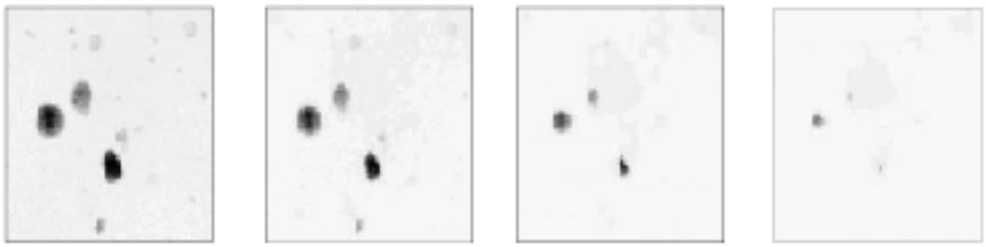
is defined by the maximum of translation of I by the vectors b of B.

$$\delta_B(I) \quad = \quad \bigvee_{b \in B} I_b \quad , \tag{1.9}$$

Hence the dilated value of a given pixel is the maximum of the image in the window defined by the structuring element when its origin is placed on $\mathbf{p}$:

$$[\delta_B(I)](\mathbf{p}) \quad = \quad \max_{b \in B} I(\mathbf{p} + b) \quad , \tag{1.10}$$

This transformation fills the "valleys" and it thickens "peaks" : it homogenizes the image, thinning and tending to make disappear the dark objects.



Dilation : (a) original grayscale image, (b) dilation with SE square at size 3, (c) dilation with SE square at size 6, (d) dilation with SE square at size 10. [48]

Properties

Dilation and erosion are adjoint transformations and they respect the principle of complementarity. This means that erosion on an image is equivalent to the complementary of the dilation on the complementary image with the same structuring element (and vice versa).

Morphological opening

The opening of an image I by a structuring element B is denoted $\gamma_B(I)$ and it is defined by erosion of I by B followed by a dilation of I by the symmetrical structuring element $\breve{B}$.

$$\gamma_B(I) \quad = \quad \delta_{\breve{B}} \circ \varepsilon_B(I) \ . \tag{1.11}$$

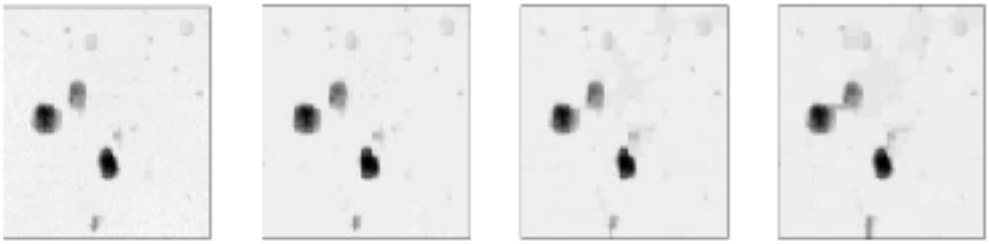


Figure 1.10: Opening : (a) original grayscale image, (b) opening with SE square at size 3, (c) opening with SE square at size 6, (d) opening with SE square at size 10. [48]

Opening removes the peaks but preserves the valleys, it homogenizes the image but it preserves the dark objects.

Morphological closing

The closing of an image I by a structuring element B is noted $\varphi_B(I)$ and it is defined by the dilation of I by the structuring element B, followed by the erosion by the symmetrical structuring element $\breve{B}$:

$$\varphi_B(I) \quad = \quad \varepsilon_{\breve{B}} \circ \delta_B(I) \ . \tag{1.12}$$

The closing fills the valleys, it homogenizes and it brightens the image as shown in (Fig 1.11).

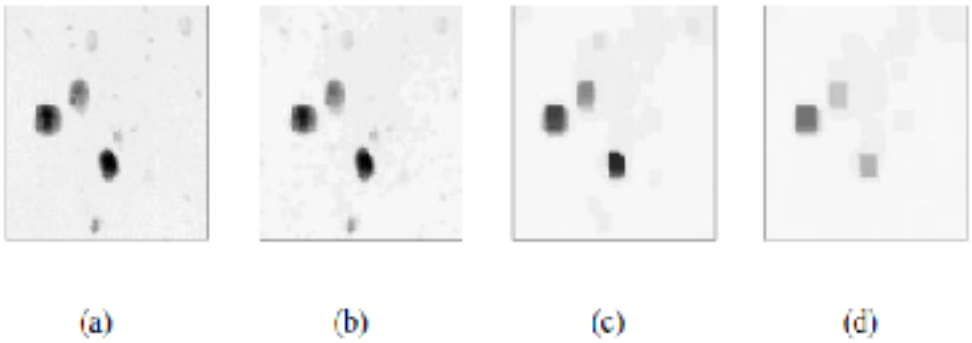


Figure 1.11: Closing : (a) original grayscale image, (b) closing with SE square at size 3, (c) closing with SE square at size 6, (d) closing with SE square at size 10. [48]

Chapter 2

Gaussian Process

Gaussian process (GP) is very promising novel machine learning concept that is based on a probabilistic model of the underlying function/class probabilities. It represents a powerful and interesting theoretical framework for Bayesian classification.

In this chapter we will see the Gaussian process definition, its covariance functions and some examples of them.

2.1 A brief history of Gaussian process

The study of Gaussian processes and their use for prediction is far from new [37]. Indeed, the underlying theory dates back to Weiner-Kolmogorov prediction theory and time series analysis in the 1940's [37], [62], [28] and [42]. More recent is the introduction of kriging [45], and its subsequent development as a method for the interpolation of geostatistical data [16].

Kriging, named after the mining engineer D.G.Krige, is identical to Gaussian process regression, but is derived and interpreted somewhat differently to that above (e.g. see [33]).

Furthermore, as a geostatistical method, it is mainly concerned with low-dimensional problems and tends to ignore any probabilistic interpretations [37]. In the wider statistical community, the use of Gaussian processes to define prior distributions over functions dates back to 1978, where O'Hagan [52] applied the theory to one-dimensional curve fitting.

In the machine learning community, the use of Gaussian processes for supervised learning is a more recent development which traces back to introduction of back-propagation for learning in neural networks [64]. This original non-probabilistic treatment was subsequently enhanced by Buntine [11], MacKay [43], and Neal [51] who introduced a Bayesian interpretation that provided a consistent method for handling network complexity (see [40], [9] and [36] or reviews).

Soon after, Neal [49] showed that under certain conditions these Bayesian Neural Networks converge to Gaussian processes in the limit of an infinite number of units. This resulted in the introduction of Gaussian processes for regression in a machine learning context [83], [59] and [50].

Briefly, this work included a description of how to :

1) Specify and parameterise a covariance function.

2) Build a covariance matrix and hence express the prior distribution over function values.

3) Find the posterior distribution over parameters using Bayes' Theorem.

4) Either optimise to find the most likely (ML) or maximum a posteriori (MAP) parameters, or integrate over the posterior density using Hamiltonian Monte Carlo.

5) Calculate the predictive distribution at any test point.

For good introductions to Gaussian processes for regression refer to the 1997 thesis of Gibbs [24], the Gaussian processes chapter in MacKay's book [37], and the recent book by Williams and Rasmussen [62]. Additionally, Seeger provides recent reviews [65], [66] and [67] and relates Gaussian processes for machine learning to other kernel machine methods.

Since the original introduction of Gaussian processes for regression, there have been numerous enhancements and applications. One of the main areas of interest has been on developing methods to reduce the computational cost of Gaussian process regression, both in the training and prediction phases.

The fundamental problem is that for a training set of size N, exact calculation of the marginal-likelihood (Eq 3.24) has complexity $O(N^3)$. This cost is a direct result of inverting an $N \times N$ matrix, so some of the methods aim to approximate this calculation. For example, [24], [25] describe and analyse an iterative method to approximate the inverse with complexity $O(N^2)$. Another interesting approach is presented by Williams et al.[84], [85], who make use of the Nyström method to form a rank $m < N$ matrix approximation to the covariance matrix, which can then be inverted with a cost $O(m^2 N)$.

There have been many more recent developments (e.g. [69], [76], [17], [18], [19] and [68]). For a good review and summary of these methods see [62] and [57].

Other recent work has been extensive and varied. For example, Gibbs [24] and Paciorek [54] and [55] developed methods for creating non-stationary covariance functions, and hence, models of non-stationary data. We have seen methods to deal with input-dependent noise [29] and non-Gaussian noise [70]. Mixtures of Gaussian processes were introduced by [77] followed by an extension to a tractable infinite mixture of Gaussian processes experts [63].

Interesting machine learning applications include Gaussian processes for reinforcement learning [61], the incorporation of derivative observations into Gaussian process models [72], Gaussian processes to speed up the evaluation of Bayesian integrals [60], and Gaussian process models of dynamical systems [80].

Gaussian processes have also proved useful for classification problems. However, in this case the likelihood function and evidence and hence the posterior distribution are not Gaussian, so exact inference is not possible. As a result, much work has gone into developing approximations.

Many of the resultant classifiers make use of the Laplace approximation [5], Markov Chain Monte Carlo [50], and variational methods [24] and [26]. Although Gaussian process classifiers are powerful and promising, this thesis is concerned only with Gaussian processes for classification.[10]

2.2 Gaussian process definition

Formally, when a stochastic process $f(\mathbf{x})$ is stated Gaussian, the joint distribution of any subset of its aleatoires variables $\mathbf{f} = \{f_1,\ f_2, ...,\ f_N\}$ on finite indices sets $\mathbf{X} = \{\mathbf{x}_1,\ \mathbf{x}_2, ...,\ \mathbf{x}_N\}$, by hypothesis, the form of $\mathcal{N}$ normal multivariate distribution dimensions. By adopting this hypothesis, we get a Gaussian process that are completely

defined by their mean function μ and the covariance function k. These functions allow to obtain the distribution.

$$P(\mathbf{f}) \quad = \quad \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \quad , \tag{2.1}$$

where $\mathcal{N}$ present the normal distribution, the mean vector $\boldsymbol{\mu}$ is composed of the element $\mu_i = \mu(\mathbf{x}_i)$, the covariance matrix $\mathbf{K}$ is constructed such that $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. While the functions μ and k have a direct influence on the joint distribution (Eq 2.1) , The various forms that these functions can take cause process with differents behaviors.

Note that it is also possible to define the mean and covariance functions on a finite space, in which case the Guassian process becomes simply a normal multivariate distribution. The Gaussian process is a general may be used on a large type of spaces.

Définition 2.1 *A process is called Gaussian if all its finite dimensional laws* $\mathcal{L}(X_{t1}, ..., X_{tN})$ *are Gaussians* $(\forall N \in \mathbf{N}, \forall t_1, ..., t_N \in T)$.
In other words $X = (X_t)_t$ *is Gaussian if all the linear combinaison* $a_1 X_{t1} + ... + a_n X_{tN}$ *are Gaussian (for all* $\forall N \in \mathbf{N}$, $\forall t_1, ..., t_N \in T$ *and* $a_1, ..., a_N \in \mathbf{R}$).

It's well known that the Gaussian vector $(X_{t1}, ..., X_{tN})$ is known (via its characteristic function) by the mean vector $(E[X_{t1}], ..., E[X_{tN}])$ and the covariance matrix $(Cov(X_{ti}, X_{tj}), 1 \leq i, j \leq N)$.

As soon when we understand that any law of a Gaussian process is known as soon as one takes the mean function. $\mu(t) = E[X_t]$ and the covariance operator $k(s, t) = cov(X_s, X_t)$.

Indeed, the finite dimensional laws of $(X_{t1}, ..., X_{tN})$ is the normal laws of dimension N , $\mathcal{N}(\boldsymbol{\mu}_N, \mathbf{K}_N)$ with $\boldsymbol{\mu}_N = (\mu(t_1), ..., \mu(t_N))$ and $\mathbf{K}_N = (k(t_i, t_j))_{1 \leq i,j \leq N}$. So, the functions μ and k define all the finite dimensional laws of X and therefore its law.

2.3 Examples of Gaussian process

2.3.1 Brownian motion

Let $T = \mathbf{R}_+$, the Brownian motion (BM) $(W_t)_{t\geq 0}$ is the Gaussian process defined by $E[W_t] = 0$ et $k(s,\, t) = min(s,\, t)$. Also it is called process of Wiener.

2.3.2 Brownian bridge

Let $T = [0, 1]$, brownian bridge $(W_t^0)_{t\in[0,1]}$ is the centred Gaussian process defined by the covariance function $k(s,t) = min(s,\, t) - st$.

2.3.3 Process of Ornstein-Uhlenbeck

Let $T = \mathbf{R}$, the process of Ornstein-Uhlenbeck (O.U) is the centred Gaussian process defined by :

$$U_t \quad \simeq \quad e^{-t/2} W(e^t) \quad , \tag{2.2}$$

where W is a MB. It is easily to shown that $U_t = \mathcal{N}(0,\, 1)$ because $var(U_t) = 1$, so, this process is stationary.

Its covariance function is given by :

$$k(s,t) \quad = \quad e^{-|t-s|/2} \quad , \tag{2.3}$$

It depends only on the difference $(t - s)$, it is indeed a stationary process of more simply covariance function given by $k(t) = e^{-|t|/2}$. It is given under integral form with the spectral measure $\rho(du) = \dfrac{1}{\pi}\dfrac{du}{1+u^2}$.

2.3.4 Geometric Brownian

It is not a Gaussian process but the exponential of Gaussian process. It is

$$S_t \quad = \quad x\ exp(\eta t + \varsigma W_t - \varsigma^2 t/2) \quad . \tag{2.4}$$

Such a process modeled the cours of an active S_t submitted has an interest rate η and it has a volatility ς and which is worth x at time 0.

Assuming it is found as Samuelson that the returns between two periods are measured by logarithms of cours S_t.

More than we suppose that the returns between 0 and t follow a tend Brownian motion of drift $\eta - \varsigma^2/2$ and a diffusion coefficient ς.

2.3.5 Gaussian white noise

Let $(\mathcal{A},\, \theta)$ a measured space and $\boldsymbol{U} = \{\mathbf{A} \in \mathcal{A},\ \ \theta(\mathbf{A}) < +\infty\}$.

The white noise is a Gaussian process $(X_{\mathbf{A}})_{\mathbf{A}\in\mathcal{A}}$ indexed by the set of measurables $\mathcal{A}$ defined by $E[X_{\mathbf{A}}] = 0$ and $Cov(X_{\mathbf{A}},\, X_{\mathbf{B}}) = \theta(\mathbf{A} \cap \mathbf{B})$.

We must understand the white noise as a random noise $\mathbf{A} \to X_{\mathbf{A}}(\omega)$. It is random because $X_{\mathbf{A}}$ depends of ω.

2.3.6 Fractional Brownian motion

Let $T = \mathbf{R}_+$. the fractional Brownian motion (MBF) $(B^H(t))_{t\geq 0}$ is the centered Gaussian process defined by the covariance function :

$$k(s,t) \quad = \quad \tfrac{1}{2}(|s|^{2H} + |t|^{2H} - |s-t|^{2H}) \quad . \tag{2.5}$$

2.4 Covariance functions

To specify a particular GP prior, we need to define the mean $\boldsymbol{\mu}$ and covariance $\mathbf{K}$ of where

$$P(\mathbf{f}|X) \quad = \quad \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \quad . \tag{2.6}$$

The GPs we will use as priors will have a zero mean. Although this sounds restrictive, offsets and simple trends can be subtracted out before modelling, and so in practice it is not. It is worth noting however, that the posterior GP $P(\mathbf{f}|D)$ that arises from the regression is not a zero mean process.

The important quantity is the covariance matrix $\mathbf{K}$. We construct this from a covariance function $k(x, \grave{x})$:

$$K_{ij} \quad = \quad k(x_i, x_j) \quad , \tag{2.7}$$

This function characterises the correlations between different points in the process :

$$k(x, \grave{x}) \quad = \quad E[f(x)\ f(\grave{x})] \quad , \tag{2.8}$$

where E denotes expectation and we have assumed a zero mean. We are free in our choice of covariance function, so long as the covariance matrices produced are always symmetric and positive semidefinite ($\mathbf{v}^T\mathbf{K}\mathbf{v} \geq 0, \ \forall \mathbf{v}$).

The particular choice of covariance function determines the properties of sample functions drawn from the GP prior (e.g. smoothness, lengthscales, amplitude, ... etc). Therefore it is an important part of GP modelling to select an appropriate covariance function for a particular problem.[18]

2.5 Examples of covariance functions

2.5.1 Stationary covariance functions

It will be convenient to allow kernels to be a map from $x \in \chi$, $\grave{x} \in \chi$ into $\mathbf{C}$ (rather than $\mathbf{R}$). If a zero-mean process f is complexvalued, then the covariance function is defined as $k(x, \grave{x}) = E[f(x)\ f^*(\grave{x})]$, where * denotes complex conjugation.

A stationary covariance function is a function of $\tau = x - \grave{x}$. Sometimes in this case we will write k as a function of a single argument, i.e. $k(\tau)$.

The covariance function of a stationary process can be represented as the Fourier transform of a positive finite measure.

Théorème 2.1 *(Bochner's theorem) A complex-valued function k on $\mathbf{R}^D$ is the covariance function of a weakly stationary mean square continuous complexvalued random process on $\mathbf{R}^D$ if and only if it can be represented as*

$$k(\tau) \quad = \quad \int_{\mathbf{R}^D} e^{2\pi i s.\tau}\ d\theta(s) \quad , \tag{2.9}$$

where θ is a positive finite measure.

