Classification in functional spaces using the BV norm with applications to ophthalmologic images and air traffic complexity

Bang Giang Nguyen

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse III - Paul Sabatier (UPS)

Discipline ou spécialité :
Mathématiques appliquées

Présentée et soutenue par :

NGUYEN Bang Giang

le 03 novembre 2014

Titre:

Classification en espaces fonctionnels utilisant la norme BV
avec applications aux images ophtalmologiques et à la complexité du trafic aérien
Classification in functional spaces using the BV norm
with applications to ophthalmologic images and air traffic complexity

---

École doctorale :
Mathématiques, Informatique et Télécommunications de Toulouse (MITT) - ED 475

Unité de recherche :
Institut de Mathématiques de Toulouse (IMT) - UMR CNRS 5219
Laboratoire de Mathématiques Appliquées Informatique et
Automatique pour l'Aérien (MAIAA), École Nationale de l'Aviation Civile (ENAC)

Directeurs de thèse :
Mr MARÉCHAL Pierre, ISAE
Mr DELAHAYE Daniel, ENAC

Rapporteurs :
Mr DINH The Luc Université d'Avignon
Mr LANNES André École supérieure d'électricité

Examinateurs :
Mme NGUYEN-VERGER Mai K. Université de Cergy-Pontoise
Mr GWIGGNER Claus Universität Hamburg
Mr MALGOUYRES François Université Paul Sabatier

Acknowledgements

First of all I would like to express the deepest gratitude to my thesis supervisors Pierre MARÉCHAL and Daniel DELAHAYE for their guidance, great support and kind advice throughout my PhD studies. I also would like to thank Stéphane PUECHMOREL who gave me many ideas in the thesis and shared with me his exceptional scientific knowledge.

I would like to thank Priscille OLLE for her kindness, for proposing the first problem in my thesis. I am grateful to her for providing me the data of retinal images and showing me the base knowledge concerning eye disease in the first problem of the thesis.

I would like to express my gratitude to the reading committee, consisting of DINH The-Luc, André LANNES, NGUYEN-VERGER Mai K., Claus GWIGGNER and François MALGOUYRES.

I acknowledge the financial support of Vietnamese Government Overseas Scholarship Program for my studies.

I would like to thank to my colleagues and friends of laboratory in Applied Mathematics, Computer Science and Automatics for Air Transport (MAIAA), especially OPTIM Group for their companionship and for providing a so pleasurable and friendly atmosphere.

To Tambet, thank you for many helpful discussions, funny jokes in our office and for being great and friendly office mate.

To my friends: Bình, Phong, Minh-Liên, anh Mạnh, Đức, Hiệp, Tiến-Hằng, Trường, thank you for your unconditional friendship. Weekend nights with you were so exciting, enjoyable and unforgettable.

I express my gratitude to Vietnamese math group in Toulouse.

I thank to the Vietnamese football team in Toulouse of which I am a part. I hope one day you can get the champion cups of NNB tournament and Winter tournament.

Nothing could be greater than the support from my beloved family: my parents Dad Tám, Mom Dục, my elder brother Tuấn, my elder sister Oanh, my younger brother Linh. My heart belongs to my small family: my wife Minh in company with my two angels: Trâm and Trường. They love me, support me unconditionally and encourage me at all difficult moments.

Toulouse, 11/2014
NGUYỄN Bằng Giang

Abtract

In this thesis, we deal with two different problems using Total Variation concept. The first problem concerns the classification of vasculitis in multiple sclerosis fundus angiography, aiming to help ophthalmologists to diagnose such autoimmune diseases. It also aims at determining potential angiography details in intermediate uveitis in order to help diagnosing multiple sclerosis. The second problem aims at developing new airspace congestion metric, which is an important index that is used for improving Air Traffic Management (ATM) capacity.

In the first part of this thesis, we provide preliminary knowledge required to solve the above-mentioned problems. First, we present an overview of the Total Variation and express how it is used in our methods. Then, we present a tutorial on Support Vector Machines (SVMs) which is a learning algorithm used for classification and regression.

In the second part of this thesis, we first provide a review of methods for segmentation and measurement of blood vessel in retinal image that is an important step in our method. Then, we present our proposed method for classification of retinal images. First, we detect the diseased region in the pathological images based on the computation of BV norm at each point along the centerline of the blood vessels. Then, to classify the images, we introduce a feature extraction strategy to generate a set of feature vectors that represents the input image set for the SVMs. After that, a standard SVM classifier is applied in order to classify the images.

Finally, in the third part of this thesis, we address two applications of TV in the ATM domain. In the first application, based on the ideas developed in the second part, we introduce a methodology to extract the main air traffic flows in the airspace. Moreover, we develop a new airspace complexity indicator which can be used to organize air traffic at macroscopic level. This indicator is then compared to the regular density metric which is computed just by counting the number of aircraft in the airspace sector. The second application is based on a dynamical system model of air traffic. We propose a method for developing a new traffic complexity metric by computing the local vectorial total variation norm of the relative deviation vector field. Its aim is to reduce complexity. Three different traffic situations are investigated to evaluate the fitness of the proposed method.

Key words: Machine learning, retinal image, BV norm, infinite space, trajectory, main flows, complexity

Résumé

Dans cette thèse, nous traitons deux problèmes différents, en utilisant le concept de variation totale. Le premier problème est la classification des vascularites dans l'angiographie du fond d'oeil, et a pour but de faciliter le travail des ophtalmologistes pour diagnostiquer ce type de maladies auto-immunes. Il vise aussi à identifier sur les angiographies les éléments permettant de diagnostiquer la sclérose en plaques. A partir de certains résultats du premier problème, un second problème a pu être abordé, consistant à développer une nouvelle métrique de congestion d'espace aérien. Cette métrique permet de quantifier la complexité de gestion du trafic aérien dans une zone donnée et s'avère très utile dans les processus d'optimisation du système de gestion du trafic aérien (Air Traffic Management, ATM).

Dans la première partie de cette thèse, nous introduisons les notions requises pour résoudre ces deux problèmes. Tout d'abord nous présentons le principe de variation totale, ainsi que la manière dont il est utilisé dans nos méthodes. Ensuite, nous détaillons le fonctionnement des machines à vecteurs supports (Support Vector Machines, SVM), qui sont des algorithmes d'apprentissage automatique utilisés pour la classification et la régression.

Dans la deuxième partie de cette thèse, nous présentons d'abord un état de l'art des méthodes de segmentation et de mesure des vaisseaux sanguins dans les images rétiniennes, étape importante de notre méthode. Ensuite, nous décrivons notre méthode de classification des images rétiniennes. Pour commencer, nous détectons les régions pathologiques dans les images des patients malades en nous basant sur la norme BV calculée à chaque point le long de l'axe central des vaisseaux. Ensuite, pour classer les images, nous introduisons une stratégie d'extraction des caractéristiques pathologiques pour générer un ensemble de vecteurs de caractéristiques pathologiques qui représente l'ensemble d'images d'origine pour le SVM. Les images sont alors classées en utilisant des méthodes standard de classification par SVM.

Enfin, la troisième partie décrit deux applications de la variation totale dans le domaine de l'ATM. Dans la première application, en partant des idées développées dans la deuxième partie, nous introduisons une méthode d'extraction des flux principaux d'avions de l'espace aérien. En nous basant sur les algorithmes utilisés dans la deuxième partie, nous avons développé un indicateur de complexité de l'espace aérien utilisable au niveau macroscopique. Cet indicateur est ensuite comparé à la métrique de densité habituelle, qui consiste simplement à compter le nombre d'avions dans un secteur de l'espace aérien. La seconde application se base sur un modèle par systèmes dynamiques du trafic aérien. Nous proposons une nouvelle métrique de complexité du trafic basée sur le calcul de la norme locale de variation totale vectorielle de la déviation relative du champ de vecteurs. Le but est de réduire la complexité. Trois scénarios de trafic différents sont étudiés pour évaluer la qualité de la méthode proposée.

Mots clés: Apprentissage automatique, images rétiniennes, norme BV, espace infini, trajectoire, flux principaux, complexité

Contents

List of Figures

Introduction

The first part of the thesis presents the support vector machine and BV-norm concept which have been used in our applications. The second part describes the problem of classification of retinal image. To reach this goal, a vessel extraction algorithm is presented for which BV-norm histogram have been computed. Based on such histograms, a support vector machine algorithm has been used to classify pathological cases and healthy cases. The third part of the thesis presents a new airspace complexity metric which has been developed by using some techniques used in the second part.

The classification of retinal image

Retinal diseases are causing alteration of the visual perception leading sometimes to blindness. For this reason early detection and diagnosis of retinal pathologies are very important. Using digital image processing techniques, retinal images may be analyzed quickly and computer-assisted diagnosis systems may be developed in order to help the ophthalmologists to make a diagnosis.

Intermediate and/or posterior uveitis could precede onset of multiple sclerosis (MS) in 30-46 % of the patients [38]. The reported frequency of uveitis varies from 0.4% to 26.9% in multiple sclerosis patient [39]. But the prevalence of multiple sclerosis in patients with uveitis is 1-2%. Frequently it is difficult for an ophthalmologist to diagnose MS when the patient starts with vasculitis. It takes many years (8-9 years) until neurological symptoms help to diagnose MS. It seems that the prognosis of such uveitis is not so well known (visual acuity, disability). Our aim was first to analyze the angiography of patients who started the disease by intermediate uveitis and diagnosed as MS. With these results, we analyzed angiography of patients with autoimmune intermediate uveitis presumed as MS.

The first contribution of the thesis is that we proposed a method using the BV norm in order to classify vasculitis fundus angiography. Our method is aimed at helping ophthalmologists for the diagnosis of such autoimmune diseases. In particular, it will help determining potential angiography details in intermediate uveitis helping to diagnose multiple sclerosis.

Air traffic complexity

The second contribution of the thesis is the development of a new congestion metric for airspace.

When an aircraft travels between airports, a flight plan must be registered in order to inform the relevant air navigation services. This plan contains all the indicative elements needed to describe the planned flight, notably:

- departure time
- the first flight level[1] requested for the cruise
- the planned route, described using a series of markers.

Airplanes usually used pre-established routes known as airways. These airways take the form of tubular corridors with rectangular cross-sections, surrounding segments of straight lines; markers are located at the intersections of these lines. Collision risks, known as conflicts, most often arise around these markers. At these points, we define a horizontal distance, expressed in Nautical miles (NM) [2], the horizontal separation, and a vertical distance, which is expressed in feet (ft) [3]: the vertical separation. We say that two aircraft are separated when the distance between their projections on a horizontal plane is higher than the standard horizontal separation, OR when the distance between their projections in a vertical plane is higher than the standard vertical separation.

Air traffic control consists of organizing the flow of traffic in order to ensure flight security (in terms of managing collision risks) and of improving the capacity of the route network used by aircraft.

Three different types of control may be identified based on the nature of the traffic:

- aerodrome control: management of the taxiing, takeoff and landing phases
- approach control: management of traffic in the stage before landing or after takeoff in the vicinity of an airport
- en route control: essentially concerns traffic during the cruise phase between airports.

Currently, around 8000 movements take place each day on French territory, representing a control workload which is too large for a single controller. The workload is distributed by dividing the airspace into several sectors, each covered by a control team. The number of sectors is thus determined by the capacity of a controller to manage N aircraft simultaneously (in practice, the average appears to be from 10 to 15 aircraft; when this limit is reached, the sector is said to be saturated). A control center brings together a set of sectors across a given geographical zone.

Air control organizations are responsible for the flow of traffic in their allocated airspace. The service to users must provide perfect security, but also the best possible flow rate. Within each sector, controllers keep each airplane separate from the

[1] Flight level: altitude reading from an altimeter referred to an isobar surface 1013 mb (expressed in hundreds of feet); thus, a difference of 5000 feet gives a FL of 50.

[2] 1NM=1852m or the length of a minute of an arc on a large terrestrial circle

[3] 1 ft = 0.3048m

rest of the traffic by issuing instructions to pilots. On a sector one controller ensures coordination with neighboring sectors and is responsible for the pre-detection of conflicts. Another controller, the radar controller, monitors the traffic, ensures conflict resolution and communicates with pilots.

Controllers are not solely responsible for maintaining traffic flow. En-route control forms part of a chain of successive filters of which each element attempts to improve traffic flow. Each filter has different objectives and manages distinct spaces and time frames. Broadly speaking, we may identify five levels of elements:

1. Long-term organization: the crudest filter. Its aim is not to avoid conflicts in the strictest sense, but to organize traffic at macroscopic level in the medium and long term (above 6 months). Examples of this include traffic orientation schemes, measures taken by the flight schedule committee, inter-center agreements or arrangements with the military allowing civil aviation to use their airspace in order to manage the Friday afternoon peak.

2. Short-term organization: this is often known as pre-regulation. It consists of organizing traffic for a day d the day before $(d-1)$ or the day before that $(d-2)$. Relatively precise data is available in this case:

   - known flight plans
   - the control capacity of each center based on the workforce available on the day d
   - the maximum aircraft flow which may enter a sector in a given time, known as the sector capacity
   - data from previous weeks and years. Air traffic is relatively repetitive: traffic for any given Monday will be very similar to the previous Monday; the days before Christmas are similar to the same period the previous year, etc. This allows us to predict where congestion will occur, the capacity needed to respond to the demand, or even more limiting measures which need to be taken.

   This filter does not only act at macroscopic level, organizing traffic flow based on the available capacity, but also on each airplane, managing takeoff slots [4]. This filtering was carried out at national level across Europe until 1995, when it was transferred to European level in order to improve coordination. Short-term organization is now the responsibility of the CFMU [5].

3. Real-time regulation: this filter organizes different flows with regard to the day's events. It consists of adjustment measures which take account of events from the day before, which may not be fully understood. Thus, transatlantic traffic is not well known at $d-1$ but much better information is available 3 to 6 hours before the arrival time. The fraction of available capacity reserved for pre-regulation can then be adapted. Additional airplanes may be sent into other sectors, or the number of non-transatlantic flights dealt with by the

[4] A takeoff slot is a time window during which the aircraft is authorized to take off

[5] Central Flow Management Unit

center may be increased if there are fewer transatlantic flights than initially expected. In the same way, unused time slots (due to delays, technical incidents etc.) may be re-allocated, or changes may be made to take account of weather conditions (for example inaccessible terrain). This role is generally filled by the FMP[6] of each center.

4. Tactical: this is the last filter in the chain of air traffic control, and consists of the action of a controller on their sector. The average time an airplane spends in a sector is around fifteen minutes. The visibility of the controller is slightly higher, as flight plans become available a few minutes before the aircraft enters the sector.

   Detection, and, moreover, resolution of conflicts are not automated. Controllers are therefore trained to recognize types of conflicts and apply known maneuvers in such cases. The controller can only prevent conflicts by altering airplane trajectories.

   In a control sector, the higher the number of aircraft, the more the control workload increases (in a non-linear manner). A limit exists after which the controllers in charge of a control sector are unable to accept additional aircraft, obliging these new aircraft to travel around the sector, moving through less charged neighboring sectors. In this case, the sector is said to be saturated. This critical state should be avoided, as it provokes a cumulative overloading phenomenon in preceding sectors which can back up as far as the departure airport. The saturation threshold is very difficult to estimate, as it depends on the geometry of routes traversing a sector, the geometry of the sector itself, the distribution of aircraft along routes, the performances of the control team, etc. One widely accepted threshold is fixed at 3 conflicts and 15 aircraft for a given sector. This maximum load should not last for more than ten minutes as it places the controllers under considerable stress, with the risk that they will no longer be able to manage traffic in optimal safety conditions.

   The control workload measurement is critical in many domains of ATM as it is at the heart of optimization processes. Examples include the following applications:

   - Airspace comparison (US/Europe).
   - Validation of future concepts (SESAR, NEXTGEN, etc.).
   - Analysis of traffic control action modes (situation before and after control).
   - Optimization of sectorization.
   - Optimization of sector grouping and de-grouping (pre-tactical alert: anticipation of an increase in congestion in a group of sectors in order to carry out degrouping in an optimal manner).
   - Optimization of traffic assignment.

[6] Flow Management Position

- Determination of congestion pricing zones.
- Organic control assistance tools.
- Generation of 4D trajectories.
- Prediction of congested zones.
- etc.

Based on the work of the second part a new airspace complexity metric has been developed and tested.

Part I

Background

Chapter 1

The Total Variation

The use of the Total Variation (TV) in image processing has become very popular after celebrated paper by Rudin, Osher and Fatemi [17]. In their work, the total variation is used as a regularization functional in a Tikhonov like model, and this was shown to produce reconstructions of image with preserved edges.

By contrast, we use the total variation not for reconstruction purpose, but rather for classification. Our images are given (so we never have to minimize the TV), and we actually compute their total variation only to analyze their features. The reader interested by the variational use of TV may read Appendix A, which outlines these applications.

In this chapter, we define the TV and the BV norm, and we discuss their numerical assessment in $2D$ images.

1.1 History of the BV norm

1.1.1 Basic definition

Total variation for functions of $n \geq 1$ real variables is defined as follows.

Definition 1.1.1 Let Ω be an open subset of $\mathbb{R}^n$. Given a function f belonging to $L^1(\Omega)$, the total variation of f in Ω is defined as

$$TV(f,\Omega) = \sup\left\{ -\int_\Omega f(\mathbf{x})\mathrm{div}\phi(\mathbf{x})d\mathbf{x} : \phi \in C_c^1(\Omega,\mathbb{R}^n), \left|\phi(\mathbf{x})\right| \leq 1 \text{ for } \mathbf{x} \in \Omega \right\}$$

where $C_c^1(\Omega,\mathbb{R}^n)$ is the set of continuously differentiable vector functions of compact support contained in Ω.

We can now define the space BV as

$$BV(\Omega) := \left\{ f \in L^1(\Omega) \middle| TV(f,\Omega) < +\infty \right\}.$$

Endowed with the norm $\|f\| = \|f\|_{L^1} + TV(f)$, this space is complete and is proper superset of $W^{1,1}(\Omega)$ [16].

If $f \in BV(\Omega)$, the total variation $TV(f)$ may be regarded as a measure, whose value on an open set $U \subset \Omega$ is

$$TV(f,U) = \sup\left\{ -\int_U f(\mathbf{x}) \mathrm{div}\phi(\mathbf{x}) d\mathbf{x} : \phi \in C_c^1(U,\mathbb{R}^n), |\phi(\mathbf{x})| \leq 1 \text{ for } \mathbf{x} \in U \right\}.$$

The total variation of a differentiable function f can be expressed as an integral involving the given function instead of as the supremum of the functionals of the above definitions. More concretely, we have the following result.

Theorem 1.1 *Given a C^1 mapping f defined on a bounded open set $\Omega \subseteq \mathbb{R}^n$, the total variation of f has the following expression*

$$TV(f,\Omega) = \int_\Omega |\nabla f(\mathbf{x})| \, \mathrm{d}\mathbf{x}$$

where $|.|$ denotes the L^2-norm.

This result relates the total variation to a standard L^2 norm when the mapping is smooth enough. To prove it, let $\phi \in C_c^1(U,\mathbb{R}^n)$. Using integration by parts it comes:

$$-\int_\Omega \mathrm{div}\phi(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x} = \int_\Omega \langle \phi(\mathbf{x}), \nabla f(\mathbf{x}) \rangle \mathrm{d}\mathbf{x}$$

Since Ω is bounded, ∇f and ϕ are both square summable mappings. Since the set of compactly supported continuous mappings is dense in $L^2(\Omega)$ and using the Cauchy-Schwartz inequality, it comes that:

$$\sup_{\phi \in C_c^1(U,\mathbb{R}^n)} \int_\Omega \langle \phi(\mathbf{x}), \nabla f(\mathbf{x}) \rangle dx = \int_\Omega |\nabla f(\mathbf{x})| \, \mathrm{d}\mathbf{x}.$$

1.1.2 Discretizations of the Total Variation of an image

The most commonly used version of discrete TV is

$$TV(u) = \sum_{i=1}^{N-1} \sum_{j=1}^{M-1} \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2} \Delta x \tag{1.1}$$

where $u = (u_{i,j})$ is the discrete image and Δx is the grid size.

In practice, let us consider that the domain Ω is square and define a regular $N \times N$ grid of pixels, indexed as (i,j), for $i = 1,2,\ldots,N$, $j = 1,2,\ldots,N$. We represent images as two-dimensional matrices of dimension $N \times N$, where $u_{i,j}$ represents the value of the function u at pixel (i,j). We denote by X the Euclidean space $\mathbb{R}^{N \times N}$. Then, the image u is a vector in X. Let us introduce the discrete gradient of $u \in X$, whose two components at each pixel (i,j) are defined as follows:

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N, \\ 0 & \text{if } i = N, \end{cases} \tag{1.2a}$$

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N, \\ 0 & \text{if } j = N. \end{cases} \tag{1.2b}$$

The discrete gradient operator

$$(\nabla u)_{i,j} = \left((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}\right)$$

is a linear map from X to $Y = X \times X$.

The total variation of u then is defined by

$$TV(u) = \sum_{1 \le i,j \le N} |(\nabla u)_{i,j}|\,, \tag{1.3}$$

with $|y| = \sqrt{y_1^2 + y_2^2}$ for every $y = (y_1, y_2) \in \mathbb{R}^2$.

1.2 Vector total variation norm

In this section, we introduce the generalization of scalar total variation (TV) to vector-valued functions $\mathbf{u} : \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ in such a way such that in the case $m = 1$ both definitions coincide. Attempts to define total variation for vector-valued functions fall into two classes. The first one is based on using the scalar TV on each of the m components of the mapping and to aggregate the vector obtained that way to end up with a single value. The second class is rooted in Riemann geometry [19].

In [35] Bresson and Chan introduced a vector TV definition belonging to the first class. Their work is briefly presented below.

Let us consider a mapping $\mathbf{u}\colon \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ where Ω is a bounded open set. The (component-wise) vector TV of $\mathbf{u}$ is defined as:

$$TV(\mathbf{u}) = \sup_{\mathbf{p} \in P} \left\{ -\int_\Omega < \mathbf{u}, \nabla \cdot \mathbf{p} > \mathrm{d}\mathbf{x} \right\} \tag{1.4}$$

with P the unit ball in the space $C^1_c(\Omega; \mathbb{R}^{m \times n})$ endowed with the L^2 (resp. L^∞) norm:

$$\|p\|_2 = \left(\sum_{i=1}^m |p_i|^2 \right)^{1/2}$$

resp.

$$\|p\|_\infty = \sup_{i=1\ldots m} |p_i|$$

Thus, the vector total variation norm (1.4) can be defined in two different ways, depending on the which norm is selected. As in the scalar case, the supremum can be expressed using an integral when the mapping u is of C^1 class. When the L^∞ norm is used, it comes:

$$TV(\mathbf{u}) = \sum_{i=1}^m \int_\Omega |Du_i| \mathrm{d}\mathbf{x} = \sum_{i=1}^m TV(u_i), \tag{1.5}$$

i.e the sum of the TV in each component. When the L^2 is selected, the result is modified as:

$$TV(\mathbf{u}) = \int_\Omega \sqrt{\sum_{i=1}^m |\nabla u_i|^2} \mathrm{d}\mathbf{x} = \int_\Omega \|\nabla \mathbf{u}\| \mathrm{d}\mathbf{x}. \tag{1.6}$$

In this case, the vector TV will not reduce to a sum of component wise TVs, but still has a very intuitive interpretation.

The approach based on Riemann geometry stems from the suggestion of Di Zenzo [36] to consider a vector-valued image as a parameterized 2-dimensional Riemann manifold in a nD-space. The metric tensor of this manifold is given by

$$g_{\mu\nu} = (\partial_\mu \mathbf{u}, \partial_\nu \mathbf{u}), \quad \mu, \nu = 1, 2. \tag{1.7}$$

Based on this framework, Sapiro [37] suggests a family of possible definitions for the vectorial TV, which is of the form

$$TV_{SR} := \int_\Sigma f(\lambda_+, \lambda_-) \mathrm{d}s, \tag{1.8}$$

where $\lambda_\pm$ denote the largest and smallest eigenvalue of $(g_{\mu\nu})$, respectively, and f is a suitable scalar-valued function. TV_{SR} is in general only defined for differentiable functions, although dual formulations exist for special cases that allow extensions to locally integrable functions.

A special case of the TV_{SR} (1.8) is the choice $f(\lambda_+, \lambda_-) = \sqrt{\lambda_+ + \lambda_-}$, which generalizes to the Frobenius norm of the derivative $D\mathbf{u}$

$$TV_F(\mathbf{u}) := \int_\Omega \|D\mathbf{u}(\mathbf{x})\|_F \mathrm{d}\mathbf{x}. \tag{1.9}$$

It turns out that a convenient dual formulation can be found, so that total variation of locally integrable mappings can be obtained:

$$TV_F(\mathbf{u}) = - \sup_{(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n) \in K_F} \left\{ \sum_{i=1}^{n} \int_\Omega u_i \mathrm{div}(\boldsymbol{\xi}_i) \mathrm{d}\mathbf{x} \right\} \tag{1.10}$$

with $K_F = C_c^1(\Omega, \mathbb{R}^{m \times n})$.

In [19], Goldluecke *et al.* introduced a natural generalization of the total variation to a vector-valued functions $\mathbf{u} : \Omega \to \mathbb{R}^m$, which concerns the geometry measure theory. It is given by the integral

$$TV_J(\mathbf{u}) := \int_\Omega J_1 \mathbf{u} \mathrm{d}\mathbf{x}, \tag{1.11}$$

where Jacobian J_1 is defined as the operator norm of $\nabla \mathbf{u}$.

The authors showed a result which relates the Jacobian to the singular values of the $\sigma_1(D\mathbf{u}), \ldots, \sigma_m(D\mathbf{u})$ of the derivative matrix $D\mathbf{u}$ in case of differentiable $\mathbf{u}$.

Proposition 1.2 *For functions $\mathbf{u} \in C^1(\Omega, \mathbb{R}^m)$, the vectorial total variation $TV_J(\mathbf{u})$ equals the integral over the largest singular value of the derivative matrix,*

$$TV_J(\mathbf{u}) = \int_\Omega \sigma_1(D\mathbf{u}) \mathrm{d}\mathbf{x}. \tag{1.12}$$

In particular, TV_J is equal to the standard total variation for real-valued functions.

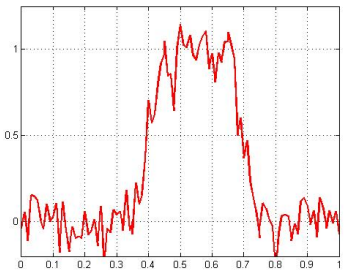
Vectorial total variation is also used in image processing, such as denoising, deblurring, super-resolution, inpainting, etc. The studies show that its use for image processing leads to a significantly better restoration of color images, both visually and quantitatively [19].

1.3 The total variation as a classification criterion

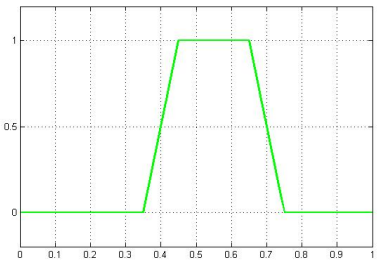
In the preceding section, we present primarily on the total variation and some of its application. In this section, we show how the total variation is used in our algorithms and how we calculate it in our experiment. The application of the TV in our method is different from the preceding methods. The calculation of the TV allows us to create a criterion in image classification.

1.3.1 The reason for choosing the TV in our methods

To answer the question why do we chose the TV in our methods, let's consider the TV of two functions given in the image as follows.



(a) The function with many moves up and down



(b) The function with fewer moves up and down

Figure 1.1: The function in the figure (b) has a much smaller Total Variation than it has in the figure (a).

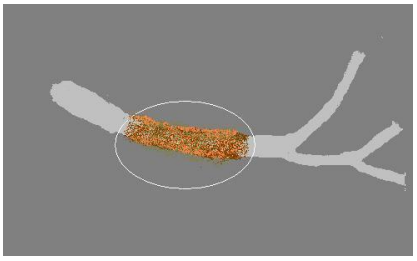
Intuitively, the TV is sum of all the "jumps" in the domain. The Figure 1.1 illustrates that if a function has more moves up and down, then its total variation will be higher. It is natural to relate this to our problem. We deal with an eye disease that is called Vasculitis in Multiple Sclerosis. We will describe it more detail in the following chapters. The data include the retinal images. The essence of the problem is that, intensity values of pixels that are at the diseased region have more changes around the vessel. As a consequence of this, if we calculate the TV along the vessel on the domain which is around each vessel segment, the abnormal region will give the higher value. Figure 1.2 demonstrates the difference between the normal vessel and the abnormal vessel.

So, the choice of total variation seems to be the best adapted to our problem.

1.3.2 Calculation of the TV in the thesis

To calculate the TV, at first, we need to extract vessels from the retinal image. We also need to get the centerline of vessels. It guarantees that the domain where we calculate the TV does not deviate from vessels. The calculation of the TV is described as follows.

Let us denote $u = (u_{i,j})_{i,j}$ an image. Firstly, as it is shown in Section 1.1.2, we compute at each pixel (i, j) two components of the discrete gradient $(\nabla u)_{i,j} =$



(a) An abnormal vessel of one retinal image.

(b) An normal vessel of one retinal image.

Figure 1.2: The image in figure (b) has a much smaller TV variation inside the marked region than the one in Figure (a).

$\big((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}\big)$.

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N, \\ 0 & \text{if } i = N, \end{cases}$$
$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N, \\ 0 & \text{if } j = N. \end{cases}$$

At each point P on the centerlines, the local total variation is given by

$$TV_D(P) = \sum_{(i,j)\in I} |\nabla u_{i,j}|$$

where I is the set of double indices corresponding to points in the disc of center P of domain D. In the context, we call it the BV norm of image u at point P. Figure 1.3 explains us how the BV norm is calculated.

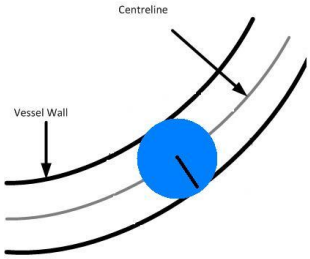


Figure 1.3: Computation of BV norm along the centerlines of one vessel

Chapter 2

The classification by SVM

2.1 Introduction

The general machine learning algorithm rely on some kind of search procedure: given a set of observations and a space of all possible hypotheses that might be considered (the "hypothesis space"), they look in this space for those hypotheses that best fit the data (or are optimal with respect to some other quality criteria). There are two classes of machine learning algorithms: supervised learning and unsupervised learning. This depends on whether the data is labeled or unlabeled. If the labels are given then the problem is one of supervised learning in that the true answer is known for a given set of data. The problem is one of classification or is one of regression which corresponds to the labels are categorical or real-valued, respectively. If the labels are not given, the problem is one of unsupervised learning and the aim is to characterize the structure of the data. The Support Vector Machines (SVMs) are a supervised learning method that generate input-output mapping functions from sets of labeled training data. From a set of labeled training data, SVMs generate input-output mapping functions that can be either a classification function, i.e., the category of the input data, or a regression function.

2.2 History

Support Vector Machines were introduced by Vladimir Vapnik and colleagues. The publication of the first papers by Vapnik, Chervonenkis and co-workers in 1964/65 went largely unnoticed till 1992 [2]. This was due to a widespread belief in the statistical and/or machine learning community that, despite being theoretically appealing, SVMs are neither suitable nor relevant for practical applications. They were taken seriously only when excellent results on practical learning benchmarks were achieved in digit recognition, computer vision and text categorization. Today, SVMs show better results than (or comparable outcomes to) Neural Networks (NNs) and other statistical models, on the most popular benchmark problems [2]. They have recently become an area of intense research owing to developments in the techniques and theory coupled with extensions to regression and density estimation.

SVMs arose from statistical learning theory; the aim being to solve only the

problem of interest without solving a more difficult problem as an intermediate step. SVMs are based on the structural risk minimization principle, closely related to regularization theory. This principle incorporates capacity control to prevent overfitting and thus is a partial solution to the bias-variance trade-off dilemma [3].

Statistical Learning Theory

Assume that the training data to be generated i.i.d (independent and identically distributed) from an unknown distribution $P(\mathbf{x}, y)$, the input points follow a probability distribution $P(\mathbf{x})$ and the output associated with a point $\mathbf{x}$ is $f(\mathbf{x})$. In fact, the available data are generated in the presence of noise so the observed values will be stochastic even if the underlying mechanism is deterministic. Thus, the distribution $P(\mathbf{x}, y)$ can be written as

$$P(\mathbf{x}, y) = P(\mathbf{x})\mathcal{N}_\sigma(f(\mathbf{x}) - y)$$

where $\mathcal{N}_\sigma$ is the distribution of noise, it is a Gaussian distribution with density $\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right)$.

The tasks of supervised learning can be formulated as the minimization of a loss function over a training set. The goal of estimation is to find a function that models its input well: if it were applied to the training set, it should predict the values (or class labels) associated with the samples in that set. The loss function quantifies the amount by which the prediction deviates from the actual values. The traditional loss functions are:

- $L(f(\mathbf{x}), y) = I_{f(\mathbf{x}) \neq y}$, where I is the indicator function: $I_A = 1 \Leftrightarrow A$ true (for classification)
- $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$ (for regression).

Value of the loss function is a random quantity because it depends on the outcome of a random variable. Decision rule involve making a choice using an optimal criterion based on the expected value of the loss function. The problem of supervised learning then is to minimize the expected loss,

$$R(f) = \int L(f(\mathbf{x}), y) dP(\mathbf{x}, y) \tag{2.1}$$

The expected loss of a function f is also called the risk.

Empirical Risk Minimization

As stated above, the learning problem is to find a function f, based on the available training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, such that f minimizes the risk $R(f)$. In practice, the expectation of the loss function cannot be computed since the distribution $P(\mathbf{x}, y)$ is unknown, and so cannot evaluate (2.1). However, we can

compute an approximation, called empirical risk, by averaging the loss function on the training set

$$R_{emp}(f) = \frac{1}{l}\sum_{i=1}^{l} L(f(\mathbf{x}_i), y_i).$$

A principled way to minimize true error is to upper bound in probability the true error and minimize the upper bound. This is the approach of statistical learning theory that lead to the formulation of the SVM [3]. We need to introduce the key concept of VC dimension named after Vapnik and Chervonenkis, which is a measure of complexity of a class $\mathcal{F}$ of functions. The SV machine creates a model with minimized VC dimension and when the VC dimension of the model is low, the expected probability of error is low as well. The method derived from VC theory is following.

VC theory

Definition 2.2.1 (VC dimension, [6]) The VC dimension h of a class of functions $\mathcal{F}$ is defined as the maximum number of points that can be learned exactly (shattered) by a function of $\mathcal{F}$,

$$h = \max\big\{|X|, X \subset \mathcal{X}, \text{ such that } \forall b \in \{-1,1\}^{|X|}, \exists f \in \mathcal{F} / \forall \mathbf{x}_i \in X, f(\mathbf{x}_i) = b_i\big\}.$$

Note that, if the VC dimension of a class of functions $\mathcal{F}$ is h, then there exists at least one set of h points that can be shattered by $\mathcal{F}$. This does not mean that all samples of size h are shattered by $\mathcal{F}$. Conversely, all samples S with cardinality $|S| > h$ are no longer shattered by $\mathcal{F}$. In order to show that the VC dimension is at most h, one must show that no sample of size $h+1$ is shattered.