The statement of Bochner's theorem is quoted from Stein [74][1999, p. 24]; a proof can be found in Gihman and Skorohod [27][1974, p. 208]. If θ has a density $S(s)$ then S is known as the spectral density or power spectrum corresponding to k.

The construction given by (Eq 2.9) , puts non-negative power into each frequency s; this is analogous to the requirement that the prior covariance matrix Σ_p on the weights $w \sim \mathcal{N}(0, \Sigma_p)$ be non-negative definite.

In the case that the spectral density $S(s)$ exists, the covariance function and the spectral density are Fourier duals of each other as shown in (Eq 2.10); this is known as the Wiener-Khintchine theorem, see, e.g. [14].

$$k(\tau) = \int S(s)e^{2\pi i s.\tau}\, ds \quad , \quad S(s) = \int k(\tau)e^{-2\pi i s.\tau}\, d\tau \quad . \tag{2.10}$$

Notice that the variance of the process is $k(0) = \int S(s)\, ds$, so the power spectrum must be integrable to define a valid Gaussian process.

To gain some intuition for the definition of the power spectrum given in(Eq 2.10) it is important to realize that the complex exponentials $e^{2\pi i s.x}$ are eigenfunctions of a stationary kernel with respect to Lebesgue measure. Thus $S(s)$ is, loosely speaking, the amount of power allocated on average to the eigenfunction $e^{2\pi i s.x}$ with frequency s. $S(s)$ must eventually decay sufficiently fast as $|s| \longrightarrow \infty$ so that it is integrable; the rate of this decay of the power spectrum gives important information about the smoothness of the associated stochastic process.

If the covariance function is isotropic (so that it is a function of r, where $r = |\tau|$) then it can be shown that $S(s)$ is a function of $s \triangleq |s|$ only [2] [Adler, 1981, Theorem 2.5.2]. In this case the integrals in (Eq 2.10) can be simplified by changing to spherical polar coordinates and integrating out the angular variables (see e.g. [12] Bracewell, 1986, ch. 12) to obtain :

$$k(\tau) = \frac{2\pi}{r^{-D/2-1}} \int_0^\infty S(s) J_{D/,-1}(2\pi r s) s^{D/2}\, ds \quad , \tag{2.11}$$

$$S(s) = \frac{2\pi}{s^{-D/2-1}} \int_0^\infty k(r) J_{D/2-1}(2\pi r s) r^{D/2}\, dr \quad , \tag{2.12}$$

where $J_{D/2-1}$ is a Bessel function of order $D/2 - 1$. Note that the dependence on the dimensionality D in (Eq 2.11) means that the same isotropic functional form of the spectral density can give rise to different isotropic covariance functions in different dimensions. Similarly, if we start with a particular isotropic covariance function $k(r)$ the form of spectral density will in general depend on D (see, e.g. the Matérn class spectral density

given in (Eq 2.19)) and in fact $k(r)$ may not be valid for all D.

A necessary condition for the spectral density to exist is that $\int r^{D-1}|k(r)|\ dr < \infty$.

We now give some examples of commonly-used isotropic covariance functions. The covariance functions are given in a normalized form where $k(0) = 1$; we can multiply k by a (positive) constant σ_f^2 to get any desired process variance.

Squared Exponential Covariance Function

The squared exponential (SE) covariance function has the form :

$$k_{SE}(r) \quad = \quad \exp(-\frac{r^2}{2l^2}) \ , \tag{2.13}$$

with parameter l defining the characteristic length-scale. This covariance function is infinitely differentiable, which means that the GP with this covariance function has mean square derivatives of all orders, and is thus very smooth. The spectral density of the SE covariance function is $S(s) = (2\pi l^2)^{D/2} \exp(-2\pi l^2 s^2)$.

Stein [1999] [74] argues that such strong smoothness assumptions are unrealistic for modelling many physical processes, and recommends the Matern class (see below). However, the squared exponential isprobably the most widely-used kernel within the kernel machines field.

The SE kernel is infinitely divisible in that $(k(r))^t$ is a valid kernel for all $t > 0$, the effect of raising k to the power of t is simply to rescale l.

We now digress briefly, to show that the squared exponential covariance function can also be obtained by expanding the input x into a feature space defined by Gaussian-shaped basis functions centered densely in x-space. For simplicity of exposition we consider scalar inputs with basis functions :

$$\Phi_{\mathbf{c}}(x) \quad = \quad \exp(-\frac{(x-\mathbf{c})^2}{2l^2}) \quad , \tag{2.14}$$

where $\mathbf{c}$ denotes the centre of the basis function. We recall that with a Gaussian prior on the weights $w \sim \mathcal{N}(0, \sigma_p^2 I)$, this gives rise to a GP with covariance function

$$k(x_p, x_q) = \sigma_p^2 \sum_{\mathbf{c}=1}^{N} \Phi_{\mathbf{c}}(x_p)\Phi_{\mathbf{c}}(x_q) \quad , \tag{2.15}$$

Now, allowing an infinite number of basis functions centered everywhere on an interval (and scaling down the variance of the prior on the weights with the number of basis functions) we obtain the limit :

$$\lim_{N\longrightarrow\infty} \frac{\sigma_p^2}{N} \sum_{\mathbf{c}=1}^{N} \Phi_{\mathbf{c}}(x_p)\Phi_{\mathbf{c}}(x_q) = \sigma_p^2 \int_{\mathbf{c}_{\min}}^{\mathbf{c}_{\max}} \Phi_{\mathbf{c}}(x_p)\Phi_{\mathbf{c}}(x_q)\, d\mathbf{c} \quad , \tag{2.16}$$

Plugging in the Gaussian-shaped basis functions (Eq 2.14) and letting the integration limits go to infinity we obtain :

$$\begin{aligned} k(x_p, x_q) &= \sigma_p^2 \int_{-\infty}^{\infty} \exp(-\frac{(x_p - \mathbf{c})^2}{2l^2}) \exp(-\frac{(x_q - \mathbf{c})^2}{2l^2})\, d\mathbf{c} \\ &= \sqrt{\pi} l \sigma_p^2 \exp(-\frac{(x_p - x_q)^2}{2(\sqrt{2}l)^2}) \end{aligned} \quad , \tag{2.17}$$

which we recognize as a squared exponential covariance function with a times longer length-scale. The derivation is adapted from MacKay [1998] [39]. It is straightforward to generalize this construction to multivariate x.

The Matérn Class of Covariance Functions

The Matérn class of covariance functions is given by :

$$k_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} (\frac{\sqrt{2\nu} r}{l})^{\nu} k_{\nu}(\frac{\sqrt{2\nu} r}{l}) \quad , \tag{2.18}$$

with positive parameters ν and l, where k_ν is a modified Bessel function [Abramowitz and Stegun, 1965, sec. 9.6] [1]. This covariance function has a spectral density in D

dimensions.

$$S(s) = \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu) l^{2\nu}} (\frac{2\nu}{l^2} + 4\pi^2 s^2)^{-(\nu + D/2)} \quad , \tag{2.19}$$

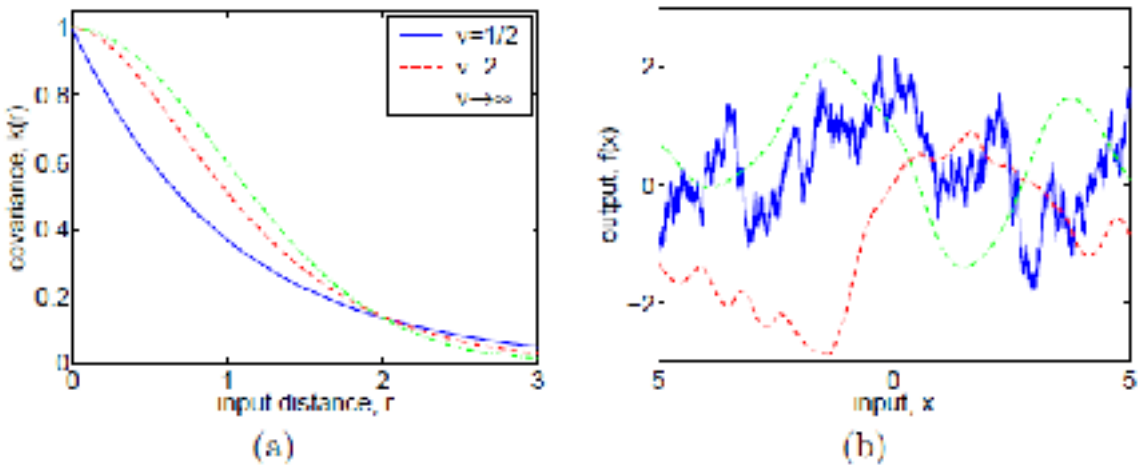


Figure 2.1: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with Matérn covariance functions, (Eq 2.18), for different values of ν, with $l = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.[15]

Note that the scaling is chosen so that for $\nu \longrightarrow \infty$ we obtain the SE covariance function $e^{-r^2/2l^2}$. For the Matérn class the process $f(\mathbf{x})$ is k-times MS differentiable if and only if $\nu > k$. The Matérn covariance functions become especially simple when ν is half-integer : $\nu = p + 1/2$, where p is a non-negative integer.

In this case the covariance function is a product of an exponential and a polynomial of order p, the general expression can be derived from [Abramowitz and Stegun, 1965, eq. 10.2.15] [1], giving

$$k_{\nu=p+1/2}(r) = \exp(-\frac{\sqrt{2\nu}r}{l}) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} (\frac{\sqrt{8\nu}r}{l})^{p-i} \quad , \tag{2.20}$$

It is possible that the most interesting cases for machine learning are $\nu = 3/2$ and

$\nu = 5/2$, for which

$$k_{\nu=3/2}(r) \quad = \quad (1 + \frac{\sqrt{3}r}{l}) \exp(-\frac{\sqrt{3}r}{l}) \ , \tag{2.21}$$

$$k_{\nu=5/2}(r) \quad = \quad (1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}) \exp(-\frac{\sqrt{5}r}{l}) \ , \tag{2.22}$$

since for $\nu = 1/2$ the process becomes very rough (see below), and for $\nu \geq 7/2$, in the absence of explicit prior knowledge about the existence of higher order derivatives, it is probably very hard from finite noisy training examples to distinguish between values of $\nu \geq 7/2$ (or even to distinguish between finite values of ν and $\nu \longrightarrow \infty$, the smooth squared exponential, in this case). For example a value of $\nu = 5/2$ was used in [Cornford et al., 2002] [15].

Ornstein-Uhlenbeck Process and Exponential Covariance Function

The special case obtained by setting $\nu = 1/2$ in the Matérn class gives the exponential covariance function

$$k(r) \quad = \quad \exp(-(r/l)) \quad , \tag{2.23}$$

The corresponding process is MS continuous but not MS differentiable. In $D = 1$ this is the covariance function of the Ornstein-Uhlenbeck (OU) process.

The OU process [Uhlenbeck and Ornstein, 1930] [58] was introduced as a mathematical model of the velocity of a particle undergoing Brownian motion. More generally in $D = 1$ setting $\nu + 1/2 = p$ for integer p gives rise to a particular form of a continuous-time AR(p) Gaussian process. The form of the Matérn covariance function and samples drawn from it for $\nu = 1/2, \nu = 2$ and $\nu \longrightarrow \infty$ are illustrated in (Fig 2.2)

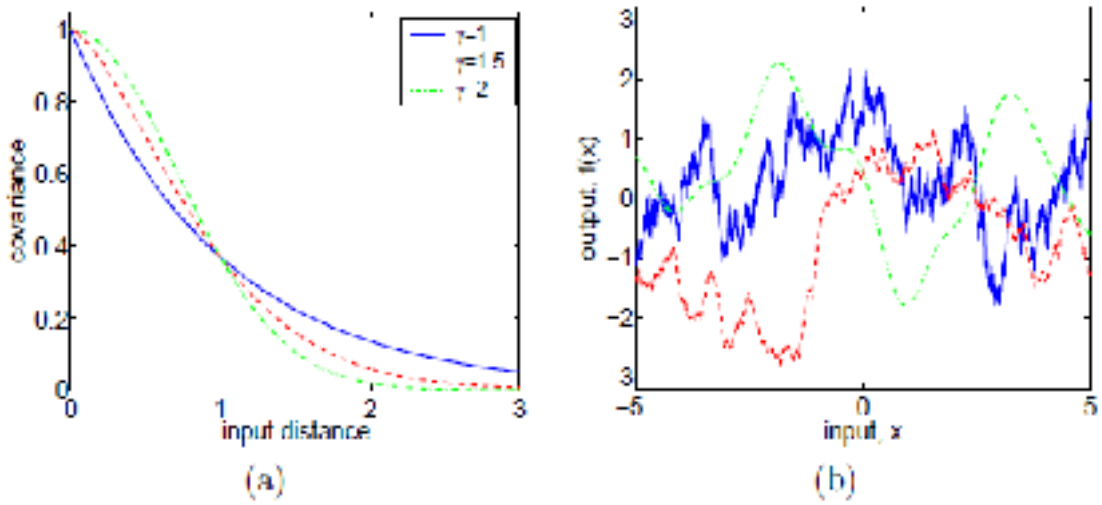


Figure 2.2: Panel (a) covariance functions, and (b) random functions drawn from Gaussian processes with the $\gamma-$exponential covariance function (Eq 2.24), for different values of γ , with $l = 1$. The sample functions are only differentiable when $\gamma = 2$ (the SE case). The sample functions on the right were obtained using a discretization of the $x-$axis of 2000 equally-spaced points. [62]

The $\gamma-$exponential Covariance Function

The $\gamma-$exponential family of covariance functions, which includes both the exponential and squared exponential, is given by

$$k(r) \quad = \quad \exp(-(r/l)^{\gamma}) \quad \text{for} \quad 0 < \gamma \leq 2 \quad , \tag{2.24}$$

Although this function has a similar number of parameters to the Matérn class, it is (as Stein [1999] [74] notes) in a sense less flexible. This is because the corresponding process is not MS differentiable except when $\gamma = 2$ (when it is infinitely MS differentiable).

Rational Quadratic Covariance Function

The rational quadratic (RQ) covariance function :

$$k_{RQ}(r) \quad = \quad (1 + \frac{r^2}{2\alpha l^2})^{-\alpha} \quad , \tag{2.25}$$

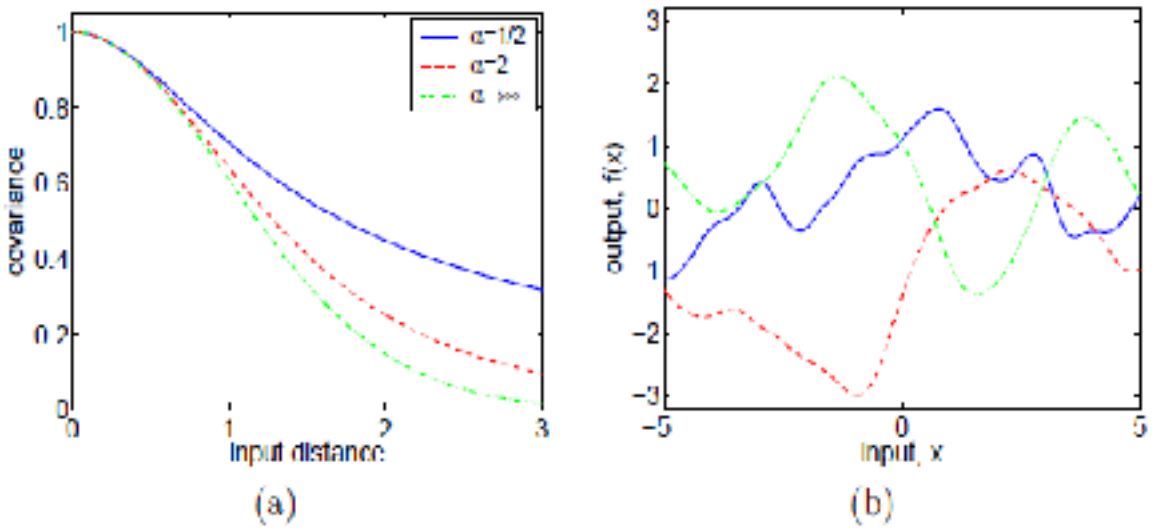


Figure 2.3: Panel (a) covariance functions, and (b) random functions drawn from Gaussian processes with rational quadratic covariance functions, (Eq 2.27), for different values of α with $l = 1$. The sample functions on the right were obtained using a discretization of the $x-$axis of 2000 equally-spaced points. [62]

with $\alpha, l > 0$ can be seen as a scale mixture (an infinite sum) of squared exponential (SE) covariance functions with different characteristic length-scales. Parameterizing now in terms of inverse squared length scales, $\tau = l^{-2}$, and putting a gamma distribution on

$$P(\tau|\alpha,\beta) \quad \propto \quad \tau^{\alpha-1}\exp(-\alpha\tau/\beta) \qquad . \tag{2.26}$$

we can add up the contributions through the following integral

$$\begin{aligned} k_{RQ}(r) \quad &= \quad \int P(\tau|\alpha,\beta)\; k_{SE}(r/\tau)\; d\tau \\ &\propto \quad \int \tau^{\alpha-1}\exp(-\frac{\alpha\tau}{\beta})\exp(-\frac{\tau r^2}{2})\; d\tau \quad \propto \quad (1+\frac{r^2}{2\alpha l^2})^{-\alpha} \end{aligned} \qquad , \tag{2.27}$$

where we have set $\beta^{-1} = l^2$. The rational quadratic is also discussed by Matérn [1960, p. 17] [44] using a slightly different parameterization; in our notation the limit of the RQ covariance for $\alpha \longrightarrow \infty$ is the SE covariance function with characteristic length-scale l (Fig 2.13). (Fig 2.3) illustrates the behaviour for different values of α ; note that the

process is infinitely MS differentiable for every α in contrast to the Matérn covariance function in (Fig 2.1).