Now lets take a look at an example of how we might calculate the VC-Dimension.

Example 1 This example demonstrates that the VC dimension of the class of separating hyperplanes in $\mathbb{R}^2$ is 3. Given 3 points in $\mathbb{R}^2$, then there are $2^3 = 8$ ways of assigning 3 points to two classes. As shown in Figure 2.1-(a) , all 8 possibilities can be realized using separating hyperplanes, i.e. the function class can shatter 3 points. It is not possible to shatter 4 points in $\mathbb{R}^2$ by hyperplanes (see Figure 2.1-(b)). Therefore, the maximum points in $\mathbb{R}^2$ that can be shattered by the function class is 3, in other words, the VC dimension is 3.

The result in the Example 1 can be generalized by the following theorem.

Theorem 2.1 (VC dimension of hyperplanes) [10, 7] *Let $\mathcal{F}$ be the set of hyperplanes in $\mathbb{R}^n$,*

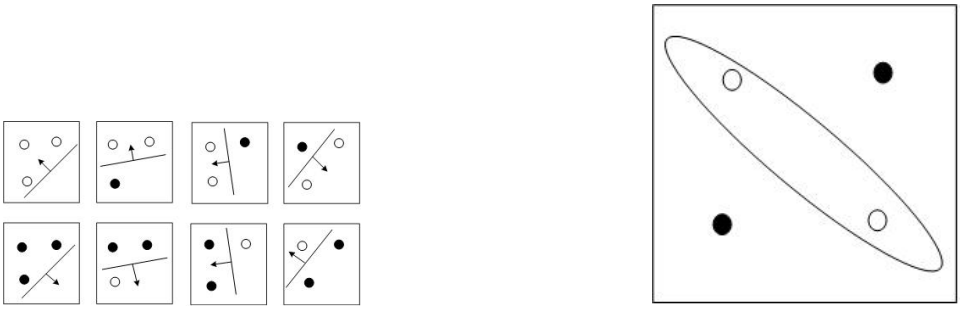
$$\mathcal{F} = \{\mathbf{x} \mapsto sign(\mathbf{w}\cdot\mathbf{x} + b), \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}\}.$$

The VC dimension of $\mathcal{F}$ is $n+1$.

Here, the term $\mathbf{w}\cdot\mathbf{x}$ is dot product in Hilbert space $\mathbb{R}^n$. It can be defined as follows:

Let $\mathbf{w}$, $\mathbf{x} \in \mathbb{R}^n$ and suppose that $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, then

$$\mathbf{w}\cdot\mathbf{x} = w_1x_1 + w_2x_2 + \cdots + w_nx_n. \tag{2.2}$$



(a) (b)

Figure 2.1: A linearly separable case and a non-linearly separable case.
(a) the class of separating hyperplanes can shatter set of 3 points in $\mathbb{R}^2$. (b) the class of separating hyperplanes can not shatter set of 4 points in $\mathbb{R}^2$.

After having the concept of VC dimension, we now introduce the main theorem in VC theory that provides the bounds on the test error.

Theorem 2.2 [5] *Let $\mathcal{F}$ be a class of functions of VC dimension h. Then for any distribution P and for any sample $(\mathbf{x}_i, y_i)$ $1 \leq i \leq l$ drawn from this distribution, the following inequality holds true*

$$P\left\{\sup_{f\in\mathcal{F}} |R(f) - R_{emp}(f)| > \varepsilon\right\} < 4\exp\left\{h\left(1+\log\left(\tfrac{2l}{h}\right)\right) - \left(\varepsilon - \tfrac{1}{l}\right)^2 l\right\}.$$

Note that, for infinite training data, the law of large number assures,

$$\forall f, R_{emp}(f) \xrightarrow[l\to\infty]{} R(f).$$

However, in general, there is no guarantee for a solution based on the expected risk minimization.

The following result about upper bound which leads to the idea of structural risk minimization, is very important in learning theory.

Theorem 2.3 [6] *Let $\mathcal{F}$ be a class of functions of VC dimension h, then for any distribution P and for any sample $(\mathbf{x}_i, y_i)$, $1 \leq i \leq l$ drawn from this distribution, with probability $1-\eta$, the following inequality holds*

$$\forall f \in \mathcal{F}, R(f) \leq R_{emp}(f) + \sqrt{\left(\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}\right)} \tag{2.3}$$

They called the right hand side of Eq (2.3) the "risk bound". The second term on the right hand side is called the "VC confidence".

To separate correctly all training examples (permit $R_{emp}(f)$ get small value), the machine will necessarily require a large VC dimension h. Therefore, the VC confidence, which increases monotonically with h, will be large, and the bound (2.3) will show that the small training error does not guarantee a small test error. (A counterexample given by E. Levin, J.S. Denker [6] shows that VC dimension is infinite but $R_{emp} = 0$).

In [7] Christopher J.C. Burges noted some key points about the risk bounds as follows.

- It is independent of $P(\mathbf{x}, y)$. It assumes only that both the training data and the test data are drawn independently according to some $P(\mathbf{x}, y)$.
- It is usually not possible to compute the left hand side.
- If we know h, we can easily compute the right hand side. Thus given several different learning machines, and choosing a fixed, sufficiently small η, by then taking that machine which minimizes the right hand side, we are choosing that machine which gives the lowest upper bound on the actual risk. This gives a principled method for choosing a learning machine for a given task, and is the essential idea of structural risk minimization. Given a fixed family of learning machines to choose from, to the extent that the bound is tight for at least one of the machines, one will not be able to do better than this. To the extent that the bound is not tight for any, the hope is that the right hand side still gives useful information as to which learning machine minimizes the actual risk. The bound not being tight for the whole chosen family of learning machines gives critics a justifiable target at which to fire their complaints. At present, for this case, we must rely on experiment to be the judge.

2.3 Basic approach

2.3.1 Linear separation and Non-linear separation

In the SVMs models, there are a linearly separable case and a non-linearly separable case. For the linearly separable case, the given training data is linearly separable. A data set is linearly separable if they can be completely separated by a hyperplane. The Theorem 2.1 shows one result concerning the linearly separable case. This is the simplest of SVMs because it permits us to find easily the linear classifier. Linear classifiers sometimes aren't complex enough. Actually, real data almost are non-linearly separable. The kernel functions are then introduced in order to construct non-linear decision surfaces. We shall express this in Section 2.4. In the last section of the chapter, for noisy data, when complete separation of the two classes may not be desirable, slack variables are introduced to allow for training errors.

2.3.2 Maximal Margin Hyperplane

For a labeled training data $\{\mathbf{x}_i, y_i\}, i = 1, \ldots, l, y_i \in \{-1, 1\}, \mathbf{x}_i \in \mathbb{R}^n$. Suppose the data is linearly separable and two-class. (We can easily extend to k-class classification by constructing k two-class classifiers). We need to find a function in family $\mathcal{F}$ of linear function $f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ to perform separation. Recall that, if $\mathbf{w} \neq \mathbf{0}$, the equation $f_{\mathbf{w},b}(\mathbf{x}) = 0$ is that of an hyperplane in $\mathbb{R}^n$. The decision functions are then given by

$$\operatorname{sgn}(\mathbf{w} \cdot \mathbf{x} + b). \tag{2.4}$$

The vector $\mathbf{w} \in \mathbb{R}^n$ is termed the weight vector. The scalar b is termed the bias, it specifies the shift.

Since the data are linearly separable, there exist many hyperplanes which can separate the training data (see Figure 2.2 -(a)). The SVM will find the optimal separating hyperplane that has the largest margin. (Margin is the distance from the nearest point to the hyperplane).

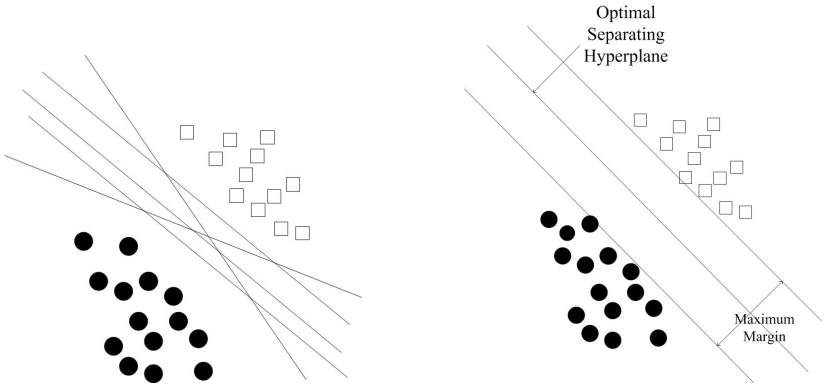


Figure 2.2: Choosing the optimal separating hyperplane.
(a) Many hyperplanes correctly separate the training examples. (b) The optimal separating hyperplane which has the largest margin is less sensitive to the noise in the data.

It is based on two facts ([9], page 11). First, among all hyperplanes separating the data, there exists a unique optimal hyperplane, distinguished by the maximum margin of separation between any training point and the hyperplane. It is the solution of

$$\rho = \max_{\mathbf{w},b} \min_{1 \le i \le l} \{\|\mathbf{x} - \mathbf{x}_i\| | \mathbf{x} \in \mathbb{R}^n, \mathbf{w} \cdot \mathbf{x} + b = 0\}. \tag{2.5}$$

Second, the capacity (as discussed in Section 2.2) of the class of separating hyperplanes decreases with increasing margin. Hence there are theoretical arguments supporting the good generalization performance of the optimal hyperplane. Deriving from the structural risk minimization theory ([5], page 430), the maximal margin minimizes the following error bound function:

$$R = \frac{D^2}{\rho^2},$$

where D is the radius of the smallest sphere that contains training vectors. In addition, it is computationally attractive, since we will show below that it can be constructed by solving a quadratic programming problem for which efficient algorithms exist.

Intuitively, the large margin brings safety for separation of a new data. We find the optimal separating hyperplane, which locates in the "middle" of two classes. Furthermore, if we find the classifier that performs well over the training data, it is clear that it will give a good classification for new data.

We now return to the linear separation problem. We consider a pair $(\mathbf{w}, b)$ satisfying the following constrains:

$$\mathbf{w} \cdot \mathbf{x}_i + b > 0 \quad \text{for } y_i = +1, \tag{2.6}$$
$$\mathbf{w} \cdot \mathbf{x}_i + b < 0 \quad \text{for } y_i = -1. \tag{2.7}$$

For such a pair, clearly $\mathbf{w} \neq \mathbf{0}$. These can be combined into one set of inequalities:

$$\forall i \in \{1, \ldots, l\}, \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0. \tag{2.8}$$

There is no loss of generality, we can scale the pair $(\mathbf{w}, b)$ so that

$$\min_{1 \leq i \leq l} \|\mathbf{w} \cdot \mathbf{x}_i + b\| = 1. \tag{2.9}$$

The constrains (2.8) the become

$$\forall i \in \{1, \ldots, l\}, \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{2.10}$$

Let $\mathbf{x}_1, \mathbf{x}_2$ be two training samples closest to the hyperplane from each side ($y_1 = +1$, $y_2 = -1$), we have $\mathbf{w} \cdot \mathbf{x}_1 + b = 1$ and $\mathbf{w} \cdot \mathbf{x}_2 + b = -1$ (see Figure 2.3). Therefore, the distances d_1 (resp. d_2) from $\mathbf{x}_1$ (resp. $\mathbf{x}_2$) to the hyperplane are given by

$$d_1 = \frac{|\mathbf{w} \cdot \mathbf{x}_1 + b|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}, \tag{2.11}$$

$$d_2 = \frac{|\mathbf{w} \cdot \mathbf{x}_2 + b|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}. \tag{2.12}$$

Hence, $d_1 = d_2 = 1/\|\mathbf{w}\|$ and the margin is simply $2/\|\mathbf{w}\|$. Thus we can find the

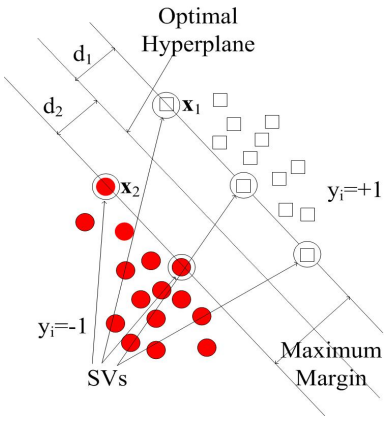


Figure 2.3: The Optimal Separating Hyperplane and Support vectors

pair of hyperplanes which gives the maximum margin by minimizing $\|\mathbf{w}\|^2$ subject to constraints (2.10). The optimization is now a convex quadratic programming (QP) problem:

$$(P) \quad \left| \begin{array}{l} \min\limits_{\mathbf{w},b} \ \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1, \quad i = 1, \ldots, l. \end{array} \right.$$

The problem (P) is dealt with by solving its dual problem, derived from introducing Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian,

$$L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1\right]. \tag{2.13}$$

This problem can be solved either in a primal space (which is the space of parameters $\mathbf{w}$ and b) or in a dual space (which is the space of Lagrange multipliers α_i). But we will switch to a Lagrangian formulation of the problem. There are two reasons for doing this [7]. The first is that the constraints (2.9) will be replaced by constraints on the Lagrange multipliers themselves, which will be much easier to handle. The second is that in this reformulation of the problem, the training data will only appear (in the actual training and test algorithms) in the form of dot products between vectors. This is a crucial property which will allow us to generalize the procedure to the nonlinear case (Section 2.4). We find the saddle point $(\mathbf{w}_0, b_0, \boldsymbol{\alpha}_0)$ because Lagrangian L_P must be minimized with respect to $\mathbf{w}$ and b, and has to be maximized with respect to non-negative α_i. Here, the Karush-Kuhn-Tucker (KKT) conditions are used for the optimum of a constrained function. (We will introduce Karush-Kuhn-Tucker (KKT) conditions in the Appendix B). Since our problem is a convex quadratic programming problem, the KKT conditions are necessary and sufficient conditions for a maximum of (2.13). The KKT conditions are stated as follows

- at the saddle point, derivatives of Lagrangian L_P with respect to primal variables should vanish, which yields

$$\begin{aligned} \nabla L_P(\cdot, b, \boldsymbol{\alpha})(\mathbf{w}) = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i = 0 \\ \Leftrightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \end{aligned} \tag{2.14}$$

$$\begin{aligned} \frac{\partial}{\partial b} L_P(\mathbf{w}, \cdot, \boldsymbol{\alpha})(b) = -\sum_{i=1}^{l} \alpha_i y_i = 0 \\ \Leftrightarrow \sum_{i=1}^{l} \alpha_i y_i = 0. \end{aligned} \tag{2.15}$$

- at the saddle point the products between dual variables and constraints equals zero (the KKT complementarity conditions) that mean the saddle point satisfies

$$\alpha_i \left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1\right] = 0 \quad i = 1, \ldots, l. \tag{2.16}$$

Substituting equations (2.14) and (2.15) into (2.13), the primal Lagrangian $L(\mathbf{w}, b, \boldsymbol{\alpha})$ changes to dual variables Lagrangian $L_D(\boldsymbol{\alpha})$

$$\begin{aligned}
L_D(\boldsymbol{\alpha}) =& \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right] \\
=& \frac{1}{2}\left(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i\right)\left(\sum_{j=1}^{l} \alpha_j y_j \mathbf{x}_j\right) + \sum_{i=1}^{l} \alpha_i - \sum_{i=1}^{l} \alpha_i y_i b \\
& - \sum_{i=1}^{l}\left[\left(\alpha_i y_i \mathbf{x}_i\right)\left(\sum_{j=1}^{l} \alpha_j y_j \mathbf{x}_j\right)\right] \\
=& \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\
=& \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \qquad (2.17)
\end{aligned}$$

Thus, instead of solving primal problem (P), we can solve following dual problem

$$(D) \left| \begin{array}{l} \max\limits_{\boldsymbol{\alpha}} \ \sum\limits_{i=1}^{l} \alpha_i - \frac{1}{2}\sum\limits_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} \quad \sum\limits_{i=1}^{l} \alpha_i y_i = 0, \\ \qquad\qquad \alpha_i \geq 0, \quad i = 1, \ldots, l. \end{array} \right.$$

We can express it in a matrix notation as follows:

$$(D) \left| \begin{array}{l} \max\limits_{\boldsymbol{\alpha}} \ \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{H} \boldsymbol{\alpha} \\ \text{subject to} \quad \mathbf{y}^\top \boldsymbol{\alpha} = 0 \\ \qquad\qquad \boldsymbol{\alpha} \geq 0, \end{array} \right.$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_l]^\top$, $\mathbf{H}$ denotes the Hessian matrix ($\mathbf{H}_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j = y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$) and $\mathbf{1}$ is the vector $\mathbf{1} = [1 \quad 1 \ldots 1]^\top$.

Suppose that $\boldsymbol{\alpha}_0 = [\alpha_1^0, \ldots, \alpha_l^0]^\top$ is the solution of the dual problem (D), then from Equation (2.14), we have the optimal weight vector

$$\mathbf{w}_0 = \sum_{i=1}^{l} \alpha_i^0 y_i \mathbf{x}_i. \qquad (2.18)$$

Since $\alpha_i^0 \geq 0$, $y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 \geq 0$ for $\forall i = 1, \ldots, l$ and from Equation (2.16) it follows that if $\alpha_i^0 > 0$ with some i then $y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) = 1$. It means the primal points that correspond to this case are the closest points to the optimal hyperplane. These points play a crucial role, since they are the only points needed in the expression of the Optimal Separating Hyperplane. They are called support vectors to point out the fact that they "support" the expansion of $\mathbf{w}_0$.

The parameter b_0 can be obtained from Equation (2.16)

$$b_0 = y_i - \mathbf{w}_0 \cdot \mathbf{x}_i \tag{2.19}$$

for any support vector $\mathbf{x}_i$.

In practice, it is safer to average over all support vectors, as follows

$$b_0 = \frac{1}{|I|} \sum_I \left(y_i - \mathbf{w}_0 \cdot \mathbf{x}_i \right), \tag{2.20}$$

where $I = \{i \in \{1, \cdots, l\} | \mathbf{x}_i \text{ is a support vector}\}$. The optimal weight vector $\mathbf{w}_0$, is obtained in (2.18) as a linear combination of the training points and $\mathbf{w}_0$ (same as the bias term b_0) is calculated by using only the support vectors (SVs). All remaining samples in the training set are irrelevant. This is important when the data set to be classified are very large. The support vectors are generally just a small portion of all training data ($|I| \ll l$).

Once having the Optimal Separating Hyperplane, the problem of classifying a given test pattern $\mathbf{x}$ is determined by considering the *sign* of $\mathbf{w}_0 \cdot \mathbf{x} + b_0$.

By using (2.18) the decision function can be written as

$$f(\mathbf{x}) = \operatorname{sgn} \left(\sum_I \alpha_i^0 y_i \mathbf{x}_i \cdot \mathbf{x} + b_0 \right). \tag{2.21}$$

2.4 The approach with kernel

2.4.1 Idea

A linear classifier may not be the most suitable hypothesis for the two classes. If the training data is non-linearly separable, the linear classifiers presented in the previous sections are no longer suitable. The SVM can be used to learn non-linear decision functions by first mapping the data to some higher dimensional space and then constructing a separating hyperplane in this space. So, instead of trying to fit a non-linear model, one can map the problem from the input space to a new higher-dimensional space by doing a non-linear transformation using suitably chosen basis functions and then use a linear model in the new space. We call this new space the *feature space*. The idea is illustrated in Figure 2.4.

In the feature space - which can be very high dimensional - the data points can be separated linearly. An important advantage of the SVM is that it is not necessary to implement this transformation and to determine the separating hyperplane in the possibly very-high dimensional feature space. Instead, a kernel representation can be used, where the solution is written as a weighted sum of the values of a certain kernel function evaluated at the support vectors. This is explained below.

2.4.2 The learning algorithm for a nonlinear SV machine

In this part, we design the optimal separating hyperplane in a feature space. We again consider labeled training data $\{\mathbf{x}_i, y_i\}, i = 1, \ldots, l, y_i \in \{-1, 1\}, \mathbf{x}_i \in \mathbb{R}^n$.

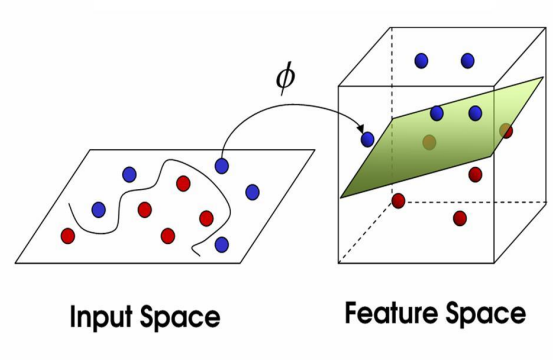


Figure 2.4: Input and feature spaces for the non-linearly separable case

This time, we suppose that, the data is non-linearly separable. We denote by Φ the mapping from the input space to the feature space:

$$\begin{aligned} \Phi : \mathbb{R}^n &\longrightarrow \mathcal{H} \\ \mathbf{x} &\longmapsto \Phi(\mathbf{x}). \end{aligned}$$

Here, $\mathcal{H}$ is a Hilbert space (i.e. a complete vector space endowed with an inner product, which may be a finite or infinite dimensional). The mapping $\Phi(\mathbf{x})$, which is typically non-linear mapping, is chosen in advance. By performing such a mapping, we hope that in some Hilbert space $\mathcal{H}$, our learning algorithm will be able to linearly separate images of $\mathbf{x}$ by applying the linear SVM formulation presented above.

There are two basic problems when mapping an input space into higher dimensional feature space [2]:

- the choice of a mapping $\Phi(\mathbf{x})$, that should result in a *rich* class of decision hypersurfaces,
- the calculation of the dot product $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, that can be computationally very discouraging if the dimension of feature space is very large.

For simplicity of the exposition, we assume below that $\mathcal{H}$ is finite dimensional¡ However, one should keep in mind that the forthcoming theoretical developments can easily be extended to the infinite dimensional case. The problem now is to find the optimal hyperplane separating linearly the data $\{\Phi(\mathbf{x}_i),\ y_i\}, i = 1, \ldots, l$, $y_i \in \{-1, 1\}$. The maximization of the margin in the feature space is stated as follows:

$$(P) \quad \left| \begin{array}{l} \min\limits_{\mathbf{w},b} \ \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } y_i \left(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b\right) \geq 1, \quad i = 1, \ldots, l. \end{array} \right.$$

The Lagrangian for this problem is

$$L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b\right) - 1\right]. \tag{2.22}$$

Again, we consider a solution in a dual space as given below by using standard conditions for an optimum of a constrained function

$$\nabla L_P(\cdot, b, \boldsymbol{\alpha})(\mathbf{w}) = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i) = 0$$

$$\Leftrightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i), \tag{2.23}$$

$$\frac{\partial}{\partial b} L_P(\mathbf{w}, \cdot, \boldsymbol{\alpha})(b) = -\sum_{i=1}^{l} \alpha_i y_i = 0$$

$$\Leftrightarrow \sum_{i=1}^{l} \alpha_i y_i = 0, \tag{2.24}$$

and the KKT complementarity conditions below:

$$\alpha_i \left[y_i \left(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b \right) - 1 \right] = 0 \quad i = 1, \ldots, l. \tag{2.25}$$

Similar to the previous section, we obtain the dual Lagrangian L_D

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \tag{2.26}$$

and the dual problem for primal problem (P) is written as

$$(D) \left| \begin{array}{l} \max\limits_{\boldsymbol{\alpha}} \sum\limits_{i=1}^{l} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ \text{subject to} \quad \sum\limits_{i=1}^{l} \alpha_i y_i = 0, \\ \qquad\qquad \alpha_i \geq 0, \quad i = 1, \ldots, l. \end{array} \right.$$

Instead of dealing in the Hilbert space $\mathcal{H}$ (which may have infinite dimension), we solve the dual problem (D) in l-dimensional space. Suppose that $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*]$ is a solution of the dual problem (D), then

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(\mathbf{x}_i), \quad \text{and} \quad b = y_i - \sum_{j=1}^{l} \alpha_j^* y_j \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i),$$

for any i with $\alpha_i^* \neq 0$, that means, $\mathbf{x}_i$ is a support vector.

The formula for decision function can be written in the form as follows

$$f(\mathbf{x}) = \operatorname{sgn} \left(\sum_{I} \alpha_i^* y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b \right), \tag{2.27}$$

where $I = \{i \in \{1, \cdots, l\} | \mathbf{x}_i \text{ is a support vector}\}$. It is a linear classifier in a feature space. It will create a nonlinear separating hypersurface in the original input space.

In some context, one call it the decision surface. Note that the input data appear in the decision function (2.21) only in the form of dot products $\mathbf{x}_i \cdot \mathbf{x}$, and in the decision function (2.27) only in the form of dot products $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})$ [3]. Since the data only appear in dot products we require a computable function that gives the value of the dot product in $\mathcal{H}$ without explicitly performing the mapping. Hence, we introduce the so-called *kernel function*:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \tag{2.28}$$

The kernel function allows us to construct an optimal separating hyperplane in the space $\mathcal{H}$ without explicitly performing calculations in this space. Instead of calculating dot products we will compute the value of K. Only K is needed in the training algorithm and the mapping Φ is never explicitly used. This requires that K be an easily computable function. The decision function (2.27) can be rewritten as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i \in I} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}_j) + b_0\right), \tag{2.29}$$

where the *scale bias* b_0 is given by

$$b_0 = y_i - \sum_{j=1}^{l} \alpha_j^* y_j K(\mathbf{x}_i, \mathbf{x}_j),$$

for any support vector $\mathbf{x}_i$.

Example 2 *Suppose our input data lie in* $\mathbb{R}^2$. *Let* $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, *the mapping* Φ *maps* $\mathbf{x}$ *to feature space is given below*

$$\begin{aligned} \Phi : \mathbb{R}^2 &\longrightarrow \mathcal{H} \\ \mathbf{x} &\longmapsto \Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2). \end{aligned}$$

With $\mathbf{x}' = (x_1', x_2')$ *then* $\Phi(\mathbf{x}') = (x_1'^2, \sqrt{2}x_1'x_2', x_2'^2)$, *the dot product in feature space:*

$$\begin{aligned} \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') &= x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2 x_2'^2 \\ &= (x_1x_1' + x_2x_2')^2 \\ &= (\mathbf{x} \cdot \mathbf{x}')^2. \end{aligned}$$

Therefore, we can calculate $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$ *without using the mapping* Φ *by using kernel function* $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^2$.

Example 3 *For* $\mathbf{x} = (x_1, x_2), \mathbf{x}' = (x_1', x_2') \in \mathbb{R}^2$. *Consider mapping*

$$\begin{aligned} \Phi : \mathbb{R}^2 &\longrightarrow \mathbb{R}^5 \\ \mathbf{x} &\longmapsto \Phi(\mathbf{x}) = (1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2), \end{aligned}$$

then

$$\begin{aligned} \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') &= (1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2) \cdot (1, x_1'^2, \sqrt{2}x_1'x_2', x_2'^2, \sqrt{2}x_1', \sqrt{2}x_2') \\ &= 1 + x_1^2x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 + 2x_1x_1' + 2x_2x_2'. \end{aligned} \tag{2.30}$$

Choosing kernel function $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^2$ *we have,*

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= (1 + \mathbf{x} \cdot \mathbf{x}')^2 \\ &= (1 + x_1 x_1' + x_2 x_2')^2 \\ &= 1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_2 x_1' x_2'. \end{aligned} \tag{2.31}$$

Combining (2.30) with (2.31), we obtain

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}'),$$

hence, we can use kernel $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^2$ *to instead of inner product* $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$ *in* $\mathbb{R}^5$.

The remaining problem is only now specification of the kernel function, the kernel should be easy to compute, well-defined and span a sufficiently rich hypothesis space. This will be presented in the next part.

2.4.3 Kernel

As mentioned above, we want to avoid the problems with the mapping Φ which maps the input space to feature space. The high dimensionality of $\mathcal{H}$- feature space makes it very expensive in terms of both memory and time to represent the feature vectors $\Phi(\mathbf{x}_i)$ corresponding to the training vectors $\mathbf{x}_i$. Moreover, it might be very hard to find the transformation Φ that separates linearly the transformed data. We will construct the kernel function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^n$ for given training data vectors in input space. It can be generally defined as follows:

Definition 2.4.1 A kernel is a symmetric function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}$ so that for all $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathcal{X}$: $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ where Φ is a (non-linear) mapping from the input space $\mathcal{X}$ into the Hilbert space $\mathcal{H}$.

Some conditions are imposed on K so that all optimization results for the SVM still hold. The kernel function $K(\mathbf{x}, \mathbf{x}')$ allow us to compute the value of the dot product in feature space without having to explicitly compute the map Φ.

It can also be useful to remember that the way in which the kernel was applied in designing an SVM can be utilized in all other algorithms that depend on the scalar product (e.g., in principal component analysis or in the nearest neighbor procedure) [2]. Any algorithm for vectorial data that can be expressed only in terms of dot products between vectors can be performed implicitly in the feature space associated with any kernel by replacing each dot product by a kernel evaluation.

Polynomial Kernel

Suppose we are given patterns $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ where most information is contained in the dth order products (so-called monomials) of entries x_i of $\mathbf{x}$

$$x_{i_1} \cdot x_{i_2} \cdot \cdots \cdot x_{i_d}, \tag{2.32}$$

where $i_1, i_2, \ldots, i_d \in \{1, \ldots, n\}$. Often, these monomials are referred to as product features. These features form the basis of many practical algorithms; indeed, there

is a whole field of pattern recognition research studying polynomial classifiers, which is based on first extracting product features and then applying learning algorithms to these features [9].

The following lemma is about polynomial approximation.

Lemma 2.4 Define Φ to map $\mathbf{x} \in \mathbb{R}^n$ to the vector $\Phi(\mathbf{x})$ whose entries are all possible dth degree ordered products of the entries of $\mathbf{x}$ ($\Phi : \mathbb{R}^n \longmapsto \mathbb{R}^{n^d}$). Then the corresponding kernel computing the dot product of vectors mapped by Φ is

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d. \tag{2.33}$$

Proof. The inner product in $\mathbb{R}^{N_{\mathcal{H}}}$ is given by

$$\begin{aligned}
\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') &= \sum_{i1=1}^{n} \sum_{i2=1}^{n} \cdots \sum_{id=1}^{n} (x_{i1} x_{i2} \cdots x_{id}) \cdot (x'_{i_1} x'_{i_2} \cdots x'_{i_d}) \\
&= \sum_{i_1=1}^{n} x_{i_1} x'_{i_1} \sum_{i_2=1}^{n} x_{i_2} x'_{i_2} \cdots \sum_{i_d=1}^{n} x_{i_d} x'_{i_d} \\
&= \left(\sum_{i=1}^{n} x_i x'_i \right)^d = (\mathbf{x} \cdot \mathbf{x}')^d.
\end{aligned}$$

□

This result shows that, in the case of a monomial linearization function, it is not necessary to explicitly map the vectors $\mathbf{x}$ and $\mathbf{x}'$ to the n^d dimensional linearization space to calculate the dot product of the two vectors in this space. It is enough to calculate the standard inner product in input space and take it to the power of d.

Every mapping Φ defines a kernel function via $K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$, but conversely, given a kernel K, which are the conditions for an implicit mapping to exist? We address this question in the following paragraph.

Reproducing Kernel Hilbert Space

In this part, we will construct the feature space from a given kernel. We now introduce some definitions as follows:

Definition 2.4.2 (Gram Matrix)Given a function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^n$ and patterns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m \in \mathcal{X}$, the $m \times m$ matrix G with elements

$$G_{ij} = K(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j}), \tag{2.34}$$

is called the Gram matrix (or kernel matrix) of K with respect to $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$.

Definition 2.4.3 A real $m \times m$ matrix M satisfying

$$\langle \boldsymbol{c}, M\boldsymbol{c} \rangle = \sum_{i,j=1}^{m} c_i c_j M_{ij} \geq 0, \tag{2.35}$$

for all $\boldsymbol{c} = (c_1, c_2, \ldots, c_m) \in \mathbb{R}^m$ is called *positive definite*. If strict inequality holds in (2.35) for every $\boldsymbol{c} \neq \mathbf{0}$, then M is said to be *strictly positive definite*.

Definition 2.4.4 A real and symmetric function $K(\cdot,\cdot)$, that is a function with the property $K(\mathbf{x},\mathbf{x}') = K(\mathbf{x}',\mathbf{x})$, is called a positive definite kernel (resp. strictly positive definite kernel) if for all choices of m points the corresponding Gram matrix K is positive definite (resp. strictly positive definite).

From now on, positive definite kernels will merely be called kernels.

With the definition of a positive definite kernel in mind, it is possible to construct a vector space, an inner product, and a linearization function Φ such that the kernel condition (2.28) is fulfilled. In [9], the authors constructed the feature space corresponding to a given kernel as follows.