The previous example is a special case of kernels which can be written as superpositions of SE kernels with a distribution $P(l)$ of length-scales l ,

$$k(r) \quad = \quad \int \exp(\frac{-r^2}{2l^2}) P(l)\ dl. \qquad , \tag{2.28}$$

This is in fact the most general representation for an sotropic kernel which defines a valid covariance function in any dimension D, see [Stein [74] , 1999, sec. 2.10].[62]

2.5.2 Non-stationary covariance functions

We briefly list a few examples of common nonstationary covariances. The linear covariance produces straight line sample functions, and using it in GP regression is therefore equivalent to doing Bayesian linear regression :

$$k_{Lin}(x, \grave{x}) \quad = \quad \sigma_0^2 + \sigma_1^2 x \grave{x}^T \qquad , \tag{2.29}$$

The periodic covariance can be used to generate periodic random functions $(1D)$:

$$k_{Per}(x, \grave{x}) \quad = \quad \exp\left\{ -\frac{2\sin^2(\frac{x - \grave{x}}{2})}{l^2} \right\} \qquad , \tag{2.30}$$

The Wiener process, or continuous time Brownian motion, is a one-dimensional nonstationary GP

$$k_{Wien}(x, \grave{x}) \quad = \quad \min(x, \grave{x}), \quad x, \grave{x} \geq 0 \qquad , \tag{2.31}$$

A nonstationary neural network covariance function can be constructed as a limit of a particular type of neural network with an infinite number of hidden units [Williams, 1998] [81].

There are many other examples of covariance functions, both stationary and nonstationary. It is also possible to combine covariances in sums, products and convolutions to obtain more flexible and complex processes. See [62] for further details. [71]

Chapter 3

Gaussian Process Classification

Gaussian process classifiers (GPCs) are Bayesian probabilistic kernel classifiers. In GPCs, the probability of belonging to a certain class at an input location is monotonically related to the value of some latent function at that location. Starting from a Gaussian process prior over this latent function, data are used to infer both the posterior over the latent function and the values of hyperparameters to determine various aspects of the function.[35]

In this chapter, we are giving the Bayesian classification with Gaussian process, Laplace approximation for binary GPC, the multi-class Laplace approximation and the expectation propagation method.

3.1 Classification

Supervised classification of remote sensing images has received great attention from the remote sensing community for several decades. For such a purpose, many simple and sophisticated techniques have been considered, such as the Statistical classifier, the k-Nearest Neighbor classifier, the Artificial Neural-Network (NN) classifier, and, more recently, the Kernel-based classifier [21] and [75]. Among the most popular kernelbased classifiers available in the literature, one can find Support Vector Machine (SVM) classifiers [78].

They are based on the margin maximization principle, which aims at providing them with a good generalization capability. SVM classifiers have been used extensively and proved to be successful in dealing with remote sensing data [22] and [23].

Another potentially interesting kernel-based classification approach is the one based on Gaussian processes (GPs) [82] and [62]. In contrast to SVM classifiers, GP classifiers (GPCs) have not yet received sufficient attention from the remote sensing community, despite being theoretically attractive statistical models that permit a fully Bayesian treatment of the considered classification problem. Compared to SVM classifiers, they have the advantage of providing probabilistic outputs rather than discriminant function values. Moreover, they can use evidence for solving the model selection issue in a completely automatic way.

The main idea of GPCs is to assume that the probability of belonging to a class label for an input sample is monotonically related to the value of some latent function at that sample. Such a monotonic relationship is defined according to a so-called squashing function. A GP prior characterized by a covariance matrix embedding a set of hyperparameters is placed on this latent function. Inference is made by integrating over the latent function. Since such an integral is analytically intractable, solutions based on Monte Carlo sampling or analytical approximation methods are adopted.

The two key analytical approximation algorithms are the Laplace and expectation-propagation (EP) algorithms. Both approximate the non-Gaussian joint posterior over the latent variables with a Gaussian one. In the Laplace approximation, the Gaussian model is defined with mean and covariance matrix as the maximum point of the posterior and the negative Hessian matrix at that point, respectively.

The identification of this maximum is carried out according to the iterative Newton method. The EP algorithm is a more sophisticated approximation technique that tries in some way to minimize locally the Kullback–Leibler divergence measure between the true posterior and the approximated one. This is done sequentially through the so-called cavity distribution.

In the prediction phase, the approximate predictive mean and variance for the (approximated) Gaussian posterior over the latent variable of the considered sample are computed first. Then, the class posterior probability for the sample target is derived either analytically or by approximation, depending on the adopted squashing function. The multiclass implementation of GPCs is obtained through an intrinsicmulticlass formulation, which can be complex [28], or simply by decomposition into binary classification problems.[8]

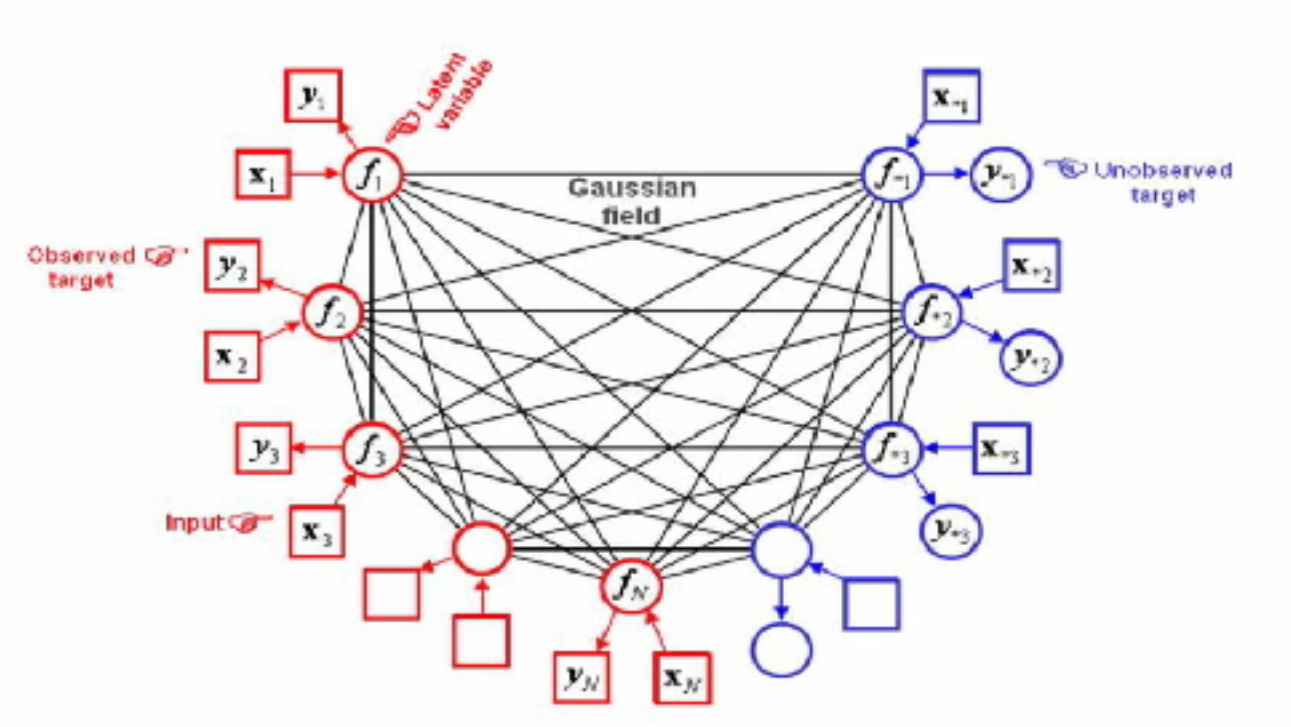


Figure 3.1: Graphical model for GPCs with N training data points and one test data point. [8]

3.2 Baysian classification with Gaussian process

Let us consider a training set $\boldsymbol{D} = (\mathbf{X}, \mathbf{y})$ consisting of a matrix of training data $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2, ...\ \mathbf{x}_N]^T$ where N is the number of samples and $\mathbf{y} = [y_1\ y_2\ ...\ y_N]^T$ is the corresponding target vector. To each vector $\mathbf{x}_i \in \mathbf{R}^D (i = 1, 2, ..., N)$, we associate a target $y_i \in \{-1, +1\}$. Given this training set $\boldsymbol{D}$, we aim at predicting the label y_* of a new test sample $\mathbf{x}_*$ by computing the output probability $P(y_*|\boldsymbol{D}, \mathbf{x}_*)$.

In GPC, the probability of belonging to a class label $y_i = +1$ for an input sample $\mathbf{x}_i$ is monotonically related to the value of some latent function f_i. Such monotonic relationship is defined according to a squashing function, which can take several forms. In this work, we will consider the probit function :

$$P(y_i = +1|f_i) \quad = \quad \phi(y_i f_i) \quad , \tag{3.1}$$

where ϕ is the Gaussian cumulative distribution function :

$$\phi(z) \quad = \quad \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})\ dx \quad , \tag{3.2}$$

A Gaussian process prior (GP) characterized by a zero mean and a covariance matrix embedding a set of hyperparameters Θ is placed on this latent function. The prediction of the output probability for the test sample $\mathbf{x}_*$ is obtained by integrating over the latent function f_* as follows:

$$P(y_* = +1|\boldsymbol{D}, \mathbf{x}_*, \Theta) \quad = \quad \int P(y_*|f_*, \Theta)\ P(f_*|\boldsymbol{D}, \mathbf{x}_*, \Theta)\ df_* \qquad , \tag{3.3}$$

The second part of the integral (Eq 3.3) represents the distribution of the latent variable corresponding to the test sample $\mathbf{x}_*$.

It is obtained by further integrating over $\mathbf{f} = [f_1\ f_2\ ...\ f_N]$:

$$P(f_*|\boldsymbol{D}, \mathbf{x}_*, \Theta) \quad = \quad \int P(f_*|\mathbf{X}, \mathbf{x}_*, \mathbf{f}, \Theta)\ P(\mathbf{f}|\boldsymbol{D}, \Theta)\ d\mathbf{f} \qquad , \tag{3.4}$$

where, $P(\mathbf{f}|\boldsymbol{D}, \Theta)$ is the posterior over the latent variables :

$$P(\mathbf{f}|\boldsymbol{D}, \Theta) \quad = \quad P(\mathbf{y}|\mathbf{f}, \Theta)\ P(\mathbf{f}|\mathbf{X}, \Theta)/P(\mathbf{y}|\mathbf{X}, \Theta) \qquad , \tag{3.5}$$

$P(\mathbf{y}|\mathbf{f}, \Theta)$ is the probability of each observed class label given the latent function value.

A possible form is the one adopted in (Eq 3.1). $P(\mathbf{y}|\mathbf{X}, \Theta)$ is the marginal likelihood (evidence), and $P(\mathbf{f}|\mathbf{X}, \Theta)$ is the GP prior over the latent functions, i.e.

$$P(\mathbf{f}|\mathbf{X},\Theta) \quad = \quad \frac{1}{(2\pi)^{N/2}\,|\mathbf{K}|^{\frac{1}{2}}}\exp\{-\tfrac{1}{2}\mathbf{f}\ \mathbf{K}^{-1}\mathbf{f}\} \qquad , \tag{3.6}$$

where each term of the covariance function $\mathbf{K}$ is a function of $\mathbf{x}_i$ and $\mathbf{x}_j$. A popular covariance function is the squared exponential (or Gaussian RBF), i.e.

$$k(x_i^{(m)}, x_j^{(m)}) \quad = \quad \sigma^2 \exp\left\{ -\frac{\sum_{m=1}^{d}(x_i^{(m)} - x_j^{(m)})^2}{2l^2} \right\} \qquad , \tag{3.7}$$

where σ is the variance and l is the length scale. Together they form the hyperparameter vector Θ, i.e. $\Theta = [l\ \sigma]$.

Since the integrals in (Eq 3.3) and (Eq 3.4) are not analytically tractable due to the nonlinearity in the likelihood terms, analytical approximation or Monte Carlo methods have to be adopted. In next section, we describe two well known analytical approximation methods, namely the Laplace and the Expectation Propagation (EP) algorithms.[7]

3.3 Laplace approximation for binary GP classifier

Laplace's method utilizes a Gaussian approximation $q(\mathbf{f}|\mathbf{X},\mathbf{y})$ to the posterior $P(\mathbf{f}|\mathbf{X},\mathbf{y})$ in the integral (Eq 3.4). Doing a second order Taylor expansion of $\ln P(\mathbf{f}|\mathbf{X},\mathbf{y})$ around the maximum of the posterior, we obtain a Gaussian approximation :

$$q(\mathbf{f}|\mathbf{X},\mathbf{y}) \quad = \quad \mathcal{N}(\mathbf{f}|\hat{\mathbf{f}}, A^{-1}) \quad \propto \quad \exp(-\tfrac{1}{2}(\mathbf{f}-\hat{\mathbf{f}})^T A(\mathbf{f}-\hat{\mathbf{f}})) \qquad , \tag{3.8}$$

where $\hat{\mathbf{f}} = argmax_{\mathbf{f}}\ P(\mathbf{f}|\mathbf{X},\mathbf{y})$ and $A = -\nabla\nabla \ln p(\mathbf{f}|\mathbf{X},\mathbf{y})|_{\mathbf{f}=\hat{\mathbf{f}}}$ is the Hessian of the negative log posterior at that point.

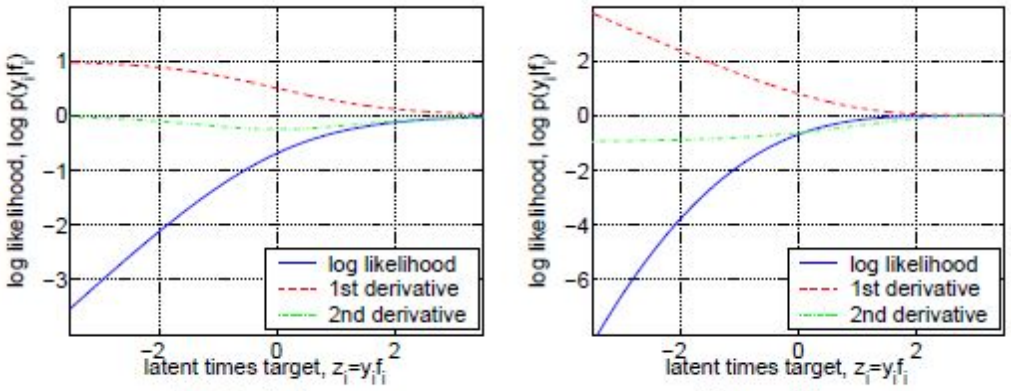


Figure 3.2: likelihood functions are fairly similar, the main qualitative difference being that for large negative arguments the log logistic behaves linearly whereas the log cumulative Gaussian has a quadratic penalty. Both likelihoods are log concave. [62]

3.3.1 Posterior

By Bayes' rule the posterior over the latent variables is given by :

$$P(\mathbf{f}|\mathbf{X},\mathbf{y}) \quad = \quad P(\mathbf{y}|\mathbf{f})\ P(\mathbf{f}|\mathbf{X})/P(\mathbf{y}|\mathbf{X}) \quad , \tag{3.9}$$

but as $P(\mathbf{y}|\mathbf{X})$ is independent of $\mathbf{f}$, we need only consider the un-normalized posterior when maximizing w.r.t. $\mathbf{f}$. Taking the logarithm and introducing expression $\ln P(\mathbf{f}|\mathbf{X}) = -\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\ln|\mathbf{K}| - \frac{N}{2}\ln 2\pi$ for the GP prior gives

$$\begin{aligned} \psi(\mathbf{f}) &\triangleq \ln P(\mathbf{y}|\mathbf{f}) + \ln P(\mathbf{f}|\mathbf{X}) \\ &= \ln P(\mathbf{y}|\mathbf{f}) - \tfrac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} - \tfrac{1}{2}\ln|\mathbf{K}| - \frac{N}{2}\ln 2\pi \end{aligned} \quad , \tag{3.10}$$

Differentiating (Eq 3.10) w.r.t. $\mathbf{f}$ we obtain :

$$\nabla\psi(\mathbf{f}) \quad = \quad \nabla \ln P(\mathbf{y}|\mathbf{f}) - \mathbf{K}^{-1}\mathbf{f} \quad , \tag{3.11}$$

$$\nabla\nabla\psi(\mathbf{f}) \quad = \quad \nabla\nabla \ln P(\mathbf{y}|\mathbf{f}) - \mathbf{K}^{-1} \quad = \quad -W - \mathbf{K}^{-1} \qquad , \tag{3.12}$$

where $W \triangleq -\nabla\nabla \ln P(\mathbf{y}|\mathbf{f})$ is diagonal, since the likelihood factorizes over cases (the distribution for y_i depends only on f_i, not on $f_{j\neq i}$). Note, that if the likelihood $P(\mathbf{y}|\mathbf{f})$ is log concave, the diagonal elements of W are non-negative, and the Hessian in (Eq 3.12) is negative definite, so that $\psi(\mathbf{f})$ is concave and has a unique maximum.