We first need to define a vector space. Denote $\mathbb{R}^{\mathcal{X}}$ the space of functions mapping $\mathcal{X}$ into $\mathbb{R}$, $\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \longrightarrow \mathbb{R}\}$. Given a kernel K, define the *reproducing kernel feature map* $\Phi : \mathcal{X} \longrightarrow \mathbb{R}^{\mathcal{X}}$ as:

$$\begin{aligned} \Phi : \mathcal{X} &\longrightarrow \mathbb{R}^{\mathcal{X}} \\ \mathbf{x} &\longmapsto \Phi(\mathbf{x}) = K(\cdot,\mathbf{x}). \end{aligned}$$

Consider the vector space:

$$\mathcal{H}_K = \operatorname{span}\{\Phi(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = \left\{ f = \sum_{i=1}^{l} \alpha_i K(\cdot,\mathbf{x}_i) \;\middle|\; l \in \mathbb{N}, \mathbf{x}_i \in \mathcal{X}, \alpha_i \in \mathbb{R} \right\} \quad (2.36)$$

For $f = \sum_{i=1}^{l} \alpha_i K(\cdot,\mathbf{x}_i)$ and $g = \sum_{j=1}^{l'} \beta_j K(\cdot,\mathbf{x}'_j)$, the inner product of f and g is then defined by

$$\langle f, g\rangle := \sum_{i=1}^{l}\sum_{j=1}^{l'} \alpha_i\beta_j K(\mathbf{x}_i,\mathbf{x}'_j), \quad (2.37)$$

where $l, l' \in \mathbb{N}, \alpha_i, \beta_j \in \mathbb{R}$ and $\mathbf{x}_1,\ldots,\mathbf{x}_l,\mathbf{x}'_1,\ldots,\mathbf{x}'_{l'} \in \mathcal{X}$. We will show that (2.37) is an inner product in $\mathcal{H}_K$. Indeed,

1. Symmetry:

$$\langle f, g\rangle = \sum_{i=1}^{l}\sum_{j=1}^{l'} \alpha_i\beta_j K(\mathbf{x}_i,\mathbf{x}'_j) = \sum_{j=1}^{l'}\sum_{i=1}^{l} \beta_j\alpha_i K(\mathbf{x}'_j,\mathbf{x}_i) = \langle g, f\rangle.$$

2. Bilinearity: Note that

$$\begin{aligned} \langle f, g\rangle &= \sum_{i=1}^{l}\sum_{j=1}^{l'} \alpha_i\beta_j K(\mathbf{x}_i,\mathbf{x}'_j) \\ &= \sum_{i=1}^{l} \alpha_i \left(\sum_{j=1}^{l'} \beta_j K(\mathbf{x}_i,\mathbf{x}'_j) \right) \\ \Leftrightarrow \langle f, g\rangle &= \sum_{i=1}^{l} \alpha_i g(\mathbf{x}_i). \end{aligned} \quad (2.38)$$

Similarly,

$$\begin{aligned}\langle f, g\rangle &= \sum_{i=1}^{l}\sum_{j=1}^{l'} \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{x}'_j) \\ &= \sum_{j=1}^{l'} \beta_j \left(\sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x}'_j) \right) \\ \Leftrightarrow \langle f, g\rangle &= \sum_{j=1}^{l'} \beta_j f(\mathbf{x}'_j). \end{aligned} \tag{2.39}$$

Hence, with $\forall \alpha \in \mathbb{R}$, then $\alpha f = \sum_{i=1}^{l} \alpha\alpha_i K(\cdot, \mathbf{x}_i)$ and by combining with (2.38), we have

$$\langle \alpha f, g\rangle = \sum_{i=1}^{l} \alpha\alpha_i g(\mathbf{x}_i) = \alpha \left(\sum_{i=1}^{l} \alpha_i g(\mathbf{x}_i) \right) = \alpha\langle f, g\rangle$$

In other words, for any $f_1, f_2 \in \mathcal{H}_K$, from Equation (2.39) we deduce

$$\begin{aligned}\langle f_1 + f_2, g\rangle &= \sum_{j=1}^{l'} \beta_j \left(f_1(\mathbf{x}'_j) + f_2(\mathbf{x}'_j) \right) \\ &= \sum_{j=1}^{l'} \beta_j f_1(\mathbf{x}'_j) + \sum_{j=1}^{l'} \beta_j f_2(\mathbf{x}'_j) \\ &= \langle f_1, g\rangle + \langle f_2, g\rangle. \end{aligned}$$

3. Positive definiteness:

$$\langle f, f\rangle = \sum_{i,j}^{l} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \text{ with equality if and only if } f \equiv 0.$$

From equations (2.38) and (2.39), we can see that the inner product which is defined in (2.37), does not depend on the particular expansion of f and g (since the particular expansions of f and g may not be unique). Therefore, the definition of inner product in (2.37) is well-defined.

The only point remaining is completeness, the proof of this we can find in [9] (Schölkopf and Smola, 2002)).

So, we showed that, feature space $\mathcal{H}_K$ is a Hilbert space. Now, we define a linearization function $\Phi(\mathbf{x}) = K(\cdot, \mathbf{x})$, it is a space of functions over $\mathcal{X}$. Based on the definition of inner product, we derive the formula below

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}')\rangle = \langle K(\cdot, \mathbf{x}), K(\cdot, \mathbf{x}')\rangle = K(\mathbf{x}, \mathbf{x}'). \tag{2.40}$$

In summary, deriving from a positive definite kernel, we have already constructed a vector space, an inner product and a linearization function, such that the kernel condition (2.28) holds.

The following theorem, due to Mercer, characterizes kernels.

Theorem 2.5 (Mercer 1909) *Let $K(\mathbf{x}, \mathbf{x}')$ be a continuous symmetric function in $L_2(\mathcal{X}^2)$. Then, there exists a mapping Φ and an expansion*

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \Phi_i(\mathbf{x}) \cdot \Phi_i(\mathbf{x}') \tag{2.41}$$

if and only if, for any compact subset C and $g \in L_2(\mathcal{C})$,

$$\iint_{\mathcal{C}\times\mathcal{C}} K(\mathbf{x}, \mathbf{x}')g(\mathbf{x})g(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \geq 0. \tag{2.42}$$

To prove the Mercer's theorem, the following results are used.

Theorem 2.6 (Parseval's theorem) *If $\{\phi_n\}$ is an orthonormal system of a Hilbert space $\mathcal{H}$, then, for all $f \in \mathcal{H}$, $\sum_n \langle f, \phi_n \rangle^2 \leq \|f\|^2$. Equality holds for all $f \in \mathcal{H}$ if and only if $\{\phi_n\}$ is complete.*

Definition 2.4.5 Let $T : \mathcal{H} \to \mathcal{H}$ be a linear operator on a Hilbert space $\mathcal{H}$. It is said to be *self-adjoint* if, for all $f, g \in \mathcal{H}$, $\langle Tf, g \rangle = \langle f, Tg \rangle$. It is said to be *positive* (resp. *strictly positive*) if it is self-adjoint and, for all nontrivial $f \in \mathcal{H}$, $\langle Tf, f \rangle \geq 0$ (resp. $\langle Tf, f \rangle > 0$). It is called compact if T maps bounded subsets of $\mathcal{H}$ into precompact subsets of $\mathcal{H}$ (precompact subset is a subset whose closure is compact).

Definition 2.4.6 (Hilbert-Schmidt operator) Let $T : \mathcal{H} \to \mathcal{H}$ be an operator on a Hilbert space $\mathcal{H}$. We say that T is a Hilbert-Schmidt operator if there exists an orthonormal basis $\{e_n\}_{n=1}^{\infty}$ such that $\sum_{n=1}^{\infty} \|Te_n\|^2 < \infty$.

Theorem 2.7 *Hilbert-Schmidt operators are compact.*

Theorem 2.8 (Spectral theorem) *Let T be a compact self-adjoint linear operator on a Hilbert space $\mathcal{H}$. Then there exists in $\mathcal{H}$ an orthonormal basis $\{\phi_i\}_i$ consisting of eigenvectors of T. If λ_i is the eigenvalue corresponding to ϕ_i, then either the set $\{\lambda_i\}$ is finite or $\lambda_i \to 0$ when $n \to \infty$. In addition, $\max_{n\geq 1} |\lambda_n| = \|T\|$. If, in addition, T is positive, then $\lambda_i \geq 0$ for all $i \geq 1$, and if T is strictly positive, the $\lambda_n > 0$ for all $i \geq 1$.*

Let $\mathcal{C}$ be a fixed compact set. We introduce an integral operator which associates to kernel K as follows.

$$[T_K f](\mathbf{x}) = \int_{\mathcal{C}} K(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')d\mathbf{x}' \tag{2.43}$$

T_K is a Hilbert-Schmidt operator, since kernel $K \in L^2(\mathcal{C}\times\mathcal{C})$. Since K is continuous symmetric non-negative definite function, the operator T_K satisfies properties below.

- $T_K f$ is continuous $\forall f \in L_2(\mathcal{C})$.
- The map $K \longmapsto T_K$ is injective.
- T_K is Hilbert-Schmidt thus compact, symmetric and non-negative definite.

Now apply the spectral theorem for compact operators on Hilbert spaces to T_K to show the existence of an orthonormal basis $\{\phi_i\}_i$ of $L_2(\mathcal{C})$ and of a sequence $\{\lambda_i\}_i$ such that for all $i \in \mathbb{N}^*$

$$\lambda_i \phi_i(\mathbf{x}) = [T_K \phi_i](\mathbf{x}) = \int_{\mathcal{C}} K(\mathbf{x}, \mathbf{x}') \phi_i(\mathbf{x}') d\mathbf{x}'.$$

Observe that since T_K compact, $\lambda_i \longrightarrow 0$ and since T_K non-negative, $\lambda_i \geq 0$. If $\lambda_i \neq 0$, the eigenfunction, $\phi_i(\mathbf{x})$, is a continuous function.

Theorem 2.9 *Let $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ a continuous, symmetric, and positive semi-definite kernel. Let T_K be the corresponding operator via (2.43), and let λ_i be the ith eigenvalue of T_K, and ϕ_i the corresponding normal eigenfunction. Then $\{\sqrt{\lambda_i}\phi_i : \lambda_i > 0\}$ form an orthonormal system in $\mathcal{H}_K$.*

Here, $\mathcal{H}_K$ is the Hilbert space defined in (2.36).

Proof of Mercer's theorem [12]

Proof. "$\Longrightarrow$": assuming (2.41) to hold. We will prove that kernel $K(\mathbf{x}, \mathbf{x}')$ satisfies (2.42). Let

$$K_N(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{N} \Phi_i(\mathbf{x}) \cdot \Phi_i(\mathbf{x}').$$

We have, $K_N(\mathbf{x}, \mathbf{x}')$ converges uniformly to $K(\mathbf{x}, \mathbf{x}')$. This implies that, for all $g \in L_2(\mathcal{C})$

$$\begin{aligned}
\iint_{\mathcal{C}\times\mathcal{C}} K(\mathbf{x}, \mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} d\mathbf{x}' &= \iint_{\mathcal{C}\times\mathcal{C}} \lim_{N\to\infty} K_N(\mathbf{x}, \mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\
&= \iint_{\mathcal{C}\times\mathcal{C}} \lim_{N\to\infty} \sum_{i=1}^{N} \Phi_i(\mathbf{x}) \cdot \Phi_i(\mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\
&= \lim_{N\to\infty} \sum_{i=1}^{N} \iint_{\mathcal{C}\times\mathcal{C}} \Phi_i(\mathbf{x}) \cdot \Phi_i(\mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\
&= \lim_{N\to\infty} \sum_{i=1}^{N} \left[\int_{\mathcal{C}} \Phi_i(\mathbf{x}) g(\mathbf{x}) d\mathbf{x} \right]^2 \\
&\geq 0.
\end{aligned}$$

This proves necessary condition.

"$\Longleftarrow$": suppose kernel $K(\mathbf{x}, \mathbf{x}')$ is non-negative definite. Then there exists an orthonormal basis $\{\phi_i\}$ of $L_2(\mathcal{C})$ consisting of eigenfunctions of T_K. Let λ_i be the ith eigenvalue of T_K, and ϕ_i the corresponding normal eigenfunction. By Theorem 2.9, the sequence $\{\sqrt{\lambda_i}\phi_i\}_{i\geq 1}$ is an orthonormal system of $\mathcal{H}_K$. Let $\mathbf{x} \in \mathcal{X}$. The Fourier coefficients of the function $K(\mathbf{x}, \cdot) \in \mathcal{H}_K$ with respect to this system are

$$\langle \lambda_i \phi, K(\mathbf{x}, \cdot) \rangle_K = \sqrt{\lambda_i} \phi_i(\mathbf{x}),$$

where $\langle \cdot, \cdot \rangle_K$ is defined by (2.37). Then, by Parseval's theorem, we have

$$\begin{aligned}\sum_{i\geq 1} \lambda_i |\phi_i(\mathbf{x})|^2 &= \sum_{i\geq 1} |\sqrt{\lambda_i}\phi_i(\mathbf{x})|^2 \\ &= \sum_{i\geq 1} \langle \lambda_i \phi, K(\mathbf{x}, \cdot)\rangle_K^2 \leq \|K(\mathbf{x}, \cdot)\|_K^2 = K(\mathbf{x}, \mathbf{x}) < \infty. \end{aligned} \tag{2.44}$$

Now, by Cauchy-Schwarz inequality

$$\begin{aligned}\sum_{i=1}^{N} \lambda_i |\phi_i(\mathbf{x}) \cdot \phi_i(\mathbf{x}')| &\leq \left(\sum_{i=1}^{N} \lambda_i |\phi_i(\mathbf{x})|^2\right)^{\frac{1}{2}} \left(\sum_{i=1}^{N} \lambda_i |\phi_i(\mathbf{x}')|^2\right)^{\frac{1}{2}} \\ &\leq \max_{\mathbf{z}\in\mathcal{C}} \sum_{i=1}^{N} \lambda_i |\phi_i(\mathbf{z})|^2 \\ &\leq \max_{\mathbf{z}\in\mathcal{C}} K(\mathbf{z}, \mathbf{z}) < \infty,\end{aligned}$$

where the last inequality stem from (2.44). It follows that the series $\sum_{i\geq 1} \phi_i(\mathbf{x})\phi_i(\mathbf{x}')$ converges absolutely and uniformly to a kernel K_0 on $\mathcal{C} \times \mathcal{C}$.

Now we fix point $\mathbf{x} \in \mathcal{C}$. As a function in $L_2(\mathcal{C})$, $K(\mathbf{x}, \cdot)$ can be expanded into the orthonormal basis $\{\phi_i\}_i$:

$$\begin{aligned}K(\mathbf{x}, \cdot) &= \sum_{i\geq 1} \langle K(\mathbf{x}, \cdot), \phi_i \rangle_{L_2(\mathcal{C})} \phi_i(\cdot) \\ &= \sum_{i\geq 1} T_K(\phi_i)(\mathbf{x})\phi(\cdot) = \sum_{i\geq 1} \lambda_i \phi_i(\mathbf{x})\phi_i(\cdot) = K_0(\mathbf{x}, \cdot).\end{aligned}$$

Thus, as functions in $L_2(\mathcal{C})$, $K(\mathbf{x}, \cdot) = K_0(\mathbf{x}, \cdot)$. Since both are continuous functions therefore they must be equal on $\mathcal{C}$. It follows that for $(\mathbf{x}, \mathbf{x}') \in \mathcal{C} \times \mathcal{C}$,

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \cdot \phi_i(\mathbf{x}').$$

This implies that $K(\mathbf{x}, \mathbf{x}')$ corresponds to a dot product in ℓ_2, since $K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ with

$$\begin{aligned}\Phi : \mathcal{C} &\longrightarrow \ell_2 \\ \mathbf{x} &\longmapsto \left(\sqrt{\lambda_i}\phi_i(\mathbf{x})\right)_i\end{aligned}$$

□

Properties of Kernels

Suppose K_1, K_2 are kernels on $\mathcal{X}$, $a > 0$, $f : \mathcal{X} \longrightarrow \mathbb{R}$, $\phi : \mathcal{X} \longrightarrow \mathbb{R}^N$, and K_3 is a kernel on $\mathbb{R}^N$. Then these are all kernel functions on X:

1. $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}')$

2. $K(\mathbf{x}, \mathbf{x}') = aK_1(\mathbf{x}, \mathbf{x}')$

3. $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') \cdot K_2(\mathbf{x}, \mathbf{x}')$

4. $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) \cdot f(\mathbf{x}')$

5. $K(\mathbf{x}, \mathbf{x}') = K_3\big(\phi(\mathbf{x}), \phi(\mathbf{x}')\big)$.

Mercer's theorem tells us whether or not a prospective kernel is actually a dot product in some space $\mathcal{H}$, but it does not tell us how to construct Φ or even what $\mathcal{H}$ is. However, we can explicitly construct below the mapping for some kernels.

The functions below are the most commonly used kernels:

- homogeneous polynomials: $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}')^d, \quad d \in \mathbb{N}$
- inhomogeneous polynomials: $K(\mathbf{x}, \mathbf{x}') = (\beta\mathbf{x}^T\mathbf{x}' + r)^d$, $\beta, c > 0$, $d \in \mathbb{N}$
- radial basis functions (RBF): $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$, $\beta > 0$.
- sigmoid: $K(\mathbf{x}, \mathbf{x}') = \tanh(\beta\mathbf{x}^T\mathbf{x}' + r)$, $\beta > 0$.

Here, β, r, and d are kernel parameters.

We can show that, the feature space which associates with Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$ is infinite dimensional. Indeed, we express the Gaussian kernel as:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}\text{'}\|^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{2\sigma^2}\right) \exp\left(\frac{\mathbf{x} \cdot \mathbf{x}'}{\sigma^2}\right). \end{aligned} \tag{2.45}$$

The first two factors depend on $\mathbf{x}$ and $\mathbf{x}'$ separately. They are simply scalars based on the magnitude of each instance respectively. Applying the Taylor expansion for the third term around point $\mathbf{0}$, we have:

$$\exp\left(\frac{\mathbf{x} \cdot \mathbf{x}'}{\sigma^2}\right) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{\mathbf{x} \cdot \mathbf{x}'}{\sigma^2}\right)^k \tag{2.46}$$

We now only focus on the term $(\mathbf{x} \cdot \mathbf{x}')^k$. Call Φ_k the degree-k monomial feature mapping corresponding to homogeneous polynomial kernel $(\mathbf{x} \cdot \mathbf{x}')^k$, that is,

$$(\mathbf{x} \cdot \mathbf{x}')^k = \Phi_k(\mathbf{x}) \cdot \Phi_k(\mathbf{x}').$$

The lemma below gives the monomial feature mapping.

Lemma 2.10 [7, 8] *For $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ and $k \in \mathbb{N}$, the feature mapping of the degree-k monomial feature kernel function $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^k$ can be defined as:*

$$\Phi_k : \mathbb{R}^n \longrightarrow \mathbb{R}^{N_{\mathcal{H}}} \quad \textit{with } N_{\mathcal{H}} = \begin{pmatrix} n+k-1 \\ k \end{pmatrix} = \frac{(n+k-1)!}{k!(n-1)!}$$

$$\Phi_k^j(\mathbf{x}) = \sqrt{\frac{k!}{\prod_{i=1}^n (\alpha_i^j)!}} \prod_{i=1}^n x_i^{\alpha_i^j}, \tag{2.47}$$

where $\alpha^j = (\alpha_1^j, \cdots, \alpha_n^j) \in \mathbb{N}^n$ *runs over the set of all* n*-tuples in* $\mathbb{N}^n$ *such that*

$$\sum_{i=1}^n \alpha_i^j = k$$

Notice that we do not need to specify the order in which we take the α^j, but that this order needs to be used consistently.