There are many possible functional forms of the likelihood, which gives the target class probability as a function of the latent variable f. Two commonly used likelihood functions are the logistic, and the cumulative Gaussian, (see Fig 3.2). The expressions for the log likelihood for these likelihood functions and their first and second derivatives w.r.t. the latent variable are given in the following table :

Table 3.1: The expressions for the log likelihood

$\ln P(y_i\|f_i)$	$\frac{\partial}{\partial f_i} \ln P(y_i\|f_i)$	$\frac{\partial^2}{\partial f_i^2} \ln P(y_i\|f_i)$
$-\ln(1+\exp(-y_i f_i))$	$t_i - \pi_i$	-$\pi_i(1-\pi_i)$
$\ln \Phi(y_i f_i)$	$\frac{y_i \mathcal{N}(f_i)}{\Phi(y_i f_i)}$	$\frac{-\mathcal{N}(f_i)^2}{\Phi(y_i f_i)^2} \frac{y_i f_i \mathcal{N}(f_i)}{\Phi(y_i f_i)}$

where we have defined $\pi_i = P(y_i = 1|f_i)$ and $t = (\mathbf{y}+1)/2$. At the maximum of $\psi(\mathbf{f})$ we have

$$\nabla\psi(\mathbf{f}) \quad = \quad 0 \quad \Longrightarrow \quad \hat{\mathbf{f}} \quad = \quad \mathbf{K}\nabla \ln P(\mathbf{y}|\hat{\mathbf{f}}) \qquad , \tag{3.13}$$

as a self-consistent equation for $\hat{\mathbf{f}}$ (but since $\nabla \ln P(\mathbf{y}|\hat{\mathbf{f}})$ is a non-linear function of $\hat{\mathbf{f}}$, (Eq 3.13) can not be solved directly). To find the maximum of ψ we use Newton's method, with the iteration

$$\begin{aligned} \mathbf{f}^{new} &= \mathbf{f} - (\nabla\nabla\psi)^{-1}\nabla\psi &&= \mathbf{f} + (\mathbf{K}^{-1} + W)^{-1}(\nabla \ln P(\mathbf{y}|\mathbf{f})) - \mathbf{K}^{-1}\mathbf{f} \\ &&&= (\mathbf{K}^{-1} + W)^{-1}(W\mathbf{f} + \nabla \ln P(\mathbf{y}|\mathbf{f})) \end{aligned} \quad , \tag{3.14}$$

To gain more intuition about this update, let us consider what happens to datapoints that are well-explained under $\mathbf{f}$ so that $\partial \ln P(y_i|f_i)/\partial f_i$ and W_{ii} are close to zero for these points. As an approximation, break $\mathbf{f}$ into two subvectors, $\mathbf{f}_1$ that corresponds to points that are not well-explained, and $\mathbf{f}_2$ to those that are. Then it is easy to show that

$$\mathbf{f}_1^{new} = \mathbf{K}_{11}(I_{11} + W_{11}\mathbf{K}_{11})^{-1}(W_{11}\mathbf{f}_1 + \nabla \ln P(\mathbf{y}_1|f_1)) \quad , \tag{3.15}$$

$$\mathbf{f}_2^{new} = \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{f}_1^{new} \quad , \tag{3.16}$$

where $\mathbf{K}_{21}$ denotes the $n_2 \times n_1$ block of $\mathbf{K}$ containing the covariance between the two groups of points, etc. This means that $\mathbf{f}_1^{new}$ is computed by ignoring entirely the well-explained points, and $\mathbf{f}_2^{new}$ is predicted from $\mathbf{f}_1^{new}$ using the usual GP prediction methods (i.e. treating these points like test points). Of course, if the predictions of $\mathbf{f}_2^{new}$ fail to match the targets correctly they would cease to be well-explained and so be updated on the next iteration.

Having found the maximum posterior $\hat{\mathbf{f}}$, we can now specify the Laplace approximation to the posterior as a Gaussian with mean $\hat{\mathbf{f}}$ and covariance matrix given by the negative inverse Hessian of ψ from (Eq 3.12) .

$$q(\mathbf{f}|\mathbf{X}, \mathbf{y}) = \mathcal{N}(\hat{\mathbf{f}}, (\mathbf{K}^{-1} + W)^{-1}) \quad , \tag{3.17}$$

One problem with the Laplace approximation is that it is essentially uncontrolled, in that the Hessian (evaluated at $\hat{\mathbf{f}}$) may give a poor approximation to the true shape of the

posterior. The peak could be much broader or narrower than the Hessian indicates, or it could be a skew peak, while the Laplace approximation assumes it has elliptical contours.

3.3.2 Predictions

The posterior mean for f_* under the Laplace approximation can be expressed by combining the GP predictive mean $\bar{f}_* = k_*^T(\mathbf{K} + \sigma_n^2 I)^{-1}\mathbf{y}$ with (Eq 3.13) into

$$E_q[f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*] \quad = \quad k(\mathbf{x}_*)^T\mathbf{K}^{-1}\hat{\mathbf{f}} \quad = \quad k(\mathbf{x}_*)^T\nabla \ln P(\mathbf{y}|\hat{\mathbf{f}}) \qquad , \tag{3.18}$$

Compare this with the exact mean, given by Opper and Winther [2000] [**?**] as

$$\begin{aligned} E_p[\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*] &= \int E[\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{x}_*]P(\mathbf{f}|\mathbf{X}, \mathbf{y})d\mathbf{f} \\ &= \int k(\mathbf{x}_*)^T\mathbf{K}^{-1}\mathbf{f}\ P(\mathbf{f}|\mathbf{X}, \mathbf{y})d\mathbf{f} \quad = \quad k(\mathbf{x}_*)^T\mathbf{K}^{-1}E[\mathbf{f}|\mathbf{X}, \mathbf{y}] \end{aligned} \quad , \tag{3.19}$$

where we have used the fact that for a GP $E[f_*|\mathbf{f}, \mathbf{X}, \mathbf{x}_*] = k(\mathbf{x}_*)^T\mathbf{K}^{-1}\mathbf{f}$ and have let $E[\mathbf{f}|\mathbf{X}, \mathbf{y}]$ denote the posterior mean of $\mathbf{f}$ given $\mathbf{X}$ and $\mathbf{y}$. Notice the similarity between the middle expression of (Eq 3.18)) and (Eq 3.19), where the exact (intractable) average $E[\mathbf{f}|\mathbf{X}, \mathbf{y}]$ has been replaced with the modal value $\hat{\mathbf{f}} = E_q[\mathbf{f}|\mathbf{X}, \mathbf{y}]$.

A simple observation from (Eq 3.18) is that positive training examples will give rise to a positive coefficient for their kernel function (as $\nabla_i \ln P(y_i|f_i) > 0$ in this case), while negative examples will give rise to a negative coefficient; this is analogous to the solution to the support vector machine.

Also note that training points which have $\nabla_i \ln P(y_i|f_i) \simeq 0$ (i.e. that are well-explained under $\hat{\mathbf{f}}$) do not contribute strongly to predictions at novel test points; this is similar to the behaviour of non-Support Vectors in the Support Vector Machine.

We can also compute $V_q[f_*|\mathbf{X},\mathbf{y}]$, the variance of $f_*|\mathbf{X},\mathbf{y}$ under the Gaussian approximation. This comprises of two terms, i.e.

$$\begin{aligned} V_q[f_*|\mathbf{X},\mathbf{y},\mathbf{x}_*] &= E_{P(f_*|\mathbf{X},\mathbf{x}_*,\mathbf{f})}[(f_* - E[f_*|\mathbf{X},\mathbf{x}_*,\mathbf{f}])^2] \\ &+ E_{q(\mathbf{f}|\mathbf{X},\mathbf{y})}[(E[f_*|\mathbf{X},\mathbf{x}_*,\mathbf{f}] - E[f_*|\mathbf{X},\mathbf{y},\mathbf{x}_*])^2] \end{aligned} \quad , \tag{3.20}$$

The first term is due to the variance of f_* if we condition on a particular value of $\mathbf{f}$, and is given by $k(\mathbf{x}_*,\mathbf{x}_*) - k(\mathbf{x}_*)^T\mathbf{K}^{-1}k(\mathbf{x}_*)$, cf. eq. (2.19).

The second term in (Eq 3.20) is due to the fact that $E[f_*|\mathbf{X},\mathbf{x}_*,\mathbf{f}] = k(\mathbf{x}_*)^T\mathbf{K}^{-1}\mathbf{f}$ depends on $\mathbf{f}$ and thus there is an additional term of $k(\mathbf{x}_*)^T\mathbf{K}^{-1}cov(\mathbf{f}|\mathbf{X},\mathbf{y})\mathbf{K}^{-1}k(\mathbf{x}_*)$.

Under the Gaussian approximation $cov(\mathbf{f}|\mathbf{X},\mathbf{y}) = (\mathbf{K}^{-1}+W)^{-1}$, and thus

$$\begin{aligned} V_p[f_*|\mathbf{X},\mathbf{y},\mathbf{x}_*] &= k(\mathbf{x}_*,\mathbf{x}_*) - k_*^T\mathbf{K}^{-1}k_* \\ &+k_*^T\mathbf{K}^{-1}(\mathbf{K}^{-1}+W)^{-1}\mathbf{K}^{-1}k_*k(\mathbf{x}_*,\mathbf{x}_*) - k_*^T(\mathbf{K}+W^{-1})k_* \end{aligned} \quad , \tag{3.21}$$

where the last line is obtained using the matrix inversion lemma eq. (A.9).

Given the mean and variance of f_* , we make predictions by computing

$$\bar{\pi}_* \simeq E_q[\pi_*|\mathbf{X},\mathbf{y},\mathbf{x}_*] = \int \sigma(f_*)q(f_*|\mathbf{X},\mathbf{y},\mathbf{x}_*)\, d\mathbf{f}_* \quad , \tag{3.22}$$

where $q(f_*|\mathbf{X},\mathbf{y},\mathbf{x}_*)$ is Gaussian with mean and variance given by (Eq 3.18) and (Eq 3.21) respectively. Notice that because of the non-linear form of the sigmoid the predictive probability from (Eq 3.22) is different from the sigmoid of the expectation of $\mathbf{f}$:

$$\hat{\pi}_* = \sigma(E_q[f_*|\mathbf{y}]) \quad , \tag{3.23}$$

We will call the latter the MAP prediction to distinguish it from the averaged predictions from (Eq 3.22).

3.3.3 Marginal likelihood

It will also be useful to compute the Laplace approximation of the marginal likelihood $P(\mathbf{y}|\mathbf{X})$. We have

$$P(\mathbf{y}|\mathbf{X}) \quad = \quad \int P(\mathbf{y}|\mathbf{f})P(\mathbf{f}|\mathbf{X})d\mathbf{f} \quad = \quad \int \exp(\psi(\mathbf{f}))d\mathbf{f} \qquad , \tag{3.24}$$

Using a Taylor expansion of $\psi(\mathbf{f})$ locally around $\hat{\mathbf{f}}$ we obtain

$$\psi(\mathbf{f}) \quad \simeq \quad \psi(\hat{\mathbf{f}}) - \tfrac{1}{2}(\mathbf{f}-\hat{\mathbf{f}})^T A(\mathbf{f}-\hat{\mathbf{f}}) \qquad , \tag{3.25}$$

and thus an approximation $q(\mathbf{y}|\mathbf{X})$ to the marginal likelihood as

$$P(\mathbf{y}|\mathbf{X}) \quad \simeq \quad q(\mathbf{y}|\mathbf{X}) \quad = \quad \exp(\psi(\hat{\mathbf{f}}))\int \exp\left\{-\tfrac{1}{2}(\mathbf{f}-\hat{\mathbf{f}})^T A(\mathbf{f}-\hat{\mathbf{f}})\right\} \qquad , \tag{3.26}$$

This Gaussian integral can be evaluated analytically to obtain an approximation to the log marginal likelihood

$$\ln q(\mathbf{y}|\mathbf{X},\Omega) \quad = \quad -\frac{1}{2}\hat{\mathbf{f}}^T A\hat{\mathbf{f}} + \ln P(\mathbf{y}|\hat{\mathbf{f}}) - \frac{1}{2}\ln|B| \qquad , \tag{3.27}$$

where $|B| = |\mathbf{K}|\,.\,|\mathbf{K}^{-1}+W| = \left|I_n + W^{\frac{1}{2}}\mathbf{K}W^{\frac{1}{2}}\right|$, and Ω is a vector of hyperparameters of the covariance function (which have previously been suppressed from the notation for brevity).[62]

3.4 Multi-class Laplace approximation

Our presentation follows Williams and Barber [1998] [82]. We first introduce the vector of latent function values at all N training points and for all C classes

$$\mathbf{f} \quad = \quad (f_1^1, ..., f_N^1, f_1^2, ..., f_N^2, ..., f_1^C, ..., f_N^C)^T \qquad , \tag{3.28}$$

Thus $\mathbf{f}$ has length CN. In the following we will generally refer to quantities pertaining to a particular class with superscript c, and a particular case by subscript i (as usual); thus e.g. the vector of C latents for a particular case is $\mathbf{f}_i$. However, as an exception, vectors or matrices formed from the covariance function for class c will have a subscript c. The prior over $\mathbf{f}$ has the form $\mathbf{f} \sim \mathcal{N}(0, \mathbf{K})$. As we have assumed that the C latent processes are uncorrelated, the covariance matrix $\mathbf{K}$ is block diagonal in the matrices $\mathbf{K}_1, ..., \mathbf{K}_C$.

Each individual matrix $\mathbf{K}_c$ expresses the correlations of the latent function values within the class c. Note that the covariance functions pertaining to the different classes can be different.

Let $\mathbf{y}$ be a vector of the same length as $\mathbf{f}$ which for each $i = 1, ..., N$ has an entry of 1 for the class which is the label for example i and 0 for the other $C - 1$ entries.

Let π_i^c denote output of the softmax at training point i, i.e.

$$P(y_i^c|\mathbf{f}_i) \quad = \quad \pi_i^c \quad = \quad \frac{\exp(f_i^c)}{\sum_{\grave{c}} \exp(f_i^{\grave{c}})} \quad , \tag{3.29}$$

Then π is a vector of the same length as $\mathbf{f}$ with entries π_i^c . The multi-class analogue of (Eq 3.10) is the log of the un-normalized posterior

$$\begin{aligned} \psi(\mathbf{f}) \quad \triangleq \quad & -\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} + \mathbf{y}^T\mathbf{f} - \sum_{i=1}^{N} \log(\sum_{c=1}^{C} \exp f_i^c) \\ & -\frac{1}{2}\log|\mathbf{K}| - \frac{CN}{2}\log 2\pi \end{aligned} \quad , \tag{3.30}$$

As in the binary case we seek the MAP value $\hat{\mathbf{f}}$ of $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$. By differentiating (Eq 3.30) w.r.t. $\mathbf{f}$ we obtain

$$\nabla\psi \quad = \quad -\mathbf{K}^{-1}\mathbf{f} + \mathbf{y} - \pi \quad , \tag{3.31}$$

Thus at the maximum we have $\hat{\mathbf{f}} = \mathbf{K}(\mathbf{y} - \hat{\pi})$. Differentiating again, and using

$$-\frac{\partial^2}{\partial f_i^c \partial f_i^{\grave{c}}} \ln \sum_j \exp(f_i^j) \quad = \quad \pi_i^c \delta_{c\grave{c}} + \pi_i^c \pi_i^{\grave{c}} \quad , \tag{3.32}$$

we obtain

$$\nabla\nabla\psi \quad = \quad -\mathbf{K}^{-1} - W \qquad \text{where} \qquad W \quad \triangleq \quad diag(\pi) - \Pi\Pi^T$$

where Π is a $CN \times N$ matrix obtained by stacking vertically the diagonal matrices $diag(\pi^c)$, and π^c is the subvector of π pertaining to class c. As in the binary case notice that $-\nabla\nabla\psi$ is positive definite, thus $\psi(\mathbf{f})$ is concave and the maximum is unique.

As in the binary case we use Newton's method to search for the mode of ψ, giving

$$\mathbf{f}^{new} \quad = \quad (\mathbf{K}^{-1} + W)^{-1}(W\mathbf{f} + \mathbf{y} - \pi) \quad , \tag{3.33}$$

This update if coded naïvely would take $O(C^3N^3)$ as matrices of size CN have to be inverted.

The Laplace approximation gives us a Gaussian approximation $q(\mathbf{f}|\mathbf{X}, \mathbf{y})$ to the posterior $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$. To make predictions at a test point x_* we need to com- predictive pute the posterior distribution $q(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ where $\mathbf{f}(x_*) \triangleq \mathbf{f}_* = (\mathbf{f}_*^1, ..., \mathbf{f}_*^C)^T$
In general we have

$$q(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) \quad = \quad \int P(\mathbf{f}_*|\mathbf{X}, \mathbf{x}_*, \mathbf{f}) q(\mathbf{f}|\mathbf{X}, \mathbf{y}) \; d\mathbf{f} \quad , \tag{3.34}$$

As $P(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ and $q(\mathbf{f}|\mathbf{X}, \mathbf{y})$ are both Gaussian, $q(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ will also be Gaussian and we need only compute its mean and covariance. The predictive mean for class c is given by

$$E_q[\mathbf{f}^c(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}, \mathbf{x}_*] \quad = \quad k_c(\mathbf{x}_*)^T \mathbf{K}_c^{-1} \hat{\mathbf{f}}^c \quad = \quad k_c(\mathbf{x}_*)^T (\mathbf{y}^c - \hat{\pi}^c) \quad , \tag{3.35}$$

where $k_c(\mathbf{x}_*)$ is the vector of covariances between the test point and each of the training points for the cth covariance function, and $\hat{\mathbf{f}}^c$ is the subvector of $\hat{\mathbf{f}}$ pertaining to class c. The last equality comes from using (Eq 3.31) at the maximum $\hat{\mathbf{f}}$. Note the close correspondence to (Eq 3.18). This can be put into a vector form $E_q[\mathbf{f}_*|\mathbf{y}] = Q_*^T(\mathbf{y} - \hat{\pi})$ by defining the $CN \times C$ matrix

$$Q_* = \begin{Bmatrix} k_1(\mathbf{x}_*) & 0 & \cdots & 0 \\ 0 & k_2(\mathbf{x}_*) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_C(\mathbf{x}_*) \end{Bmatrix} .$$

Using a similar argument to (Eq 3.20) we obtain :

$$\begin{aligned} cov_q(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) &= \Sigma + Q_*^T \mathbf{K}^{-1}(\mathbf{K}^{-1} + W)^{-1}\mathbf{K}^{-1}Q_* \\ &= diag(k(\mathbf{x}_*, \mathbf{x}_*)) - Q_*^T(\mathbf{K} + W^{-1})^{-1}Q_* \end{aligned} , \tag{3.36}$$

where $\mathbf{\Sigma}$ is a diagonal $C \times C$ matrix with $\mathbf{\Sigma}_{cc} = k_c(\mathbf{x}_*, \mathbf{x}_*) - k_c^T(\mathbf{x}_*)K_c^{-1}k_c(\mathbf{x}_*)$, and $k(\mathbf{x}_*, \mathbf{x}_*)$ is a vector of covariances, whose c'th element is $k_c(\mathbf{x}_*, \mathbf{x}_*)$.

We now need to consider the predictive distribution $q(\pi_*|\mathbf{y})$ which is obtained by softmaxing the Gaussian $q(\mathbf{f}_*|\mathbf{y})$. In the binary case we saw that the predicted classification could be obtained by thresholding the mean value of the Gaussian. In the multi-class case one does need to take the variability around the mean into account as it can affect the overall classification.

The Laplace approximation to the marginal likelihood can be obtained in the same way as for the binary case, yielding

$$\begin{aligned} \ln P(\mathbf{y}|\mathbf{X},\Omega) &\simeq \ln q(\mathbf{y}|\mathbf{X},\Omega) \\ &= -\frac{1}{2}\hat{\mathbf{f}}^T A\hat{\mathbf{f}} + \mathbf{y}^T\hat{\mathbf{f}} - \sum_{i=1}^{N}\ln(\sum_{c=1}^{C}\exp\hat{\mathbf{f}}_i^c) - \frac{1}{2}\ln\left|I_{CN+}W^{\frac{1}{2}}\mathbf{K}W^{\frac{1}{2}}\right| \quad , \end{aligned} \tag{3.37}$$

As for the inversion of $\mathbf{K}^{-1}+W$, the determinant term can be computed efficiently by exploiting the structure of W.[62]

3.5 Expectation propagation

Minka (2001a) [47] proposed the Expectation Propagation (EP) method which can be applied to Gaussian process models.

EP finds a Gaussian approximation $q(\mathbf{f}|\boldsymbol{D},\,\Omega,\,\psi) = \mathcal{N}(\mathbf{f}|\mathbf{m},\,A)$ to the posterior $P(\mathbf{f}|\boldsymbol{D},\,\Omega,\,\psi)$ by moment matching of approximate marginal distributions. The starting point is to impose a factorising structure:

$$\begin{aligned} P(\mathbf{f}|\boldsymbol{D},\Omega,\psi) &= \frac{\mathcal{N}(\mathbf{f}|0,\mathbf{K})}{P(\boldsymbol{D}|\Omega,\psi)}\prod_{i=1}^{m}P(y_i|f_i,\Omega) \\ &\simeq \frac{\mathcal{N}(\mathbf{f}|0,\mathbf{K})}{q(\boldsymbol{D}|\Omega,\psi)}\prod_{i=1}^{m}t(f_i,\mu_i,\sigma_i^2,Z_i) \quad = \quad q(\mathbf{f}|\boldsymbol{D},\Omega,\psi) \end{aligned} \tag{3.38}$$

resembling the structure of the prior times the factorising likelihood (3.4) where the terms

$$t(f_i,\mu_i,\sigma_i^2,Z_i) \quad = \quad Z_i\mathcal{N}(f_i|\mu_i,\sigma_i^2) \quad , \tag{3.39}$$

are called site functions. Note that the site functions are approximating the likelihood (which normalises over observations y_i), with a Gaussian in f_i, so we cannot expect the site functions to normalise, hence the explicit term Z_i is necessary. For notational convenience we hide the site parameters μ_i, σ_i^2 and Z_i and write $t(f_i)$ instead. From (Eq 3.39) the Gaussian approximation $q(\mathbf{f}|\boldsymbol{D},\Omega,\psi)$ as given by (Eq 3.38) has mean and covariance :

$$\mathbf{m} \quad = \quad A\boldsymbol{\Sigma}^{-1}\mu \qquad \text{and} \qquad A \quad = \quad (K^{-1} + \boldsymbol{\Sigma}^{-1})^{-1} \quad , \tag{3.40}$$

where $\mu = [\mu_1, ..., \mu_m] >$ and $\boldsymbol{\Sigma} = diag(\sigma_1^2, ..., \sigma_m^2)$ collect site function parameters.

The EP algorithm iteratively visits each site function in turn, and adjusts the site parameters to match moments of an approximation to the marginal distributions of the posterior. The kth non-central moment of f_i under the posterior is :

$$\begin{aligned} E[f_i^k] \quad &= \quad \frac{1}{P(\boldsymbol{D}|\Omega,\psi)} \int f_i^k \; P(\mathbf{y}|\mathbf{f},\Omega) \; P(\mathbf{f}|\mathbf{X},\psi) \; d\mathbf{f} \\ &= \quad \frac{1}{P(\boldsymbol{D}|\Omega,\psi)} \int f_i^k \; P(y_i|f_i,\Omega) \; P_{\setminus i}(f_i) \; df_i \end{aligned} \quad , \tag{3.41}$$

where

$$P_{\setminus i}(f_i) \quad = \quad \int \prod_{j \neq i} P(y_j|f_j,\Omega) P(\mathbf{f}|\mathbf{X},\psi) \; d\mathbf{f}^{\setminus i} \quad , \tag{3.42}$$

is called the cavity distribution and $\mathbf{f}^{\setminus i}$ denotes $\mathbf{f}$ without f_i. The marginalisation required to compute the exact cavity distribution is intractable. The key step in the EP algorithm is to replace the intractable exact cavity distribution with a tractable approximation based on the site functions :

$$P_{\setminus i}(f_i) \quad \simeq \quad q_{\setminus i}(f_i) \quad = \quad \int \prod_{j \neq i} t(f_j) P(\mathbf{f}|\mathbf{X},\psi) \; d\mathbf{f}^{\setminus i} \quad , \tag{3.43}$$

The approximate cavity function comes in the form of an unnormalised Gaussian $q_{\setminus i}(f_i) \propto \mathcal{N}(f_i/\mu_{\setminus i}, \sigma^2_{\setminus i})$. Multiplying both sides by $t(f_i)$:

$$q_{\setminus i}(f_i) t(f_i) \quad = \quad \int \mathcal{N}(\mathbf{f}|0,K) \prod_{j=1}^{m} t(\mathbf{f}_j) \; df^{\setminus i} \quad \propto \quad \mathcal{N}(f_i|m_i, A_{ii}) \quad , \tag{3.44}$$

and using basic Gaussian identities we obtain the parameters :

$$\sigma^2_{\setminus i} \quad = \quad ((A_{ii})^{-1} - \sigma{-}^2_i)^{-1} \qquad \text{and} \qquad \mu_{\setminus i} \quad = \quad \sigma^2_{\setminus i}(\frac{m_i}{A_{ii}} - \frac{\mu_i}{\sigma_i^2}) \quad , \tag{3.45}$$

of the approximate cavity function.

The core idea of EP is to adjust the site parameters μ_i, σ_i^2, and Z_i such that the approximate posterior marginal using the exact likelihood approximates as well as possible the posterior marginal based on the site function :

$$q_{\setminus i}(f_i)P(y_j|f_j,\Omega) \quad \simeq \quad q_{\setminus i}(f_i)t(f_i,\mu_i,\sigma_i^2,Z_i) \quad , \tag{3.46}$$

by matching the zeroth, first, and second moments.

Matching of moments minimises Kullback-Leibler divergence KL($P||q$). Although the classical KL argument only applies to the first and second (and higher) moments for normalised distributions, it seems natural also to match zeroth moments.

Therefore, the zeroth, first, and second non-central moment

$$m_k \quad = \quad \int f_i^k \; P(y_i|f_i,\Omega)q_{\setminus i}(f_i) \; df_i \quad = \quad \int f_i^k \; P(y_i|f_i,\Omega) \; \mathcal{N}(f_i|\mu_{\setminus i},\sigma^2_{\setminus i}) \; df_i \quad , \tag{3.47}$$

of the left hand side of (Eq 3.46) have to be computed for $k = 0, 1, 2$. This can be implemented using numerical integration techniques, but if the moments can be computed analytically this is usually computationally advantageous. In this case a generic approach is to use the moment generating function

$$M(\lambda) \quad = \quad \int \exp(\lambda f_i) \; P(y_i|f_i,\Omega) \; \mathcal{N}(f_i|\mu_{\setminus i},\sigma^2_{\setminus i}) \; df_i \quad , \tag{3.48}$$

and differentiating with respect to λ gives the non-central moments :

$$m_0 = M(0) \quad , \quad m_1 = \frac{1}{m_0}\frac{\partial M}{\partial \lambda}\,|_{\lambda=0} \quad \text{and} \quad m_2 = \frac{1}{m_0}\frac{\partial^2 M}{\partial \lambda^2}\,|_{\lambda=0} \quad , \tag{3.49}$$

By equating these moments with the right hand side of (Eq 3.46) the update equations for the site parameters become :

$$\begin{aligned} \sigma_i^2 &= ((m_2 - m_1^2)^{-1} - \sigma_{\setminus i}^{-2})^{-1} \\ \mu_i &= \sigma_i^2(m_1(\sigma_{\setminus i}^{-2} + \sigma_i^{-2}) - \frac{\mu_{\setminus i}}{\sigma_{\setminus i}^2}) \\ Z_i &= m_0\sqrt{2\pi(\sigma_{\setminus i}^2 + \sigma_i^2)}\exp(\frac{(\mu_i - \mu_{\setminus i})^2}{2(\sigma_{\setminus i}^2 + \sigma_i^2)}) \end{aligned} \quad , \tag{3.50}$$

Once the values of μ_i and σ_i^2 are updated, the effect on $\mathbf{m}$ and A has to be computed according to (Eq 3.40), which in practice is done using rank-one updates of A.

The EP algorithm iteratively updates the site parameters until convergence. A formal proof of convergence does not exist but for log-concave likelihood functions, i.e. when the posterior is concave, EP usually converges reliably.

Finally the evidence can be approximated from the normalisation of (Eq 3.38) :

$$\begin{aligned} \ln p(\boldsymbol{D}|\Omega,\psi) \simeq \ln q(\boldsymbol{D}|\Omega,\psi) &= \ln \int \mathcal{N}(\mathbf{f}|0,\,\mathbf{K}) \prod_{i=1}^{m} t(f_i)\; d\mathbf{f} \\ &= \sum_{i=1}^{m} \ln Z_i - \frac{1}{2}\ln|\mathbf{K}+\mathbf{\Sigma}| - \frac{1}{2}\mu^T(\mathbf{K}+\mathbf{\Sigma})^{-1}\mu - \frac{m}{2}\ln(2\pi) \end{aligned} \quad , \tag{3.51}$$

and its derivatives can be computed in order to implement ML-II parameter estimation of Ω and ψ The derivatives and further details on implementing EP for Gaussian process models can be found in Appendix A.3 [79], where also a pseudo-code description is given.

In practical applications the EP approximation shows to work and converge better for some likelihood models than for others. Unimodality of the posterior$-$*log* concavity

of the likelihood—seems to be an important factor. Note that in the update (Eq 3.50) of the site function parameters we ignored the possibility that updates lead to an invalid, non-positive definite covariance matrix A. In those cases one can either skip the update in the hope that a later update will be valid or dampen (soften) the update using a "learning rate" parameter small enough to obtain a positive definite A. However, in general it is not guaranteed that EP converges and often it is a challenging task to implement EP for a particular likelihood avoiding numerical difficulties.[79]

Chapter 4

Spatial Contextual Gaussian Process Classification

In this chapter, we propose a thorough investigation of the GPC effectiveness for classifying multisource and hyperspectral remote sensing images. To this end, we designed several experiments aiming also at testing the sensitivity of GPCs to the number of training samples and to the course of dimensionality. In general, the obtained classification results show clearly that the GPC is given good results.

4.1 Method description

Now, assuming a spatial neighbourhood system of size $N^* \times N^*$, let us consider a training set $\grave{\boldsymbol{D}} = (\grave{\mathbf{X}}, \grave{\mathbf{y}})$ consisting of a matrix of training data $\grave{\mathbf{X}} = [\mathbf{X}\ \mathbf{X}_n^*]$ accompanied with labels $\grave{\mathbf{y}} = [\mathbf{y}\ \mathbf{y}_n^*]$, where $\mathbf{y}_n^*$ and $\mathbf{X}_n^*$ are the spatial neighbors of y_* and $\mathbf{x}_*$ with sizes $\grave{d} \times N_*$ and $N_* = N^* \times N^*$, respectively (see Fig **??**). We aim at determining the label y_* at new test point $\mathbf{x}_*$ by computing the class posterior probability $P(y_*|\grave{\boldsymbol{D}}, \mathbf{x}_*)$.

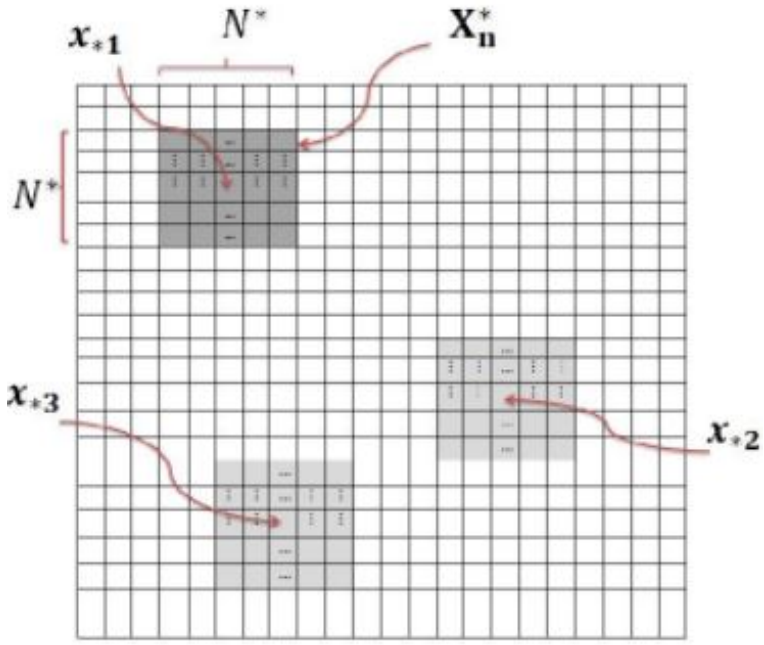


Figure 4.1: Illustration of spatial neighborhood system of size $N^* \times N^*$ centered on samples x_{*1}, x_{*2}, x_{*3}.