Using the monomial feature mapping, the Gaussian kernel function (2.45) can be equivalently transformed as:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{2\sigma^2}\right) \left(\sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{\mathbf{x} \cdot \mathbf{x}'}{\sigma^2}\right)^k\right) \\ &= \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{2\sigma^2}\right) \left(\sum_{k=0}^{\infty} \frac{1}{\sqrt{k!}\sigma^k} \left(\Phi_k(\mathbf{x}) \cdot \Phi_k(\mathbf{x}')\right) \frac{1}{\sqrt{k!}\sigma^k}\right). \end{aligned}$$

Therefore, the infinite-dimensional feature mapping $\Phi_G : \mathbb{R}^n \longrightarrow \ell^2$ induced by the Gaussian kernel function for an instance $\mathbf{x}$ can be defined as

$$\Phi_G(\mathbf{x}) = \left\{ \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \frac{1}{\sqrt{k!}\sigma^k} \Phi_k(\mathbf{x}) \right\}_{k=0}^{\infty} \tag{2.48}$$

and $K(\mathbf{x}, \mathbf{x}') = \Phi_G(\mathbf{x}) \cdot \Phi_G(\mathbf{x}')$.

2.5 Soft Margin Hyperplane

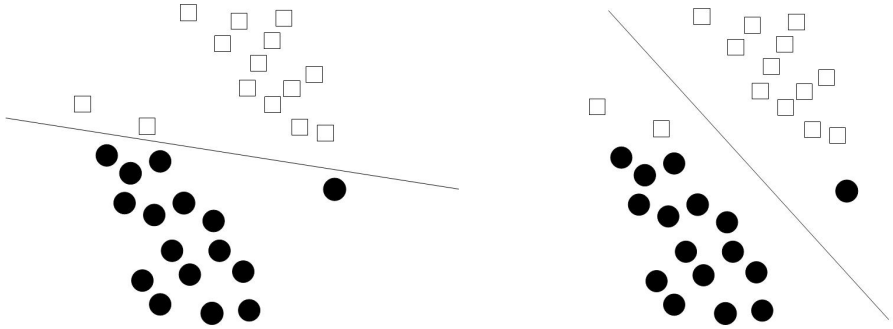
Until now only perfect data separation was considered, that means the data is noise-free. In practice such problems are very unlikely. In the case of noisy data, forcing zero training error will lead to poor generalization. The data points even can be linearly separated but the result is undesirable.

We do not need to correctly classify the noisy data points. By allowing some data points to be misclassified, we can get the larger margin solution (see Figure 2.5). We want to relax the constrains but only when necessary. This can be done by replacing the training rule $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ by

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{2.49}$$

with $\xi_i \geq 0$. The variables ξ_i are called slack variables.

Thus, for an error to occur, the corresponding ξ must exceed unity, so $\sum_i \xi_i$ is an upper bound on the number of training errors. Hence a natural way to assign an extra cost for errors is to change the objective function to be minimized from $\|\mathbf{w}\|^2/2$ to $\|\mathbf{w}\|^2/2 + C \sum_i \xi_i$, where C is a regularization parameter to be chosen by the user, a larger C corresponds to assigning a higher penalty to errors, this leads to



(a) (b)

Figure 2.5: The soft margin hyperplane should be used in the case of noisy data. (a) data can be linearly separated but it gives a very narrow margin. (b) the large margin solution violating some constrains is better.

small number of misclassifications, the bigger $\mathbf{w}^T\mathbf{w}$ and consequently to the smaller margin and vice versa.

Now, the optimization problem which find the separating hyperplane for the noisy data points can be stated as follows

$$(P) \left| \begin{array}{l} \min\limits_{\mathbf{w},b,\xi_i} \dfrac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{l}\xi_i \\ \text{subject to } y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \xi_i, \\ \qquad\qquad \xi_i \geq 0 \\ \qquad\qquad i = 1,\ldots,l. \end{array} \right. \tag{2.50}$$

The Lagrangian for this problem (2.50) can be written as

$$L_P(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i\left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{l}\beta_i\xi_i, \tag{2.51}$$

where α_i, β_i, $i = 1,\ldots,l$ are the Lagrange multipliers. The solution of this problem is the saddle point of the Lagrangian given by minimizing L_P with respect to $\mathbf{w}$, $\boldsymbol{\xi}$ and b, and maximizing with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Differentiating with respect to $\mathbf{w}$, b and $\boldsymbol{\xi}$ and setting the results equal to zero we obtain

$$\begin{aligned} \nabla L_P(\cdot, b, \boldsymbol{\alpha})(\mathbf{w}) &= \mathbf{w} - \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i = \mathbf{0} \\ \Leftrightarrow \mathbf{w} &= \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i \end{aligned} \tag{2.52}$$

$$\begin{aligned} \frac{\partial L_P}{\partial b} &= -\sum_{i=1}^{l}\alpha_i y_i = 0 \\ \Leftrightarrow \sum_{i=1}^{l}\alpha_i y_i &= 0, \end{aligned} \tag{2.53}$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$
$$\Leftrightarrow C = \alpha_i + \beta_i \tag{2.54}$$

and using the KKT complementarity conditions below,

$$\alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 + \xi_i \right] = 0 \quad i = 1, \ldots, l, \tag{2.55}$$
$$\beta_i \xi_i = 0 \tag{2.56}$$

Note that Equation (2.54) combined with Equation (2.56) shows that $\xi_i = 0$ if $\alpha_i < C$. Thus we can simply take any training point for which $0 < \alpha_i < C$ to use Equation (2.55) (with $\xi_i = 0$) to compute b. Substituting (2.52), (2.53) and (2.54) into (2.51) gives the following dual problem

$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.57}$$

We can establish soft-margin dual problem as

$$(D) \left| \begin{array}{l} \max\limits_{\boldsymbol{\alpha}} \sum\limits_{i=1}^{l} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} \quad \sum\limits_{i=1}^{l} \alpha_i y_i = 0, \\ \qquad\qquad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l. \end{array} \right. \tag{2.58}$$

The only difference from the separable case is that now the α_i have an upper bound of C. From these above conditions, we can conclude that non-zero (=active) slack variables can only be obtained for $\alpha = C$. The possible solutions for α_i are as follows (see Figure 2.6)

1. If $\alpha_i = 0$ and $\xi_i = 0$, data point is correctly classified.
2. If $0 < \alpha_i < C$, then $\xi_i = 0$. Thus $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$ and data point $\mathbf{x}_i$ is a support vector.
3. If $\alpha_i = C$, then $\xi_i \geq 0$. Data point $\mathbf{x}_i$ is also a support vector. It lies on the "wrong" side of the margin. For $0 \leq \xi_i < 1$, $\mathbf{x}_i$ is still correctly classified. And for $\xi_i \geq 1$, then $\mathbf{x}_i$ is misclassified.

Note that by setting C to infinity we can describe the "hard" margin with the formulate used for the soft margin.

As before, for $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*]$ is a solution of dual problem (D), then

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i^* y_i \mathbf{x}_i,$$

and

$$b^* = y_i(1 - \xi_i) - \sum_{j=1}^{l} \alpha_j^* y_j \mathbf{x}_j \cdot \mathbf{x}_i$$

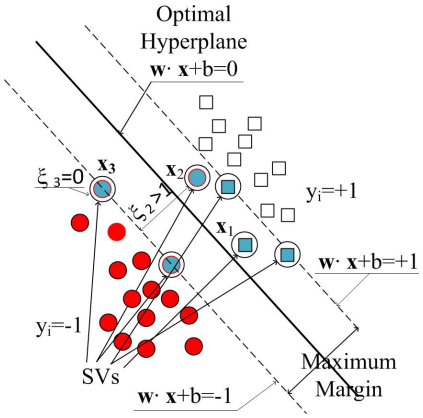


Figure 2.6: The Optimal Separating Hyperplane in the soft margin case.

for any $\alpha_i^* \neq 0$, in other way, $\mathbf{x}_i$ is a support vector.

The formula for decision function as below

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_I \alpha_i^* y_i \mathbf{x}_i \cdot \mathbf{x} + b^*\right), \tag{2.59}$$

where $I = \{i \in \{1, \cdots, l\} | \mathbf{x}_i \text{ is a support vector } \}$.

2.6 Soft margin Surface

We can use the kernel approach for the soft margin completely similar to the one for the "hard" margin in the previous section. We obtain the results below:

The primal problem

$$(P) \left| \begin{array}{l} \min\limits_{\mathbf{w},b,\xi_i} \dfrac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{l}\xi_i \\ \text{subject to } y_i\left(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b\right) \geq 1-\xi_i, \\ \qquad\qquad \xi_i \geq 0 \quad i=1,\ldots,l. \end{array} \right. \tag{2.60}$$

The Lagrangian for the primal problem (2.60) is

$$L_P(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i\left[y_i\left(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b\right) - 1 + \xi_i\right] - \sum_{i=1}^{l}\beta_i\xi_i. \tag{2.61}$$

The dual problem, in a matrix notation, reads

$$(D) \left| \begin{array}{l} \max\limits_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{1} - \dfrac{1}{2}\boldsymbol{\alpha}^\top Q\boldsymbol{\alpha} \\ \text{subject to} \quad \boldsymbol{\alpha}^\top\mathbf{y} = 0, \\ \qquad\qquad 0 \leq \alpha_i \leq C, \quad i=1,\ldots,l, \end{array} \right. \tag{2.62}$$

where $\mathbf{y} = [y_1, \ldots, y_l]^\top$, the matrix Q satisfies $Q_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ for kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$.

With $\boldsymbol{\alpha}^*$ a solution of problem (D), the decision function is written as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_I \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*\right). \tag{2.63}$$

where the bias term b^* is defined by

$$b^* = y_i(1 - \xi_i) - \sum_{j=1}^{l} \alpha_j^* y_j K(\mathbf{x}_j, \mathbf{x}_i)$$

for any support vector $\mathbf{x}_i$ corresponding to non-zero α_i.

Part II

Retinal image classification

Chapter 3

Classification of retinal images with the Vasculitis in Multiple Sclerosis

In the recent past, many research groups have developed methodologies and computer software for automated detection of retinal pathology. This is motivated by the increase of eye diseases (due to aging of the population worldwide) and made possible by the recent advance in computing power and image analysis.

Almost all of the studies focus on diabetic eye disease that is a complication from diabetes. The researchers concentrated on this field because diabetes is a significant and costly health problem in the Western world, and is growing in incidence at almost epidemic levels. Current prevalence of diabetes in the United States is 6.3% with greater prevalence in certain ethnic groups and socioeconomic classes [49][50]. Diabetic retinopathy is caused by increasing in blood sugar levels associated with diabetes, a progressive degenerative disease of the retina that has an asymptomatic stage that can start long before the onset of recognized diabetes. More recent evidence suggests that 40% to 45% of the diabetic population have some stage of diabetic retinopathy [51]. Diabetic retinopathy is divided into various stages and there are techniques to detect or to classify it for each stage.

Besides, there are some other eye pathologies that can lead to blindness. For instance, age-related macular degeneration, macular edema, glaucoma, etc. Herbert F. Jelinek and Michael J. Cree summarized one table on eye pathologies (see Table 3.1) [52]. These diseases are related or not related to diabetes.

As far as we know, there does not exist any method to detect and classify vasculitis with multiple sclerosis. In this part, we propose one algorithm for detecting and classifying this disease. The results from experiments show that the proposed algorithm is quite effective.

Table 3.1: List of Pathologies that Affect the Retina and the Possibility of Automated Detection Leading to Early Treatment

Retinal Pathology Screening Possible	Auto-detection Possible	Early Treatment Possible
Diabetic retinopathy	Yes	Yes
Refractive error in preschool children	Yes	Yes
Newborns for gonococcal eye disease (swab)		Yes
Retinopathy of prematurity	Yes	Yes
Hydroxychloroquine medication for retinophathy	Yes	Yes
Gluacoma	Yes	Yes
Age-related macular degeneration	Yes	Yes
Systemic atherosclerotic disease/ systemic hypertension	Yes	Yes

3.1 Problem description - the mathematical model

3.1.1 Patient population and data

Periphlebitis vasculitis and multiple sclerosis

Intermediate and posterior uveitis are the most frequent form of uveitis [40]. The potential risk of developing neurological disease is 60% in 5 years when there is a association of periphlebitis and optic neuropathy as 16% when there is only optic neuropathy [41]. This association is reported in 28% of the cases which are near the frequency of uveitis in Multiple Sclerosis (MS). Periphlebitis are histologically constituted by inflammatory cells in the edge of the retinal veins. In experimental autoimmune encephalopathy models, such periphlebitis were described in the area adjacent to the demyelinated lesions [42]. The authors hypothetised that there is an autoimmune reaction to a commune antigen [43].

Vasculitis

There are 3 types of vasculitis.
In the active form: vascular sheathing not well limited with irregular caliber of the vessels. This sheathing is situated at a distance from the optic nerve. It persists from few months to two years.
In the cicatricial form or venous sclerosis: the lesions are fixed, linear, distributed by segments and situated in the medium and peripheral retina. Some have a variable constriction of the vascular diameter.
In the third form, there is a reduction of the venous caliber.

Intermediate uveitis and pars-planitis

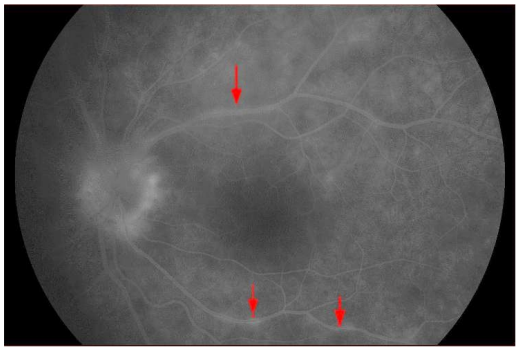
Gross pathologic examination of the peripheral *snow bank* in parsplanitis shows exudate deposited on the peripheral retina and parsplana. The histology reveals a collapsed vitreous, blood vessels, fibroglia cells including fibrous astrocytes, and scattered inflammatory cells. Peripheral veins show lymphocytic cuffing and infiltration. The vascular component of the snowbank is continuous with the retina [44]. Pars planitis is a primary peripheral perivascularitis and not a choroiditis. Once begun, the process of vascular occlusion may lead to vitritis and snowbank formation and inflammation of the adjacent tissue area by breakdown of the blood-ocular barrier [45][46].

Manifestation of the disease on the images

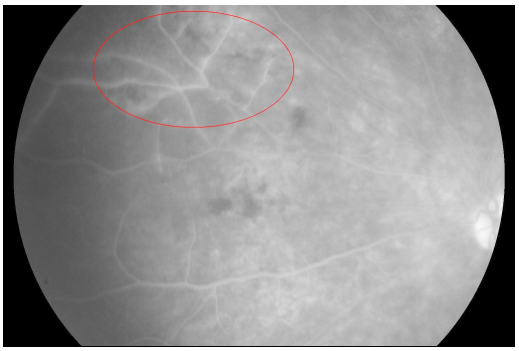
The term of vasculitis means that inflammation of blood vessels appears as an obvious clinical manifestation. The ophthalmoscopic characteristic sign of a vasculitis is the sheathing of blood vessels: the light is decreased, vessel walls are thickened and they take on a yellowish-white. The histological analysis of the concerned blood vessels shows, in the chronic phase, an accumulation of white cells, mainly lymphocytes. A partial restriction or complete blockage of the light of the blood vessel can be observed. During the early phase of obstructive phenomena, a diffusion of plasma and bleeding in the perivascular retinal tissue may occur. A vitreous hemorrhage may also be present as a sign associated. Retinal vasculitis may manifest principally as a phlebit (inflammation of inner and outer walls of the concerned vein), as periphlebitis (the outer wall of the vein is the main site of inflammation) as a capillaritis (inflammation of the capillary bed) or as one combination of various aforesaid suffering. Ocular vasculitis is a systemic disease or intraocular infection as an isolated intraocular inflammation is infectious, but also appears as an isolated manifestation primary intraocular. The sheathing of vessels is one of the earliest signs of retinal vasculitis. It may be the only evidence that demonstrates a start of an inflammatory process at the wall of vessels. The vessels are plotted with red color. In the occurrence of sheating, those vessels have yellow-white edges and higher brightness level. The following pictures illustrate the difference between normal images and abnormal images. In the Figure 3.1 (a), the sheathing is demonstrated by red arrow whereas the Figure 3.1 (b), the diseased region occurs in the red circle. The Figure 3.2 illustrates two normal retinal images.

Methods

Patients are registered at the Toulouse clinic ophthalmological and neurological department between 2000 and 2011. All started showing only Uveitis. The test group were 30 patients later diagnosed with MS and 5 patients with a presumed SEP (uncompleted criterias). The SUN (Standardization of Uveitis Nomenclature) classification was used for intermediate uveitis [47] and for neurological MS the McDonald criteria 2001 revised 2005 was used [48]. A second control group of patients with normal angiography was selected from patients with other autoimune diseases (sarcoïdosis, Behcet's syndrome , IBD (Inflammatory bowel disease)...).

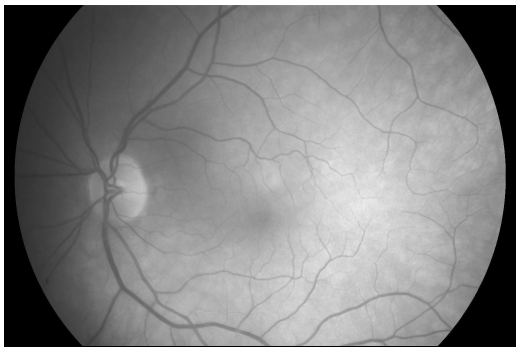


(a) A picture illustrating sheathing of vessel edges, which is shown by red arrows.

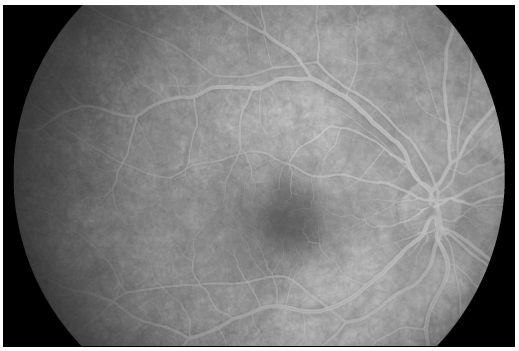


(b) One abnormal retinal image, the diseased region occurs inside the red circle.

Figure 3.1: Two pictures presenting the disease



(a) One normal image, the vessels are dark



(b) One normal retinal image, the background is dark.

Figure 3.2: Two pictures displaying normal retinal images

3.1.2 The mathematical model

In this section, we describe the mathematical model for the proposed problem. Our data includes a set $N = 49$ retinal images, we denote S. Each image is presented as a matrix of scalars (gray-scale image). We call such matrix representation $u_k = (u_k(i,j))_{i,j}$ with $1 \leq i \leq I_k, \quad 1 \leq j \leq J_k, \quad I_k, J_k \in \mathbb{N}^*$. With each observation, our expert associated it with one category to which it belongs. The response variable y_k of image u_k takes values in the categorical domain $\{-1, 1\}$. If one example u_k is diseased image, we mark $y_k = 1$, otherwise, it is marked by $y_k = -1$. The data are randomly separated into two sets: a training set, which is used for developing rules for the classification of future observations, and a test set that is used for determining the validity of the model. We use S_1 and S_2 to denote the training set and the test set, respectively. The training set S_1 includes 14 diseased images, and 10 normal images. The test set S_2 includes 13 diseased images and 12 normal images. The training set $\{(u_1, y_1), (u_2, y_2), \ldots, (u_l, y_l)\}$ will allow for the future prediction of the response variable y based on the observation of u only. Here, $l = 24$ is the number of examples in the training set.

To classify, the most important task is the extraction of the features that are used for classification. Feature extraction is the process of extracting significant

information from an image. We would like to create a set of features which helps us preserving or improving the discriminative ability of a classifier. To get features for classification, we will find a mapping F, which can be written in the following form:

$$\begin{aligned} F : S &\longrightarrow \mathbb{R}^m \\ \mathrm{u}_k &\longmapsto \mathbf{x}_k = F(\mathrm{u}_k) \in \mathbb{R}^m. \end{aligned}$$

The feature extraction is described in Sections 3.3 and 3.4.

After extracting the features, the original sample now becomes: $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$ with $\mathbf{x}_i \in \mathbb{R}^m$. Applying SVM theory, using the approach with kernel, the remaining task now is that of solving the optimization problem

$$(P) \quad \left| \begin{array}{l} \min\limits_{\mathbf{w},b} \ \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } y_i \left(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b\right) \geq 1, \\ \qquad\qquad i = 1, \ldots, l. \end{array} \right.$$

Here, Φ is the implicit mapping associated with the kernel function K, $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$.

Then, the corresponding dual problem is

$$(D) \quad \left| \begin{array}{l} \max\limits_{\boldsymbol{\alpha}} \ \displaystyle\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ \text{subject to} \quad \displaystyle\sum_{i=1}^{l} \alpha_i y_i = 0, \\ \qquad\qquad \alpha_i \geq 0, \quad i = 1, \ldots, l. \end{array} \right.$$

Suppose that $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*]$ is a solution of the dual problem (D). Then, the decision function can be written in the form

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i \in I} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b_0 \right). \tag{3.1}$$

where $I = \{i \in \{1, \cdots, l\} | \mathbf{x}_i \text{ is a support vector } \}$ and the scale bias b_0 is given by

$$b_0 = y_i - \sum_{j=1}^{l} \alpha_j^* y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

for any support vector $\mathbf{x}_i$. So, based on the extracted features, a classifier has been established, that can classify a new example as belonging to one of two classes.

Suppose that we need to verify an image u. At first, we extract the features by using the mapping F. We get the feature vector $\mathbf{x} = F(u)$. And then, we compute the output of a function f corresponding to an input $\mathbf{x}$. If $f(\mathbf{x}) = 1$ then, the considered case is positive (there is the disease appearing on the image), otherwise, it is negative.

Finally, the classifier was tested on the test set to evaluate the efficiency of the proposed algorithm.

3.2 Vessel network extraction

3.2.1 Introduction

Digital fundus imaging in ophthalmology plays important role in medical diagnosis of several pathologies like hypertension, diabetes, and cardiovascular disease. Computer-aided image analysis of the eye fundus is highly desirable in many cases. For example, the diagnosis of diabetic retinopathy, the leading cause of blindness in the Western World, requires the screening of a large number of patients from specialized personnel and can be extremely facilitated with the adoption of automatic tools.

One task of the utmost importance is the segmentation of the vasculature in retinal fundus images. Several morphological features of retinal veins and arteries, like diameter, length, branching angle, and tortuosity, have diagnostic relevance and can be used for monitoring progression of diseases. Additionally, vessel segmentation is very important in an automatic screening for fundus images and vessels are needed to be segmented and be removed from retinal images before automatically detecting the dark lesions in retinal image is automatically detected. Retinal vessel segmentation is important for the detection of numerous eye diseases. In particular, medical image segmentation extracts meaningful information and facilitate the display of this information in a clinically relevant way. A crucial role for automated information extraction in medical imaging usually involves the segmentation of regions of the image in order to quantify volumes and areas of interest of biological tissues for further diagnosis and localization of pathologies. The high accuracy of vessel segmentation can reduce the positive error detection of other lesions such as microaneurysms in a retinal image.

Manual segmentation of retinal blood vessels is a long and tedious task which also requires training and skill. It is commonly accepted by the medical community that automatic quantification of retinal vessels is the first step in the development of a computer-assisted diagnostic system for ophthalmic disorders.

3.2.2 State of the art

A large number of methods for retinal vessel segmentation have been published. In this section, we just briefly summarize state-of-the-art methods. A detail analysis exceeds the scope of this thesis. However, a more complete review of existing methods for retinal blood vessel segmentation can be referenced at [74].

3.2.2.1 Matched Filter Approach

This algorithm was first proposed by Chaudhuri *et al.* [54]. It relies on a correlation measure between the expected shape sought for and the measured signal. They observed two interesting properties of the blood vessels in retinal images. First, since the blood vessels usually have small curvatures, the anti parallel pairs may be approximated by piecewise linear segments. Second, although the intensity profile varies by a small amount from one vessel to another, it may be approximated by a Gaussian Curve.

To enhance retinal vasculature a 2D matched filter kernel was designed to convolve with the original fundus image. The kernel was rotated into either eight or twelve orientations to fit into blood vessels of various configurations. The kernel is designed to model a feature in the image at some unknown position and orientation, and the matched filter response (MFR) indicates the presence of the feature. Chaudhuri *et al.* used $f(x, y) = A\,(1 - k\exp(-d^2/2\sigma^2))$, where d is the perpendicular distance between the point (x, y) and the straight line passing through the center of the blood vessel in a direction along its length, σ defines the spread of the intensity profile, A is the gray-level intensity of the local background, and k is a measure of reflectance of the blood vessel relative to its neighborhood. The filter is applied at 12 orientations over 180 and the maximum response of these filters at each location is selected as the vessel edge. The final set of vessel segments is then obtained by applying a linear classifier algorithm. They assumed that all the blood vessels in the image are of equal width 2σ.

In [75], the author proposed a method which extracts the features at each pixel using line operators. The basic line operator is a line with length l centered at considered pixel. At each pixel the average of image gray level along line operators with 12 different orientations spanning 360 degrees are evaluated. The direction for which line operator provides the maximum gray level is selected and the corresponding gray level is denoted by L. The difference represents the line strength of the pixel is given by $S = L - N$, where N is the average gray level in the square window, centered on the pixel, with edge length equal to l. Lines of different lengths have been considered but the best performance was obtained with $l = 15$ pixels. The second feature of the line operator is evaluated using gray level of the pixel neighborhood along the line perpendicular to the line operator of the first feature. Its average gray level is denoted with L_0. Its strength is obtained again subtracting the average intensity in the square window, that is $S_0 = L_0 - N$. Finally, the feature vector which used for training a supervised classifier, is constructed as follow $\mathbf{x} = [S; S_0; I]$ where I is gray level of the pixel which is added to help reducing false detection due to pathology or to the proximity of the optic disk.

Partial Gaussian kernels have also been utilized for vessel detection. In [76], the amplitude-modified second order Gaussian filter was used as a kernel. It proves that the vessel width can be measured in a linear relationship with the spreading factor of the matched Gaussian filter when the magnitude coefficient of the Gaussian filter is suitably assigned. The vessel width measurement not only provides the size of blood vessel but it is also useful for optimizing the matched filter to improve the successful rate of detection [74]. B. Zhang *et al.* [77] extended the classical matched filter with the first-order derivative of the Gaussian (MF-FDOG) to exploit the property that for a blood vessel in the retina. Their method uses a pair of filters, the zero-mean Gaussian filter (MF) and the first-order derivative of the Gaussian (FDOG), to detect the vessels. The methodology significantly reduces the false detections produced by the original MF and detects many fine vessel that are missed by the MF.

Additionally, there are some methods which are proposed to improve the original method [78, 79]. The methodologies attain higher accuracy.

Matched filter approach is usually followed by some other image processing op-

erations like thresholding, thinning process, *etc.* to detect vessel centerlines and obtains the final vessel contours.

3.2.2.2 Tracking Methods

The tracking methods look for a continuous blood vessel fragment starting from a point given either manually or automatically, depending on certain local information. These methods normally try to get the path which best matches a vessel profile model. Vessel tracking approaches start from an initial point, detect vessel centerlines or boundaries by analyzing the pixels orthogonal to the tracking direction. Sobel edge detectors, gradient operators and matched filters were used for finding the vessel direction and boundary. Different methods are employed in determining vessel contours or center lines. The main advantage of vessel tracking methods is that they provide highly accurate vessel widths, and can provide information about individual vessels that is usually unavailable using other methods. It can also give information on vessel structure such as branching and connectivity.

I. Liu, Y. Sun introduced an adaptive tracking algorithm detecting vasculature in retinal angiograms, where the local vessel trajectories are estimated after giving an initial point within a vessel. This algorithm was based upon a recursive tracking procedure. Matched filters were used for guiding the tracking of a single vessel and the detection of side branches. The algorithm showed a good performance when it was applied to angiograms of coronary and radial arteries but it requires the user to specify vessel starting points.

Liang *et al.* [81] developed an algorithm to find the course of the vessel centerline and measure the diameter and tortuosity of a single vessel segment. The tracking algorithm is only based on vessel properties, under the constraint of a Gaussian modeled vessel profile. The matched filter helps to ignore small branches at a bifurcation point without any special handling, thus allowing the tracking process to follow one major branch continuously. However, the algorithm needs manual intervention for start and end points and definition of the tracking direction.

Delibasis *et al.* [82] presented an automatic model-base tracing algorithm for vessel segmentation and diameter estimation. The algorithm is based on a novel parametric model of a vessel that can assume arbitrarily complex shape and a simple measure of match that quantifies how well the vessel model matches a given angiographic image. A vessel tracking algorithm is described that exploits the geometric model and discovers vessel bifurcation. The vessel tracking algorithm is initialized automatically using multiple near-central axis vessel points. The vessel tracking is derived by identifying the best matching strip with the vessel by using the seed point, strip orientation, strip width and the measure of match (which quantifies the similarity between the model and the given image). Following the termination of vessel tracking, the algorithm actively seeks vessel bifurcation, without user intervention. The vessel diameter is also recovered with the defined model using the strip width parameter therefore assuming linear dependency between vessel diameter and model width parameter.

There are some other tracking strategies [83, 84] which have been proposed to obtain sequential contour tracking by incorporating the features, such as the vessel central point and search direction detected from the previous step into the next

step. However, one disadvantage of the vessel tracking approaches is that most of them are not fully automatic and require user intervention for selecting start and end points.

3.2.2.3 Morphological Processing

The mathematical morphology methods use the knowledge of vessel shape features such as piecewise linear and connected. The term mathematical morphology is used as a tool for extracting image components that are useful in the representation and description of region shapes such as features, boundaries, skeletons and convex hulls. Morphology operators apply structuring elements to images, and are typically applied to binary images but can be extended to the gray-level images. The main two morphological operators are *Dilation* and *Erosion*. *Dilation* expands objects by a defined Structuring Element, filling holes, and connecting the disjoint regions. *Erosion* shrinks the objects by a Structuring Element. The other two operations are *Closing*, which is a dilation followed by an erosion, and *Opening*, i.e. an erosion followed by a dilation. Two algorithms used in medical image segmentation and related to mathematical morphology are top hat and watershed transformations.

Zana and Klein in [85] present a vessel segmentation algorithm from retinal angiography images based on mathematical morphology and linear processing. A unique feature of the algorithm is that it uses a geometric model of all possible undesirable patterns that could be confused with vessels in order to separate vessels from them. The strength of the algorithm comes from the combination of mathematical morphology and differential operators in the segmentation process. It was possible to select vessels using shape properties, connectivity, as well as differential properties like curvature.

Mendonca and Campilho [86] utilized a Difference of Offset Gaussian (DoOG) filter in combination with multiscale morphological reconstruction for retinal vasculature extraction. The vessel centerlines are extracted by applying the DoOG filter and the vessels are enhanced by applying a modified top hat operator with variable size circular structuring elements aiming at enhancement of vessels with different widths.

Y. Yang *et al.* [87] proposed an automatic hybrid method comprising of the combination of mathematical morphology and a fuzzy clustering algorithm. The blood vessels are enhanced and the background is removed with a morphological top-hat operation then the vessels are extracted by fuzzy clustering.

Sun *et al.* [88] combined morphological multiscale enhancement, fuzzy filter and watershed transformation for the extraction of the vascular tree in the angiogram. The background is estimated by using non linear multiscale morphology opening operators with a varying size of structuring element on each pixel and later subtracted from the image for contrast normalization. The normalized angiogram is processed by a combined fuzzy morphological operation with twelve linear structuring elements rotated every 15 deg between zero and 180 deg, with nine pixels length. The vessel region is obtained by thresholding the filtered image followed by a thinning operation to approximate the vessel centerlines. Finally, the vessel boundaries were detected using watershed techniques with the obtained vessel centerline [74].

The fast discrete curvelet transform (FDCT) and multistructure mathematical

morphology [89] is employed for vessel detection. FDCT is used for contrast enhancement and the edges of blood vessels are detected by applying a multistructure morphological transformation. The false edges are removed by morphological opening by reconstruction. An adaptive connected component analysis is performed for length filtering of the detected vascular structures in order to obtain a complete vascular tree.

3.2.2.4 Region Growing Approaches

Starting from some seed point, region growing techniques segment images by incrementally recruiting pixels to a region based on some predefined criteria. Two important segmentation criteria are value similarity and spatial proximity. It is assumed that pixels that are close to each other and have similar intensity values are likely to belong to the same object. The main disadvantage of a region growing approach is that it often requires user-supplied seed points. Due to the variations in image intensities and noise, region growing can result in holes and over-segmentation. Thus, post-processing of the segmentation result is often necessary.

3.2.2.5 Multi-Scale Approaches

The width of a vessel decreases as it travels radially outward from the optic disk and such a change in vessel caliber is a gradual one. The idea behind scale-space representation for vascular extraction is to separate out information related to the blood vessel having varying width at different scales. The main advantage of using these approaches was their efficient processing speed. In these approaches larger blood vessels were segmented from regions having low resolution and finer vessels were segmented from regions having high resolution.

3.2.2.6 Pattern classification approaches

Supervised methods exploit some prior labeling information to decide whether a pixel belongs to a vessel or not, while unsupervised methods perform the vessel segmentation without any prior labeling knowledge.

In supervised methods, the rule for vessel extraction is learned by the algorithm on the basis of a training set of manually processed and segmented reference images often termed as the gold standard. This vascular structure in these ground truth or gold standard images is precisely marked by an ophthalmologist.

Artificial neural networks have been extensively investigated for segmenting retinal features such as the vasculature [90] making classifications based on statistical probabilities rather than objective reasoning. These neural networks employ mathematical weights to decide the probability of input data belonging to a particular output. This weighting system can be adjusted by training the network with data of known output typically with a feedback mechanism to allow retraining.

Nekovei and Sun [91] describe an approach using a back-propagation network for the detection of blood vessels in X-ray angiography. The method applies the neural network directly to the angiogram pixels without prior feature detection. Since angiograms are typically very large, the network is applied to a small sub-window

which slides across the angiogram. The pixels of the sub-window are directly fed as input to the network. Pre-labeled angiograms are used as the training set to set the network's weights. A modified version of the common delta-rule is to obtain these weights.

Sinthanayothin *et al.* [92] preprocessed images with principal component analysis (PCA) to reduce background noise by reducing the dimensionality of the data set and then applied a neural network to identify the pathology. They reported a success rate of 99.56% for the training data and 96.88% for the validation data, respectively, with an overall sensitivity and specificity of 83.3% (standard deviation 16.8%) and 91% (standard deviation 5.2%), respectively.

Staal *et al.* [93] used KNN-classifier with 27-D feature vector based on ridges information. Their method depends on extracting ridges in the image, forming line elements from ridges, assigning each pixel to nearest line to partition image into patches and computing features vector of each pixel based on its line and patch attributes. The methodology is tested on the publically available STARE [94] and Utrecht database obtained from a screening program in the Netherlands. The method achieves an average accuracy of 0.9516 and an area under the ROC curve of 0.9614 on the STARE data set.