The prediction of the output probability for the test sample $\mathbf{x}_*$ is obtained by integrating over the latent function f_* as follows :

$$P(y_* = +1|\grave{\boldsymbol{D}}, \mathbf{x}_*) \quad = \quad \int P(y_*|f_*) P(f_*|\grave{\boldsymbol{D}}, \mathbf{x}_*)\ df_* \qquad , \tag{4.1}$$

$P(y_*|\grave{\boldsymbol{D}}, \mathbf{x}_*)$ is the distribution of the latent variable corresponding to the sample $\mathbf{x}_*$. It is obtained by further integrating over $\grave{\mathbf{f}} = [f_1\ f_2 \dots\ f_N\ f_{(N+1)}\ f_{(N+2)} \dots\ f_{\grave{N}}]$, where $\grave{N} = N + (N_* - 1)$.

$$P(f_*|\grave{\boldsymbol{D}}, x_*) \quad = \quad \int P(f_*|\grave{\mathbf{X}}, \mathbf{x}_*, \grave{\mathbf{f}}) P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}})\ d\grave{\mathbf{f}} \qquad , \tag{4.2}$$

The second part of the integral in (Eq 4.2) represents the posterior over of the latent variables :

$$P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}}) \quad = \quad P(\grave{\mathbf{y}}|\grave{\mathbf{f}}) P(\grave{\mathbf{f}}|\grave{\mathbf{X}}) / P(\grave{\mathbf{y}}|\grave{\mathbf{X}}) \qquad , \tag{4.3}$$

$P(\grave{\mathbf{y}}|\grave{\mathbf{f}})$ is the likelihood function. It can be expressed by using one of the forms of the squashing functions. $P(\grave{\mathbf{y}}|\grave{\mathbf{X}})$ is the marginal likelihood and $P(\grave{\mathbf{f}}|\grave{\mathbf{X}})$ is the GP prior over the latent variables. The GP prior is typically characterized by a zero mean and a covariance matrix embedding a set of hyperparameters, i.e.

$$P(\grave{\mathbf{f}}|\grave{\mathbf{X}}) \quad = \quad \frac{1}{(2\pi)^{\grave{N}/2} \left|\grave{\mathbf{K}}\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\grave{\mathbf{f}}^T \grave{\mathbf{K}}^{-1} \grave{\mathbf{f}}\right\} \qquad , \tag{4.4}$$

where $\grave{\mathbf{K}}$ is the covariance matrix, i.e.,

$$\grave{\mathbf{K}} \quad = \quad \left\{ \begin{array}{cccccc} & & & k(\mathbf{x}_1, \mathbf{x}_{N+1}) & \cdots & k(\mathbf{x}_1, \mathbf{x}_{\grave{N}}) \\ & \mathbf{K} & & \vdots & \ddots & \vdots \\ & & & k(\mathbf{x}_N, \mathbf{x}_{N+1}) & \cdots & k(\mathbf{x}_N, \mathbf{x}_{\grave{N}}) \\ k(\mathbf{x}_{N+1}, \mathbf{x}_1) & \cdots & k(\mathbf{x}_{N+1}, \mathbf{x}_N) & k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) & \cdots & k(\mathbf{x}_{N+1}, \mathbf{x}_{\grave{N}}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_{\grave{N}}, \mathbf{x}_1) & \cdots & k(\mathbf{x}_{\grave{N}}, \mathbf{x}_N) & k(\mathbf{x}_{\grave{N}}, \mathbf{x}_{N+1}) & \cdots & k(\mathbf{x}_{\grave{N}}, \mathbf{x}_{\grave{N}}) \end{array} \right\} . \tag{4.5}$$

The Laplace approximation uses a Gaussian approximation $q(\grave{\mathbf{f}}|\grave{\boldsymbol{D}})$ to the non-Gaussian

posterior in integral of (Eq 4.2). This approximation is based on the second-order Taylor expansion of $\ln P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}})$ around the maximum of the posterior :

$$P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}}) \cong q(\grave{\mathbf{f}}|\grave{\boldsymbol{D}}) = \mathcal{N}(\grave{\mathbf{f}}|\hat{\mathbf{f}}', \grave{A}^{-1}) \propto \exp\left\{-\frac{1}{2}(\grave{\mathbf{f}}-\hat{\mathbf{f}}')^T \grave{A}(\grave{\mathbf{f}}-\hat{\mathbf{f}}')\right\} \quad , \quad (4.6)$$

where $\hat{\mathbf{f}}'$ and $\grave{A}$ are the mean and covariance matrix, respectively. They are given by :

$$\hat{\mathbf{f}}' = \arg\max_{\grave{\mathbf{f}}}\left(P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}})\right) \quad , \quad (4.7)$$

$$\grave{A} = -\bigtriangledown\bigtriangledown \ln\left(P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}})\right)\Big|_{\grave{\mathbf{f}}=\hat{\mathbf{f}}'} \quad , \quad (4.8)$$

The covariance matrix represents the Hessian of the negative log posterior at the maximum point. In order to compute $\hat{\mathbf{f}}'$ and $\grave{A}$, we can use the posterior $P(\grave{\mathbf{f}}|\grave{\boldsymbol{D}})$ formulated in (Eq 4.3). By taking the logarithm of this posterior and introducing the expression of (Eq 4.4) for GP priors, we obtain the following expression :

$$\psi(\grave{\mathbf{f}}) \triangleq \ln P(\grave{\mathbf{y}}|\grave{\mathbf{f}}) - \ln P(\grave{\mathbf{f}}|\grave{\mathbf{X}}) - \tfrac{1}{2}\grave{\mathbf{f}}^T\grave{\mathbf{K}}^{-1}\grave{\mathbf{f}} - \tfrac{1}{2}\ln\left|\grave{\mathbf{K}}\right| - \frac{\grave{N}}{2}\ln 2\pi \quad , \quad (4.9)$$

Differentiating twice (Eq 4.9) with respect to $\grave{\mathbf{f}}$ leads to :

$$\begin{cases} \nabla\psi(\grave{\mathbf{f}}) = \nabla \ln P(\grave{\mathbf{y}}|\grave{\mathbf{f}}) - \grave{\mathbf{K}}^{-1}\grave{\mathbf{f}} \\ \nabla\nabla\psi(\grave{\mathbf{f}}) = \nabla\nabla \ln P(\grave{\mathbf{y}}|\grave{\mathbf{f}}) - \grave{\mathbf{K}}^{-1} \end{cases} \quad (4.10)$$

At the maximum of $\psi(\grave{\mathbf{f}})$, we get

$$\hat{\mathbf{f}}' = \grave{\mathbf{K}}\nabla \ln P(\grave{\mathbf{y}}|\hat{\mathbf{f}}') \quad (4.11)$$

and the covariance matrix is approximated by the shape of $\psi(\grave{\mathbf{f}})$

$$\grave{A} \quad = \quad -(\nabla\nabla\psi(\grave{\mathbf{f}}))^{-1} \quad = \quad (\grave{\mathbf{K}}^{-1} + \grave{W})^{-1} \quad . \tag{4.12}$$

where

$$\grave{W} \quad = \quad -\nabla\nabla \ln P(\grave{\mathbf{y}}|\hat{\mathbf{f}}') \quad . \tag{4.13}$$

Since (Eq 4.11) is nonlinear, the computation of $\hat{\mathbf{f}}'$ is achieved by numerical methods such as Newton's method. Once the computation of $\hat{\mathbf{f}}'$ and $\grave{A}$ is done, the Laplace approximation to the posterior is completely defined by

$$q(\grave{\mathbf{f}}|\grave{\boldsymbol{D}}) \quad = \quad \mathcal{N}\left(\hat{\mathbf{f}}',\ (\grave{\mathbf{K}}^{-1} + \grave{W})^{-1}\right) \quad . \tag{4.14}$$

The prediction of the point $\mathbf{x}_*$ is evaluated by exploiting the Gaussian approximation in (Eq 4.1)

$$P(y_* = +1|\grave{\boldsymbol{D}}, \mathbf{x}_*) \quad \simeq \quad q(y_* = +1|\grave{\boldsymbol{D}}, \mathbf{x}_*) \quad = \quad \int P(y_*|f_*)\ q(f_*|\grave{\boldsymbol{D}}, \mathbf{x}_*)\ df_* \quad . \tag{4.15}$$

where $q(f_*|\grave{\boldsymbol{D}}, \mathbf{x}_*)$ is Gaussian with mean and variance given as follows :

$$\begin{cases} \mu_* & = & \grave{k}_*^T\ \grave{\mathbf{K}}^{-1}\ \hat{\mathbf{f}}' \\ \sigma_*^2 & = & k(\mathbf{x}_*, \mathbf{x}_*) - \grave{k}_*^T\ (\grave{\mathbf{K}} + \grave{W}^{-1})^{-1}\ \grave{k}_* \end{cases} . \tag{4.16}$$

and $\grave{k}_*^T = [k(\mathbf{x}_1, \mathbf{x}_*)\ k(\mathbf{x}_2, \mathbf{x}_*)\ ...\ k(\mathbf{x}_N, \mathbf{x}_*)\ k(\mathbf{x}_{N+1}, \mathbf{x}_*)\ ...\ k(\mathbf{x}_{\grave{N}}, \mathbf{x}_*)]$ is a vector of kernel distances (covariances) between $\mathbf{x}_*$ and all the training and neighbourhood samples.

Note that if the class posterior probability value is not desired but just the estimate of the label of the sample $\mathbf{x}_*$, one can adopt the following labeling rule: Assign $\mathbf{x}_*$ to label "+1" if $\mu_* \geq 0$; otherwise, assign it to label "−1".

In the above mathematical derivations, we made the hypothesis that the true

contextual labels in $\mathbf{y}_n^*$ are a priori known. Since it is not the case, the Spatial GPC (SGPC) will implemented within an iterative scheme, as described in the following pseudo-code.

Step 1. Initialization

1.1) Read training set $\boldsymbol{D}$ and image I.

1.2) For each sample of I, compute the predicted label using the standard pixelwise GPC.

1.3) Output: initial classification map $Y^{(0)}$.

Step 2. Iterative spatial contextual Gaussian process

2.1) Set iteration index $t \longleftarrow 1$.

2.2) Repeat up to convergence

2.2.1) For each sample $\mathbf{x}_*$ of image I:

- Generate contextual training set $\grave{\boldsymbol{D}}$ from $\boldsymbol{D}$, I and $Y^{(t-1)}$.
- Compute $\grave{W}$ according to (Eq 4.13).
- Apply (Eq 4.11) and (Eq 4.12) to get $\hat{\mathbf{f}}^{'}$ and $\grave{A}$ respectively.
- Estimate label $y_*^{(t)}$ from (Eq 4.16).

2.2.2) Output $Y^{(t)}$

2.2.3) Increment t.

4.2 Expremental results

4.2.1 Data set

We used an image acquired over an urban area, this image was acquired over a part Boumerdes city (Algeria) in 2002 by the Quickbird sensor with a resolution of 1 m (see

Fig 4.2). It is characterized by four channels (Red, Green, Blue and Near infrared). The ground truth includes nine thematic classes, namely, water, sand, trees, asphalt, pavement, rocks, roof1 (tile roof), roof2 (cement roof), and bare soil, see (Fig 4.3) and (4.4). The (Tab 4.1) lists the numbers of training and test samples used for each class.

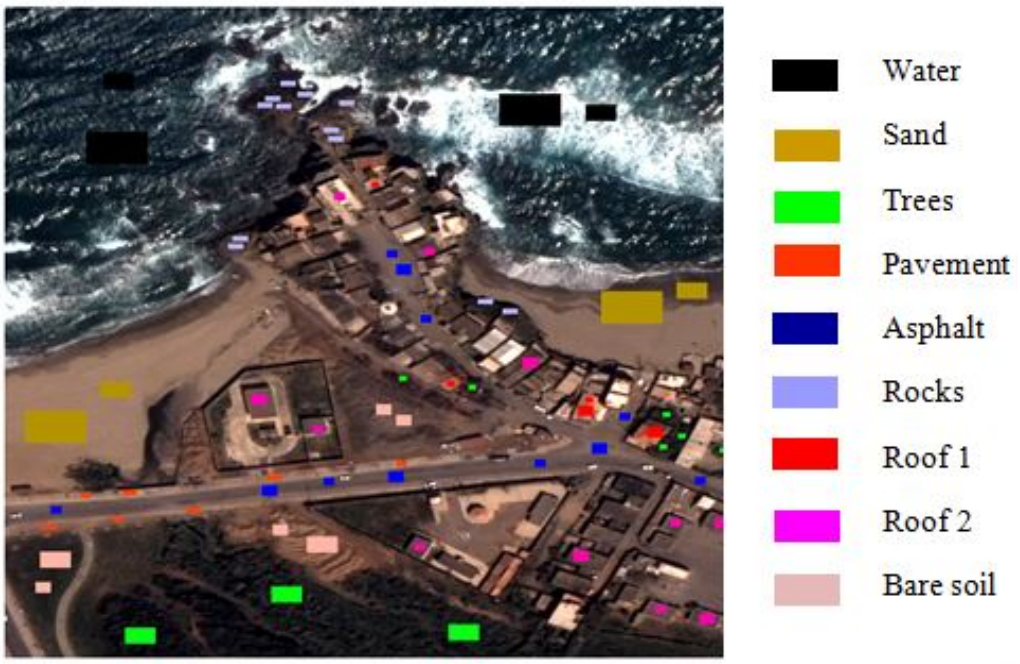


Figure 4.2: RGB composition of the image used in the experiments.

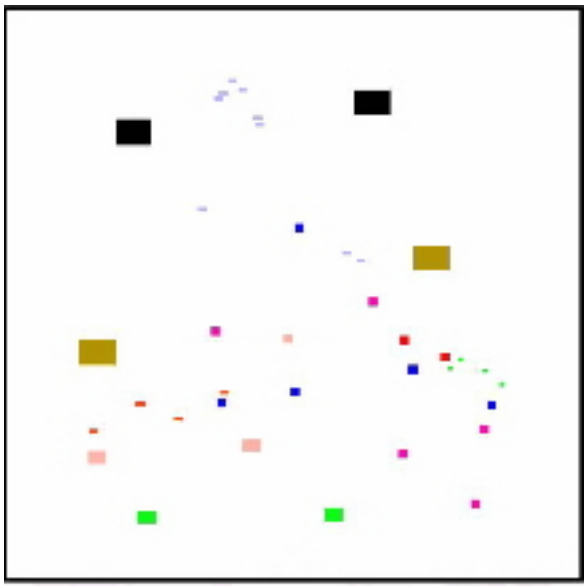


Figure 4.3: Test samples used in experiments.

Table 4.1: Numbers of training and test samples used in experiments.

Class name	Numbers of training samples	Numbers of test samples
1– Water	600	2400
2– Sand	600	2400
3– Trees	375	700
4– Pavement	105	200
5– Asphalt	343	500
6– Rocks	175	450
7– Roof1	75	200
8– Roof2	294	500
9– Bare soil	300	700
Total	**2867**	**8050**

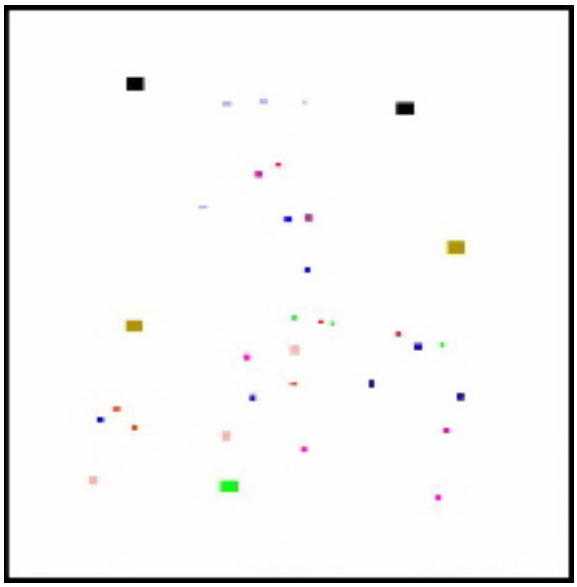


Figure 4.4: Training samples used in the experiments.

4.2.2 Results

In all experiments, the covariance function adopted is the well-known squared exponential covariance function. The hyperparameters of the models for both the standard GPC and the proposed SGPC classifiers were estimated according to the procedure based on the maximization of the log marginal likelihood as described in Rasmussen and Williams (2006).

Classification performance was evaluated in terms of four measures, which are :

1) Overall accuracy (OA), which is the percentage of correctly classified pixels among all the pixels considered (independently of the classes they belong to).

2) Average accuracy (AA), which is the average over the classification accuracies obtained for the different classes.

3) Class-specific accuracy, which is the percentage of correctly classified pixels among the pixels of the considered class.

4) McNemar's test, which gives the statistical significance of differences between the accuracies achieved by the different classification methods.

Table 4.2: Accuracies achieved by the investigated classifiers on the test samples(iteration 1).

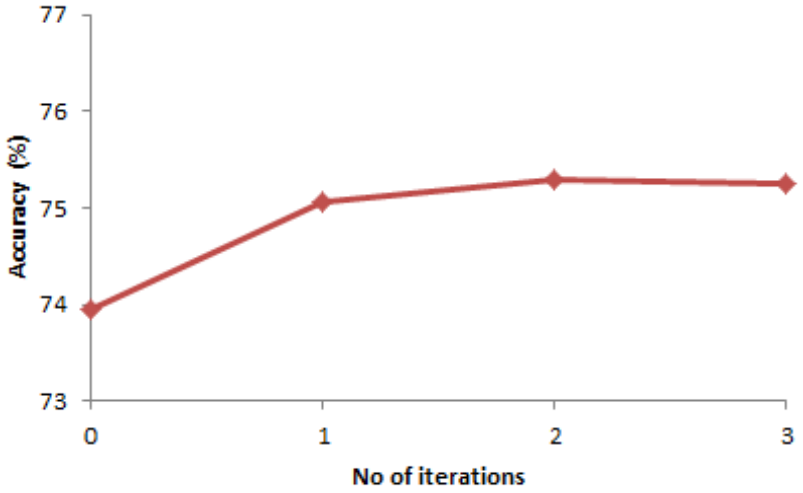
		GPC method	SGPC method (iteration 1)		
			3×3 size	5×5 size	7×7 size
OA (%)		73.95	75.07	74.98	75.03
AA (%)		66.42	67.77	67.58	67.82
Class-specific accuracies (%)	1	100	99.00	98.96	99.00
	2	61.42	63.58	63.54	63.25
	3	93.57	95.71	95.71	95.43
	4	42.50	46.00	45.50	46.50
	5	85.40	85.40	85.40	85.40
	6	34.75	36.89	36.22	36.67
	7	70.50	71.00	70.50	70.50
	8	67.20	71.40	71.40	71.80
	9	42.43	41.00	41.00	41.86

Table 4.3: Accuracies achieved by the investigated classifiers on the test samples(iteration 2).

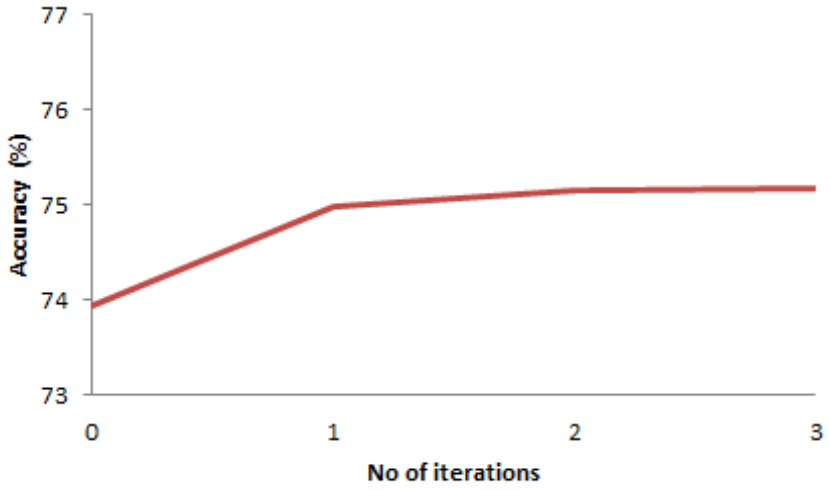
		GPC method	SGPC method (iteration 2)		
			3×3 size	5×5 size	7×7 size
OA (%)		73.95	75.29	75.15	75.22
AA (%)		66.42	68.03	67.96	68.14
Class-specific accuracies (%)	1	100	99.04	99.08	99.13
	2	61.42	63.83	63.46	63.25
	3	93.57	95.71	95.86	96.14
	4	42.50	45.50	46.00	46.00
	5	85.40	85.60	85.80	85.80
	6	34.75	36.22	35.78	36.22
	7	70.50	72.00	72.00	72.50
	8	67.20	72.40	71.80	72.60
	9	42.43	42.00	41.86	41.86

Table 4.4: Accuracies achieved by the investigated classifiers on the test samples(iteration 3).

		GPC method	SGPC method (iteration 3)		
			3×3 size	5×5 size	7×7 size
OA (%)		73.95	75.14	75.17	75.15
AA (%)		66.42	67.82	68.05	68.12
Class-specific accuracies (%)	1	100	98.96	99.08	99.03
	2	61.42	63.73	63.63	63.25
	3	93.57	95.86	95.71	96.10
	4	42.50	46.20	46.50	46.10
	5	85.40	85.40	85.00	85.60
	6	34.75	36.67	36.00	36.22
	7	70.50	70.00	73.00	72.50
	8	67.20	72.60	72.60	72.59
	9	42.43	41.29	41.00	41.70

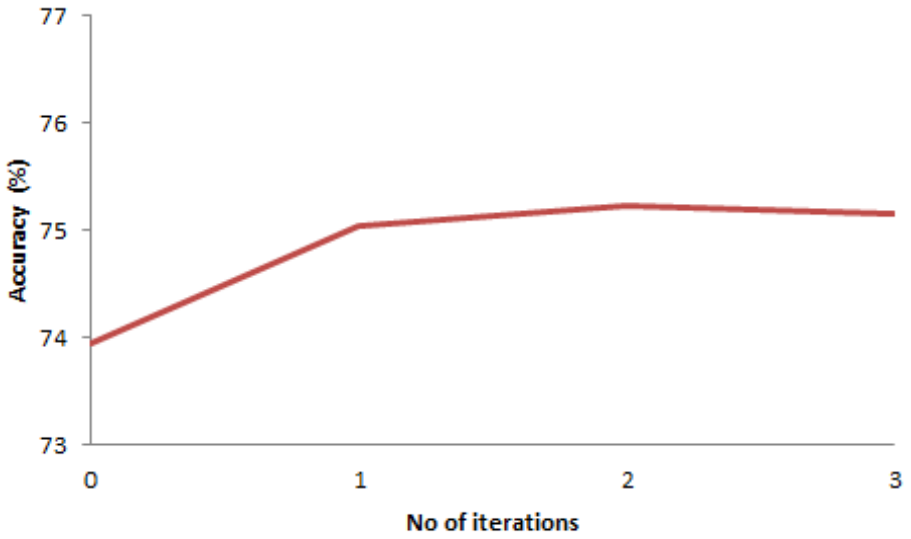


Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the number of iterations (3 pixels $\times$3 pixels) .

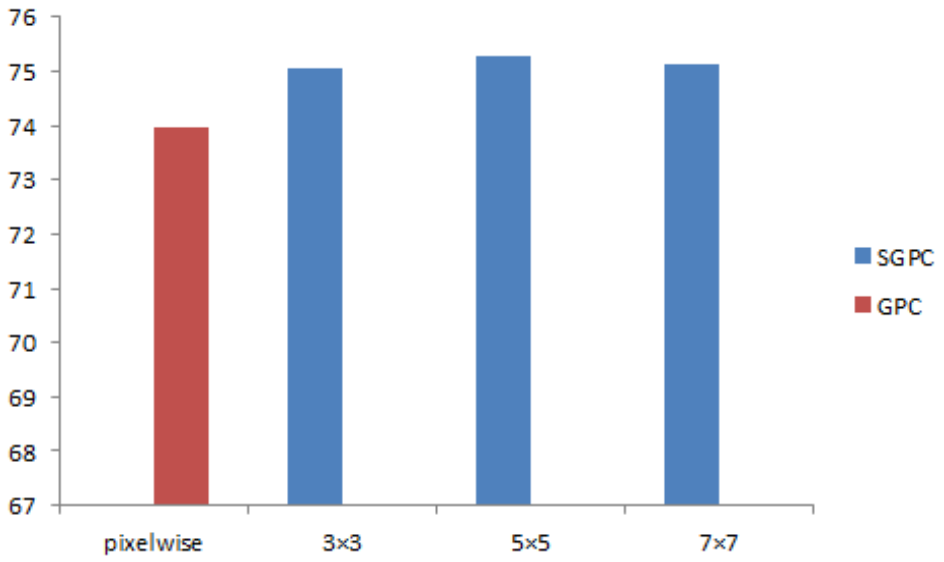


Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the number of iterations (5 pixels ×5 pixels).

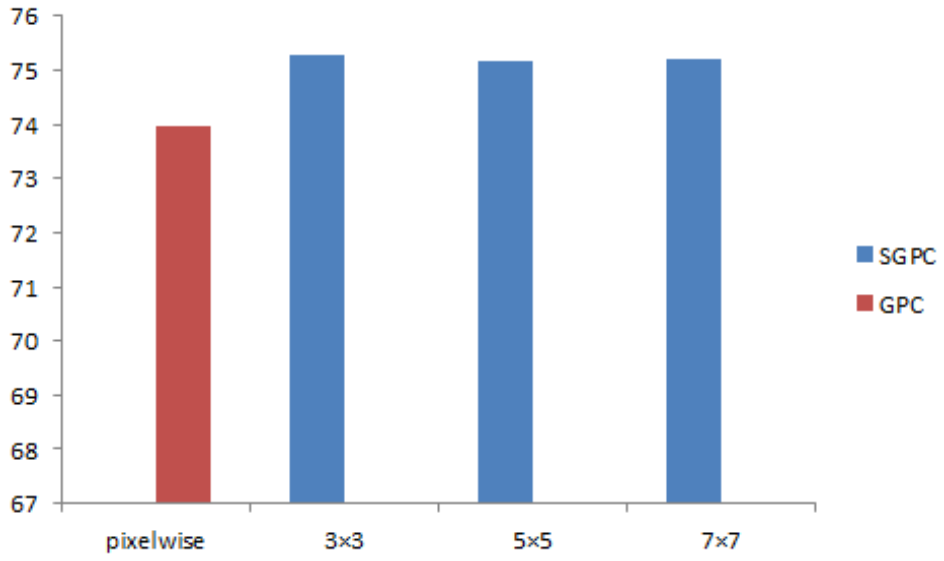
.



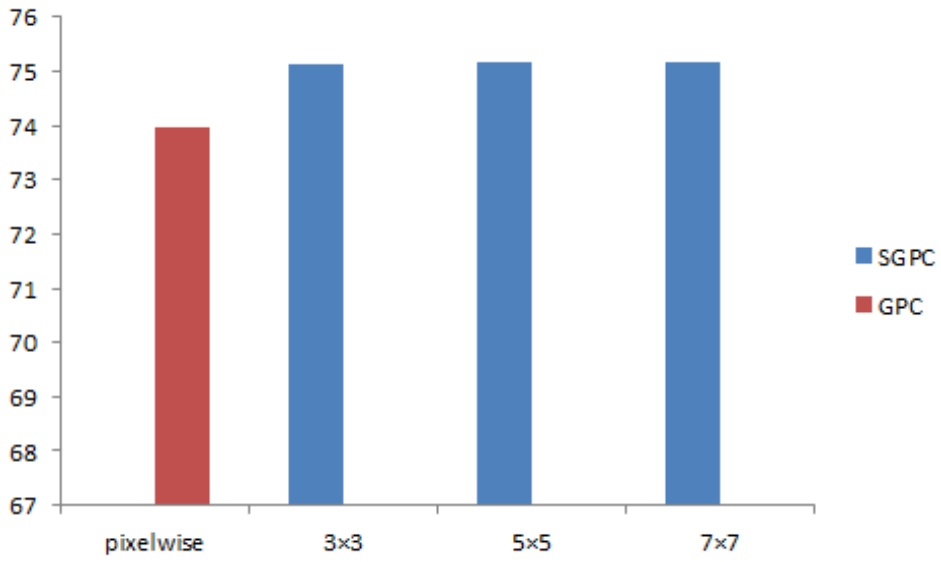
Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the number of iterations (7 pixels ×7 pixels).



Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the size of the neighborhood system. "Pixelwise" stands for the standard GPC classifier (iteration 1).



Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the size of the neighborhood system. "Pixelwise" stands for the standard GPC classifier (iteration 2).



Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the size of the neighborhood system."Pixelwise" stands for the standard GPC classifier (iteration 3).

4.2.3 Interpretation

At first, we performed experiments by considering a neighborhood system of 3 pixels ×3 pixels. In particular, we run the proposed SGPC method up to convergence. As it can be seen in (Fig 4.2.2), (Fig 4.2.2) and (Fig 4.2.2), convergence was achieved at third iterations. The main improvement was obtained at the first iteration, and then the accuracy stabilized. The detailed results achieved at convergence are reported in (Tab 4.2), (Tab 4.3) and (Tab 4.4). Compared to the standard SGPC, we have :

1) An improvement of about 1.12% in OA and 1.35% in AA when neighborhood system of 3 pixels ×3 pixels.

2) An improvement of about 1.34% in OA and 1.61% in AA when neighborhood system of 5 pixels ×5 pixels.

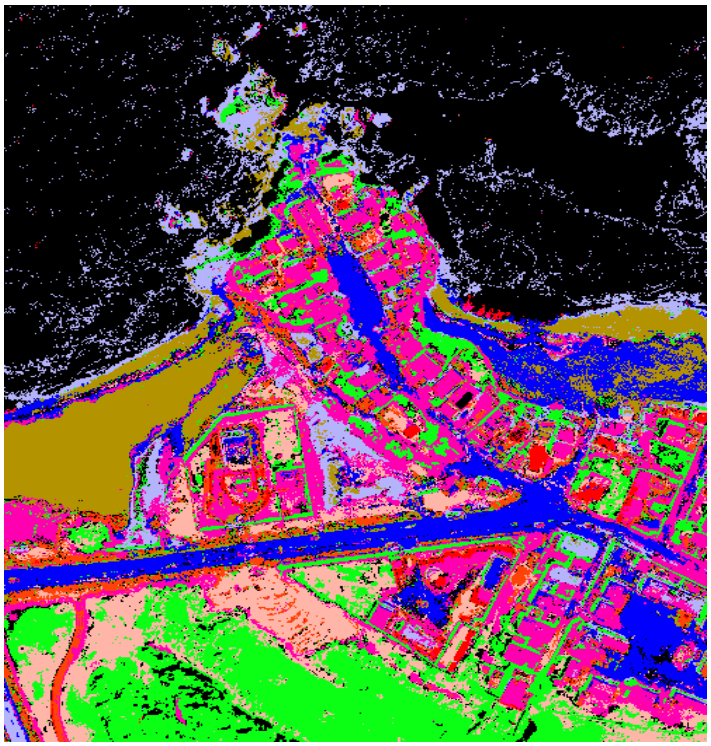
3) An improvement of about 1.20% in OA and 1.50% in AA when neighborhood system of 7 pixels ×7 pixels.

Most of the classes take profit from the exploitation of spatial contextual information, and in particular

1) The class 'roof2' for which a boost of more than 4% is observed when neighborhood system of 3 pixels $\times$3 pixels.
2) The class 'roof2' for which a boost of more than 5% is observedwhen neighborhood system of 5 pixels $\times$5 pixels.
3) The class 'roof2' for which a boost of more than 5% is observedwhen neighborhood system of 7 pixels $\times$7 pixels.

The McNemar's test provided us a value of 24, confirming thus that the differences between the SGPC and the standard GPC classifiers are statistically significant.

Finally, in order to analyze the impact of the size of the neighborhood system on the classification results, we repeated the previous experiments by adopting increasing values for the window size, namely 5 pixels $\times$ 5 pixels and 7 pixels $\times$ 7 pixels. The convergence was achieved in all cases at the third iteration. The overall accuracies yielded by the SGPC method are plotted in (Fig 4.2.2), (Fig 4.2.2) and (Fig 4.2.2), which suggests that the size of the neighborhood is not critical.



result image by SGPC method.

4.3 The comparaison between SGPC method and MP-GPC method

For the sake of comparaison, we run the GPC classifier fed with an additional set of 8 morphological profile (MP) features, concatenated with the 4 original faetures (in total 12 features).

(a) (b) (c) (d)

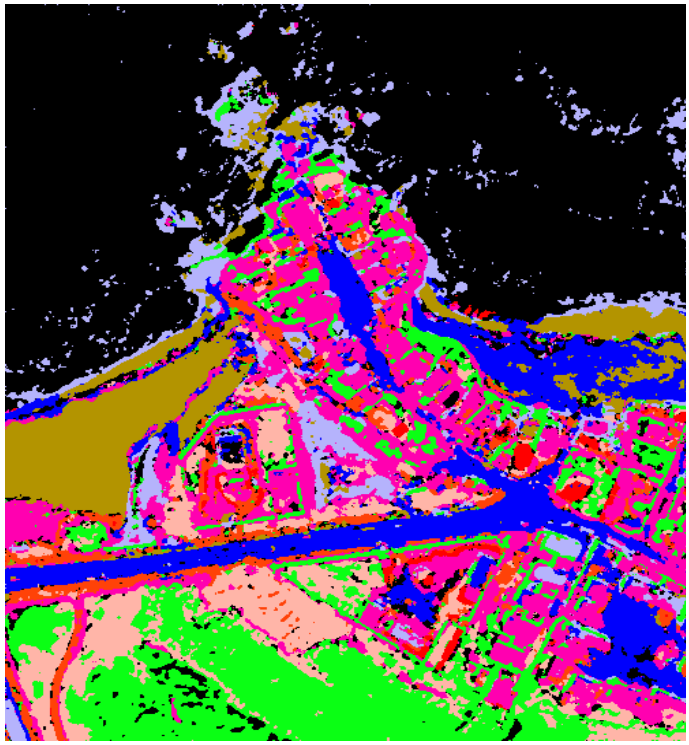
channels

(a) (b) (c) (d)

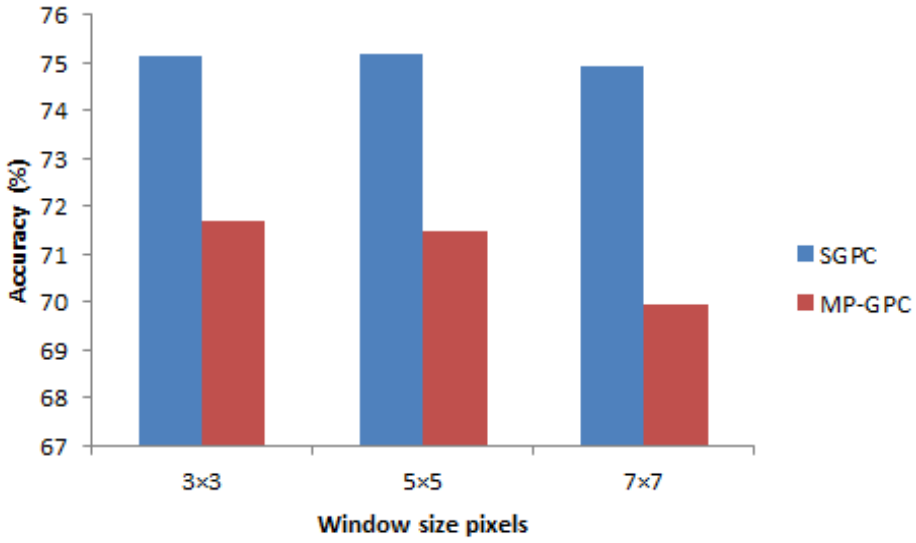
Closing with SE square at size 3

Opening with SE square at size 3

The MP was generated by applying opening and closing operations with a square-shape stucturing element. The ovell accuracies yielded by the SGPC and the MP-GPC methods are plotted in (Fig 4.3), which suggests that : 1) the size of the neighbourhood is not critical for SGPC and 2) SGPC outperforms MP-GPC.



result image by MP method.



Overall accuracy achieved on the test samples by the investigated SGPC classifier versus the standard MP-GPC classifier

Table 4.5: Accuracies achieved by the investigated classifiers (SGPC, MP-GPC) on the test samples

		MP method	SGPC method
OA (%)		71.69	75.95
AA (%)		65.00	67.61
Class-specific accuracies (%)	1	98.63	98.96
	2	57.08	63.63
	3	78.86	95.86
	4	37.50	46.00
	5	90.20	85.40
	6	31.33	36.67
	7	77.50	70.00
	8	68.80	72.60
	9	45.14	41.29

Conclusion

In this thesis, we have proposed an innovative GPC model, which embeds iteratively the spatial contextual information in the classification process. To the best of our knowledge, nothing similar has been yet introduced, at least in the remote sensing literature. It is a general method which does not impose any constraint on the typology of the input features. The proposed SGPC model showed to be able to well capture additional useful information from the spatial context. Experimental results show that, despite the high quality of the test image, the SGPC improved significantly the classification accuracy over the baseline GPC.

The size of the neighborhood system appeared not critical in our experiments. However, we think that the optimal size may depend on the image resolution. For instance, for satellite images of metric resolution, we suggest that a 3×3 or 5×5 window size pixels could be satisfactory. For centimetric resolution centimetre-resolution images (acquired with sensors mounted on airborne aeroplane or UAV), the best window size may increase be larger. Work is in progress :

1) To find a way to estimate automatically this parameter within the SGPC learning process.

2) To develop an alternative non-iterative GPC model which integrates spatial contextual information in the covariance matrix.

Bibliography

[1] ABRAMOWITZ, M. and STEGUN, I. A. *Handbook of Mathematical Functions*. Dover, New York. (1965).

[2] ADLER, R. J. *The Geometry of Random Fields*. Wiley, Chichester. (1981).

[3] ANDRE, M. *Introduction aux Techniques de Traitement d'Images*, Eyrolles (1987).

[4] ASSAS, O. *Classification Floue des Images*. PhD thesis. (2013).

[5] BARBER, D., WILLIAMS, C. K. I. *Gaussian Processes for Bayesian Classification via Hybrid Monte Carlo*. In Advances in Neural Information Processing Systems (1997),M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9, The MIT Press.

[6] BAZI, Y. ALAJLAN, N, and MELGANI, F. *Improved Estimation of Water Chlorophyll Concentration with Semisuper vised Gaussian Process Regression*. (2012). IEEE Transactions on Geoscience and Remote Sensing 50 : 2733 − 2743. doi: 10.1109/$TGRS$.2011.2174246.

[7] BAZI, Y. MELGANI, F. *Classification of Hyperspectral Remote Sensing Images Using Gaussian Process*. (2008).

[8] BAZI, Y. MELGANI, F. *Gaussian process approach to remote sensing image classification*. . (2010). IEEE Transactions on Geoscience and Remote Sensing 48 : 186−197. doi: 10.1109/TGRS.2009.2023983.

[9] BISHOP, C. *Neural Networks for Pattern Recognition*. Oxford University Press. (1995).

[10] BOYLE, P. *Gaussian Process for Regression and Optimisation*. PhD thesis, Victoria University of Wellington. (2007), pp. $10 - 12$.

[11] BUNTINE, W., AND WEIGEND, A. *Bayesian Backpropagation*. Complex Systems 5(1991), pp. $603 - 643$.

[12] BRACEWELL, R. N. *The Fourier Transform and its Applications*. McGraw-Hill, Singapore, international edition. (1986).

[13] BRETON, J. C. Processus Gaussiens. Université de la Rochelle. (2006).

[14] CHATFIELD, C. *The Analysis of Time Series : An Introduction*. Chapman and Hall, London, 4th edition. (1989).

[15] CORNFORD, D., NABNEY, I. T., and WILLIAMS, C. K. I. *Modelling Frontal Discontinuities in Wind Fields*. Journal of nonparametric statsitics. (2002).

[16] CRESSIE, N. *Statistics for Spatial Data*. Wiley. (1993).

[17] CSATÓ , L. *Gaussian Processes - Iterative Sparse Approximation*. PhD thesis, Aston University. (2002).

[18] CSATÓ , L., OPPER, M. *Sparse Representation for Gaussian Process Models*. In Advances in Neural Information Processing Systems, NIPS (2001), vol. 13, pp. $444 - 450$.

[19] CSATÓ , L., OPPER., M. *Sparse on-Line Gaussian processes*. Neural Computation 14(2002). pp. $641 - 668$.

[20] DALLAIRE, P. *Apprentissage par Renforcement Bayésien de Processus Décisionnel de Markov Partiellement Obsarvable : Une Approche Basé sur les Processus Gaussiens*. (2010).

[21] DUDA, R. HART, S. and STORK, D. G. *Pattern Classification*, 2nd edition. New York : Wiley. (2001).

[22] FOODY, G. M. MATHUR. A. *A Relative Evaluation of Multiclass Image Classification by Support Vector Machines*. IEEE Trans. Geosci. Remote Sensing. vol. 42, no. 6, pp. 1335 – 1343. (2004).

[23] GHOGGALI, N. MELGANI, F. and BAZI, Y. *A Multiobjective Genetic SVM Approach for Classification Problems with Limited Training Samples*. IEEE Trans. Geosci. Remote Sensing. vol. 47, no. 6, pp. 1707 – 1718. (1998).

[24] GIBBS, M. *Bayesian Gaussian Processes for Classification and Regression*. PhD thesis, University of Cambridge, Cambridge, U.K.. (1997).

[25] GIBBS, M., MACKAY, D. J. *Efficient Implementation of Gaussian Processes*. `http://www.inference.phy.cam.ac.uk/mackay/abstracts/gpros.html`. (1996).

[26] GIBBS, M. N., MACKAY, D. J. *Variational Gaussian Process Classifiers*. IEEE Trans. on Neural Networks $11, 6(2000)$, pp. 1458 – 1464.

[27] GIHMAN, I. I. and SKOROHOD, A. V. *The Theory of Stochastic Processes*, vol 1. Springer Verlag, Berlin. (1974).

[28] GIROLAMI, M. ROGERS, S. *Variational Baysian Multinomial Probit Regression with Gaussian Process Priors*. Neural Comput. vol. 18, no. 8, pp. 1790 – 1817. (2006).

[29] GOLDBERG, P. W. WILLIAMS, C. K. I., AND BISHOP, C. M. *Regression with Input-Dependent Noise: A Gaussian Process Treatment. In Advances in Neural Information Processing Systems*, NIPS (1998), M. J. Jordan, M.I. Kearns and S. A. Solla, Eds., vol. 10.

[30] GOSSELIN, B.*Application des réseaux de neurones artificielles aux reconnaissances automatique de caractères manuscrits*. Thèse de Doctorat, Faculté Polytique de Mons. (1996).

[31] HASSOUNA, H. MELGANI, F. and MOKHTARI, Z. *Spatial Contextual Gaussian Process Learning for Remote Sensing Image Classification*. Remote Sensing Letters,6 : $7,519-528$, doi : $10.1080/2150704X.2015.1051628$.

[32] HULTQUIST, C., G. CHEN, and K. ZENG. *A Comparison of Gaussian Process Regression, Random Forests and Support Vector Regression for Burn Severity Assessment in Diseased Forests.* (2014). Remote Sensing Letters 5 : $723-732$. doi: $10.1080/2150704X.2014.963733$.

[33] JONES, D. R. *A Taxonomy of Global Optimization Methods Based on Response Surfaces*. Journal of Global Optimization 21(2001), pp. $345-383$.

[34] JUN, G., and J. GHOSH. *Spatially Adaptive Classification of Hyperspectral Data with Gaussian Processes*. (2009). IEEE International Geoscience and Remote Sensing Symposium 2 : $II-290-II-293$. doi: $10.1109/IGARSS.2009.5418067$.

[35] KIMA, H. C, . GHAHRAMANI, Z. *Baysian Gaussian Process Classification with the EM-EP Algorithm.* (2006).

[36] LAMPINEN, J., AND VEHTARI, A. *Bayesian neural netwroks - review and case studies*. Neural Networks $14,3(2001)$. pp. $7-24$.

[37] MACKAY, D. J. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. (2003).

[28] MACKAY, D. J. *Gaussian Processes: A Replacement for Supervised Neuralnetworks? In NIPS97 Tutorial*. (1997).

[39] MACKAY, D. J. C. *Introduction to Gaussian process.* In Bishop, C. M., editor, Neural networks and machine learning. Springer-Verlag. (1998).

[40] MACKAY, D. J. C. *Probable Networks and Plausible Predictions a Review of Practical Bayesian Methods for Supervised Neural Networks.Network: Computation in Neural Systems* 6(1995). pp. 469 − 505.

[41] MACKAY, D. J. C. *Introduction to Monte Carlo Methods. In Learning in Graphical Models*, M. I. Jordan, Ed., NATO Science Series. Kluwer Academic Press. (1998), pp. 175 − 204.

[42] MACKAY, D. J. C. *Introduction to Gaussian processes. In Neural Networks and Machine Learning*, C. M. Bishop, Ed., NATO ASI Series. Kluwer. (1998), pp. 133 − 166.

[43] MACKAY, D. J. C. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology. (1992).

[44] MATERN, B. Spatial Variation. *Meddelanden fran statens skogsforor skningsinstitut*, Almänna Förlaget, stockholm. Second edition (1986), Springer-Verlag, Berlin. (1960).

[45] MATHERON, G. *Principles of Geostatistics. Economic Geology 58*(1963), pp. 1246 − 1266.

[46] MINGYUE, T. Expectation Propagation of Gaussian Process Classification and Its Application toGene Expression Analysis.

[47] MINKA, T. P. *A family of Algorithms for Approximate Bayesian Inference.* PhD thesis, Massachusetts institute of technology. (2001).

[48] MOUINE, S. *Traitement Morphologique des Images de Feuilles.*PhD thesis.

[49] NEAL, R. *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, No 118. Springer-Verlag. (1996).

[50] NEAL, R. *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification.* Tech. Rep. CRG-TR−97 − 2, Dept. of Computer Science, Univ. of Toronto. (1997).

[51] NEAL, R. M. *Bayesian Training of Backpropagation Networks by the Hybrid Monte Carlo Method*. Tech. Rep. CRG-TR−92 − 1, Dept. of Computer Science, Univ. of Toronto. (1992).

[52] O'HAGAN, A. *Curve Fitting and Optimal Design for Prediction* (with discussion). J. Roy. Statist. Soc. Ser. B 40(1978), pp. 1 − 42.

[53] OPPER, M. and Winther, O. *Gaussian processes for classification : Mean-field algorithms*. Neural Computation. (2000).

[54] PACIOREK, C. *Nonstationary Gaussian Processes for Regression and Spatial Modelling*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A. (2003).

[55] PACIOREK, C., AND SCHERVISH, M. *Nonstationary Covariance Functions for Gaussian Process Regression*. In Advances in Neural Information Processing Systems, NIPS 16(2004), pp. 273 − 280.

[56] PASOLLI, L., F. MELGANI, and E. BLANZIERI. *Gaussian Process Regression for Estimating Chlorophyll Concentration in Subsurface Waters from*. (2010). Remote Sensing Data. IEEE Geoscience and Remote Sensing Letters 7 : 464 − 468. doi: 10.1109/*LGRS*.2009.2039191.

[57] QUIÑONERO-CANDELA, J., AND RASMUSSEN, C. E. A *Unifying View of Sparse Approximate Gaussian Process Regression*. Journal of Machine Learning Research 6, 12(2005). pp. 1935 − 1959.

[58] UHLENBECK, G. E. and ORNSTIEN, L. S. *On the Theory of Brownian Motion*. Phys. Rev. (1930).

[59] RASMUSSEN, C. E. *Evaluation of Gaussian Processes and Other Methods for Non-Linear Regression*. PhD thesis, Graduate Department of Computer Science, University of Toronto. (1996).

[60] RASMUSSEN, C. E. *Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals*. In Bayesian Statistics (2003), J. M. Bernardo,M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F.M. Smith, and M. West, Eds., vol. 7, Oxford University Press, pp. $651 - 659$.

[61] RASMUSSEN, C. E., AND KUSS, M. *Gaussian Processes in Reinforcement Learning*. In Advances in Neural Information Processing Systems, NIPS (2002), S. Thrun, L. Saul, and B. Schlkopf, Eds., vol. 16, The MIT Press.

[62] RASMUSSEN, C. E., and C. K. I. Williams. *Gaussian Process for Machine Learning. Cambridge*, MA: MIT Press. (2006).

[63] RASMUSSEN, C. E., AND GHAHRAMANI, Z. *Infinite Mixtures of Gaussian Process Experts*. In Advances in Neural Information Processing Systems, NIPS (2002), T. G. Diettrich, S. Becker, and Z. Ghahramani, Eds., vol. 14, The MIT Press.

[64] RUMELHART, D., HINTON, G., AND WILLIAMS, R. *Learning Representations by Back-Propagating Errors*. Nature 323(1986), pp. $533 - 536$.

[65] SEEGER, M. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh. (2003).

[66] SEEGER, M. *Gaussian Processes for Machine Learning*. Tech. rep., Department of EECS, University of California at Berkeley. (2004).

[67] SEEGER, M. *Gaussian Processes for Machine Learning. International* Journal of Neural Systems $14, 2(2004)$, pp. $1 - 38$.

[68] SEEGER, M., AND WILLIAMS, C. *Fast Forward Selection to Speed up Sparse Gaussian Process Regression.* (2003). InWorkshop on AI and Statistics 9.

[69] SMOLA, A. J., AND BARTLETT, P. L. *Sparse Greedy Gaussian Process Regression.* In Advances in Neural Information Processing Systems, NIPS (2001), vol. 13, pp. 619 – 625.

[70] SNELSON, E., C. E. R., AND GHAHRAMANI, Z. *Warped Gaussian Processes. In Advances* in Neural Information Processing Systems, NIPS (2004), L. S. Thrun, S. and B. Schlkopf, Eds., vol. 16, pp. 337 – 344.

[71] SNELSON, E. L. *Flexible and Effcient Gaussian Process Models for Machine Learning. PhD Thesis, London University.* (2007). pp. 21 – 22.

[72] SOLAK, E., MURRAY-SMITH, R., LEITHEAD, W. E., LEITH, D., AND RASMUSSEN, C. E. *Derivative Observations in Gaussian Process Models of Dynamic Systems.* In Advances in Neural Information Processing Systems, NIPS (2003), vol. 15, The MIT Press, pp. 1033 – 1040.

[73] SUN, S., P. ZHONG, H. XIAO, and R. WANG. *Active Learning with Gaussian Process Classifier for Hyperspectral Image Classification.* (2015). IEEE Transactions on Geoscience and Remote Sensing 53 : 1746 – 1760. doi: 10.1109/$TGRS$.2014.2347343.

[74] STEIN, M. L. *Interpolation of Spatial Data.* Springer-Verlag, New York.(1999).

[75] TAYLOR, J. S. CRISTIANININ. N. *Kernel Methods for Pattern Analysis.* Cambridge, U.K.: Cambridge University. Press. (2004).

[76] TRESP, V. A *Bayesian Committee Machine.* Neural Computation 12, 11(2000), pp. 2719 – 2741.

[77] TRESP, V. *Mixtures of Gaussian Processes*. In Advances in Neural Information Processing Systems, NIPS (2001), T. K. Leen, T. G. Dietterich, and T. V., Eds., vol. 13, The MIT press.

[78] VAPNIK, V. *Statistical Learning Theory*. New York : Wiley. (2004)

[79] VON, v. *Gaussian Process Models for Robist Regression, Classification and Reinforcement Learning*. (2006).

[80] WANG, J., FLEET, D., AND HERTZMANN, A. *Gaussian Process Dynamical Models*. In Advances in Neural Information Processing Systems, NIPS (2006), Y. Weiss, B. Schlkopf, and J. Platt, Eds., vol. 18, The MIT Press, pp. $1443 - 1450$.

[81] WILLIAMS, C. K. I. *Computation with Infinite Neural Networks*. Neural computation. (1998).

[82] WILLIAMS, C. K. I. and Barber, D. *Bayesian Classification with Gaussian Process*. IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no. 12, pp. $142 - 1351$. (1998).

[83] WILLIAMS, C. K., AND RASMUSSEN, C. E. *Gaussian Processes for Regression*. In Advances in Neural Information Processing Systems (1996), D. Touretzsky, M. Mozer, and M. Hasselmo, Eds., vol. 8

[84] WILLIAMS, C. K. I., RASMUSSEN, C. E., SCHWAIGHOFER, A., AND TRESP, V. *Observations on the Nyström Method for Gaussian Process Prediction*. Tech. rep., University of Edinburgh. (2002).

[85] WILLIAMS, C. K. I., AND SEEGER, M. *Using the Nyström Method to Speedup Kernel Machines*. Advances in Neural Information Processing Systems 13. (2001).