Soares *et al.* [96] proposed a method which uses 2-D Gabor wavelet and supervised classification for retinal vessel segmentation. Each pixel is represented by a feature vector composed of the pixel's intensity and two-dimensional Gabor wavelet transform responses taken at multiple scales. At each scale the maximum response of Gabor wavelet over different orientations spanning from 0°to 179°at step of 10°is calculated. Image pixels in this method are classified using Bayesian classifier. The method achieves an average accuracy of 0.9466 and 0.9480 for DRIVE [95] and STARE [94], respectively.

Salem *et al* [97] proposed a RAdius based Clustering ALgorithm (RACAL) which uses a distance based principle to map the distributions of the image pixels. A partial supervision strategy is combined with the clustering algorithm. The features used are the green channel intensity, the local maxima of the gradient magnitude, and the local maxima of the large eigenvalue calculated from Hessian matrix. The same features are used with kNN and RACAL algorithms and later perform better for the detection of small vessels. The methodology attains a specificity of 0.9750 and sensitivity of 0.8215 on the STARE database.

Marin *et al.* [98] used 7-D feature vector consists of five features encode gray-level variation between pixel and its surroundings plus other two features based on Hu moment-invariants and their classifier was the neural network (NN). The average accuracy, on the DRIVE database is 0.9452 for the STARE database 0.9526 respectively.

Remark 3.1 *Vessel segmentation algorithms are the key components of automated radiological diagnostic systems. Segmentation methods vary depending on the imaging modality, application domain, method being automatic or semi-automatic, and other specific factors. There is no single segmentation method that can extract vasculature from every medical image modality. It is difficult to confirm that which method is effective than others. It depend on the aim of each problem. In our research, we choose the method of the authors Bankhead P et al. [59] to get the center*

line of the vessels and the diameters. The authors present a novel algorithm for the efficient detection and measurement of retinal vessels. Their algorithm is general enough so it can be applied both low and high resolution fundus photographs and fluoresce in angiograms upon the adjustment of only a few intuitive parameters. The algorithm described is fully automated analysis the retinal vessel diameters. It allows the fastest diameters computation all along the length of each vessel rather than at specific points of interest.

3.2.3 The method using wavelets and edge location refinement

In this section, we summarize the main steps of the method proposed by Bankhead P *et al.* [98] to get the features of vessels.

3.2.3.1 Vessel segmentation by wavelet thresholding

The Isotropic Undecimated Wavelet Transform (IUWT) is a powerful, redundant wavelet transform that has been used in astronomy [61] and biology [62] applications. It affords a particularly simple implementation that can be readily appreciated without recourse to wavelet theory: at each iteration j, scaling coefficients c_j are computed by lowpass filtering, and wavelet coefficients w_j by subtraction [63]. The scaling coefficients preserve the mean of the original signal, whereas wavelet coefficients have a zero mean and encode information corresponding to different spatial scales present within the signal. Applied to a signal $c_0 = f$, subsequent scaling coefficients are calculated by convolution with a filter $h^{\uparrow j}$.

$$c_{j+1} = c_j * h^{\uparrow j}$$

where $h_0 = [1, 4, 6, 4, 1]/16$ is derived from the cubic B-spline, and $h^{\uparrow j}$ is the upsampled filter obtained by inserting $2^j - 1$ zeros between each pair of adjacent coefficients of h_0. If the original signal f is multidimensional, the filtering can be applied separably along all dimensions. Wavelet coefficients are then simply the difference between two adjacent sets of scaling coefficients, i.e.

$$w_{j+1} = c_j - c_{j+1}.$$

Reconstruction of the original signal from all wavelet coefficients and the final set of scaling coefficients is straightforward, and requires only addition. After the computation of n wavelet levels,

$$f = c_n + \sum_{j=1}^{n} w_j.$$

The set of wavelet coefficients generated at each iteration is referred to as a wavelet level, and one may see that larger features (including vessels) are visible with improved contrast on higher wavelet levels. Segmentation can then be carried out very simply by adding the wavelet levels exhibiting the best contrast for vessels and thresholding based upon a percentage of the highest (if applied to an angiogram) or lowest (if applied to a fundus image) valued coefficients. The thresholds should

be computed from pixels within the field of view (FOV) in order to ensure that the dark pixels outside this do not contribute to the threshold chosen; if a FOV mask is not available, one can normally be produced by simply applying a global threshold to the image. This is best applied to the red channel of a color fundus photograph.

The choice of wavelet levels and thresholds do not typically need to be changed for similar images; indeed, in all cases for fundus images (both low and high resolution) they set the threshold to identify the lowest 20% of coefficients as vessels, and varied only the choice of wavelet levels if the image sizes were different. Because the percentage of vessel pixels within the FOV is more typically around 12-14% (as determined using manually segmented images), the thresholded image is to be over-segmented (i.e. many non-vessel pixels have been misclassified as vessels). However, the majority of the vasculature is represented by one large connected structure in the binary image, whereas misclassified pixels tend to be clustered to form isolated objects. These small objects can be removed simply based upon their area, either in terms of pixels or a proportion of the image size. Similarly, small holes present within thresholded regions can be filled in. Most remaining erroneous detections are removed during later processing steps.

3.2.3.2 Centerline computation

The next step is to apply a morphological thinning algorithm [64]. Thinning iteratively removes exterior pixels from the detected vessels, finally resulting in a new binary image containing connected lines of 'on' pixels running along the vessel centers. The number of 'on' neighbors for each of these pixels is counted: end pixels (< 2 neighbors) are identified, and branch pixels (> 2 neighbors) are removed. The removal of branches divides the vascular tree into individual vessel segments in preparation for later analysis. This is useful because diameters are not well-defined at branches or bifurcation are not directly comparable with those measured afterwards, as less blood will flow through the vessel afterwards and there will be a drop in pressure.

The elimination of as many uninteresting centerlines as possible at this stage helps to improve the speed of the later processing steps. To this end, centerlines are first cleaned up by removing short segments (< 10 pixels). Because any of these short segments that contained end pixels were likely to be spurs, which often occur as an unwanted side-effect of thinning, their corresponding branch pixels are replaced to avoid causing the main vessel to which they were connected being erroneously subdivided. A coarse estimate of vessel diameters is then calculated using the distance transform of the inverted binary segmented image. This gives the Euclidean distance of every "vessel" pixel from the closest non-vessel pixel, and therefore doubling the maximum value of the distance transform along the thinned centerlines provides an estimated diameter of every vessel segment at its widest point.

3.2.3.3 Centerline refinement using spline fitting

The orientation of a vessel segment at any point could be estimated directly from its centerline, but discrete pixel coordinates are not well suited for the computation of angles. A least-squares cubic spline (in piecewise polynomial form) is therefore

fitted to each centerline to combine some smoothing with the ability to evaluate accurate derivatives (and hence vessel orientations) at any location. A parametric spline curve is required, with appropriate parameterization essential to obtain a smooth centerline. For this, the centripetal scheme was used, which was described by Lee [65].

Adjusting the spacing of the breaks between polynomial pieces in the spline can give some control over a preference for smoothness or the ability to follow complex shapes. The precise break spacing can vary because the vessel segment is divided into polynomial pieces of equal length and the segment length is unlikely to be an exact multiple of the polynomial piece length. If the number of data points is very low, a single cubic polynomial is fit to the centerline instead.

3.2.3.4 Vessel edge identification

The measurement of diameters requires the location of edge points, but these have no single "natural" definition within the image space. Vessel profiles in fundus and fluorescein angiography images resemble Gaussian functions, and edges have previously been defined in a variety of ways, including using gradients or model fitting [66]. One of the main complications encountered when trying to develop a general vessel diameter measurement strategy is the possible presence of the 'central light reflex' [67], which is seen as a 'dip' or 'hill' approximately in the center of the vessel profile, and which is more likely to be found in higher resolution images and wider vessels. Its origins are unclear, although it is thought to emanate from the column of densely packed erythrocytes moving through the retinal microvasculature [68]. The marked enhancement of the light reflex may be of clinical interest; for example, it appears to be associated with hypertension, although further investigation and a more objective quantification of changes are needed [68]. That some vessel measurement algorithms have misidentified the light reflex as the vessel edges has been reported as problematic [66, 69], and explicit strategies for dealing with this issue are required to ensure that any measurement is sufficiently robust [66], [70]-[72].

Here, they define an edge as occurring at a local gradient maximum (the rising edge) or minimum (the falling edge), as identified to sub-pixel accuracy using the zero-crossing of the second derivative. They have adopted a four-step method to identify these edges for each vessel:

1. Estimate the average vessel width from the binary profiles. The sum of 'vessel' pixels in each profile is computed, and the median of these sums is taken as the provisional width.

2. Compute an average of all the vessel profile (omitting pixels previously identified as belonging to other vessels or outside the FOV), and identify the locations of the maximum and minimum gradient to the left and right of the center respectively, bounded to a search region of one estimated diameter from the center. These locations give the column in the vessel profile images at which edges are predicted to fall. The distance between the two columns also gives a more refined and robust estimate of mean vessel width, largely independent of the threshold used for the initial segmentation.

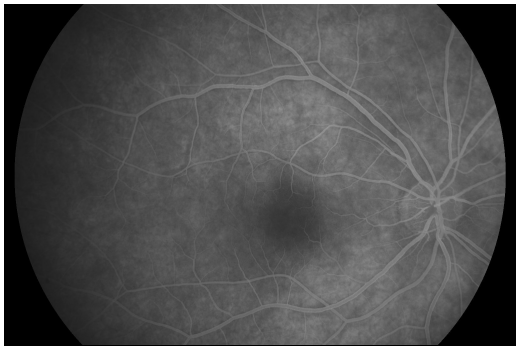
3. Apply an anisotropic Gaussian filter to the vessel profiles image to reduce noise, and then calculate a discrete estimate of the second derivative perpendicular to the vessel by finite differences.

4. Identify locations where the sign of the pixels in each filtered profile changes, and categorize these based upon the direction of the sign change into potential left and right vessel edges. Using connected components labeling, link the possible edges into distinct trails. Remove trails that never come within 1/3 of an estimated vessel diameter from the corresponding predicted edge columns. The final edges are then the zero-crossings belonging to the longest remaining trails to each side of the vessel center, and the diameter is simply the Euclidean distance between these edges.

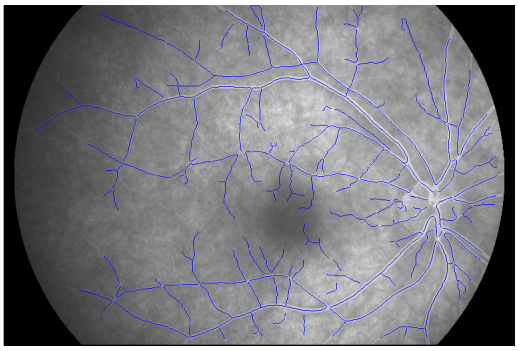
In the ideal case, a single trail of suitable zero-crossings will exist to the left and right of the vessel center and edge identification is straightforward. The additional tests are intended to produce reasonable results whenever the edge may be broken, while avoiding misclassifying zero-crossings due to the central light reflex or other image features. The smoothing in the third step deals with the sensitivity to noise of computing approximations of derivatives applied to discrete data. The horizontal and vertical sigma values σ_H and σ_V of the Gaussian filter are calculated by the scaling the square root of the estimated widths w produced by the previous step, and therefore more smoothing is applied to vessels with lager diameters. They used $\sigma_H = \sqrt{0.1w}$ and $\sigma_V = \sqrt{2w}$ for all images although scaling parameters may be adjusted according to image noise. Because this smoothing is applied to the stacked image profiles rather than the original image, the filter is effectively oriented parallel to the vessel at each point. This ensures that most blurring occurs within or alongside the vessel - rather than in all directions, which might have otherwise affected edges or merged vessels with neighboring structures.

3.2.4 The results on real data

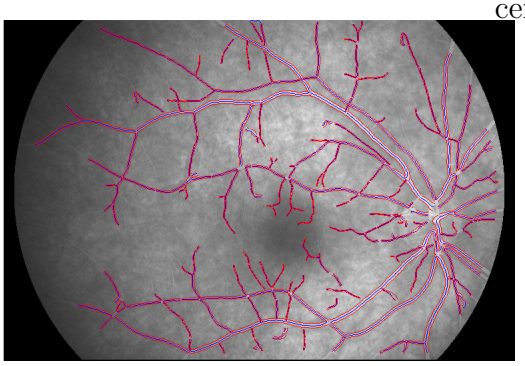
We applied the algorithm to our data to get the features of vessels. The results have shown that the algorithm can effectively extract almost all vessels from the image. It is also shown that the algorithm can be used to measure the diameter of vessels with quite high precision. Some examples of segmentation are shown in Figures 3.3 and 3.4.



(a) Original image

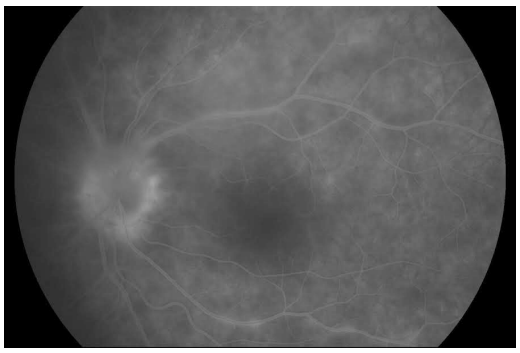


(b) Segmented image with centerlines.

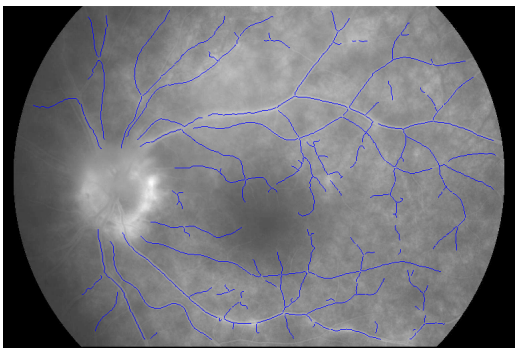


(c) Segmented image with edges.

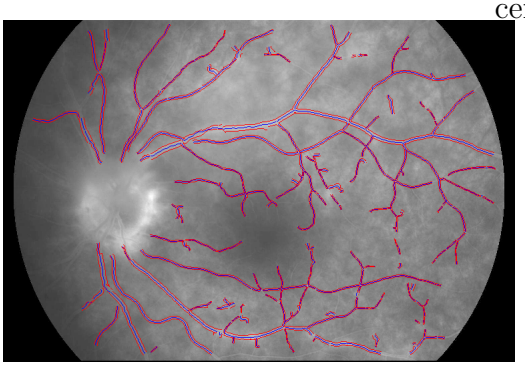
Figure 3.3: The segmentation of a healthy fundus image



(a) Original image



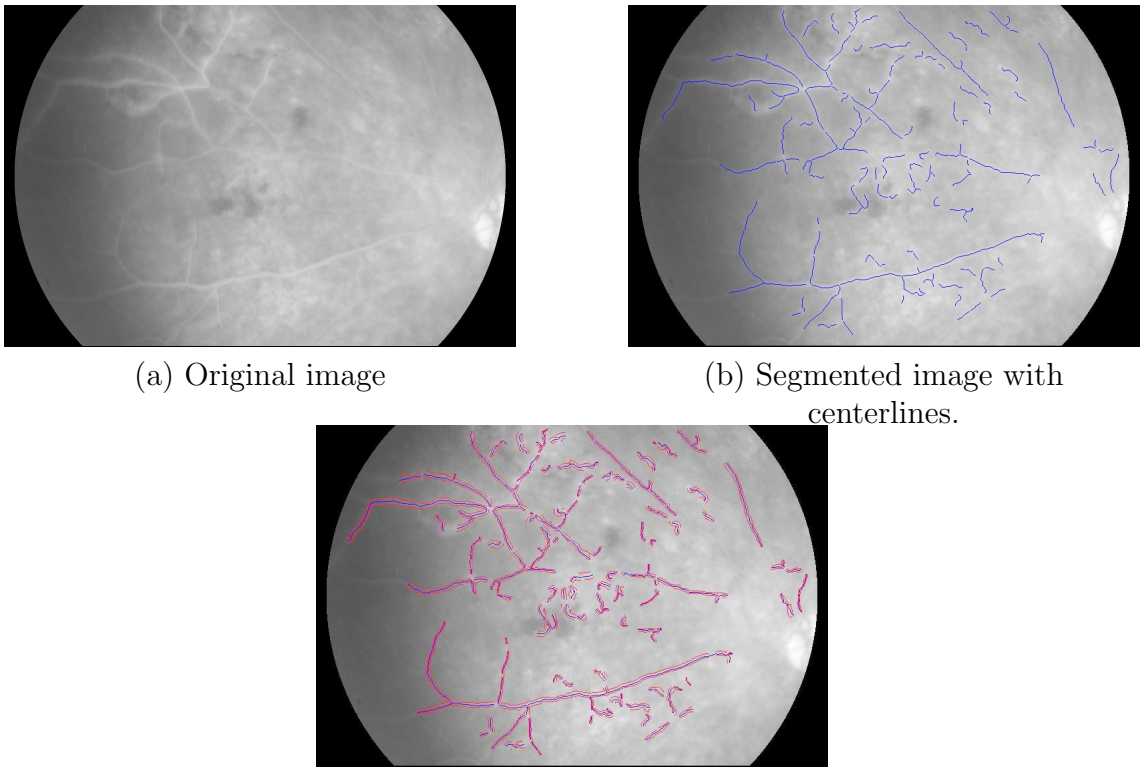
(b) Segmented image with centerlines.



(c) Segmented image with edges.

Figure 3.4: The segmentation of a pathological fundus image

However, the segmentation is not always good. In some cases, the algorithm gives out results with some mistakes. Figure 3.5 shows the segmentation of one abnormal image. There are some mistakes occurring on the result. This may be one of the causes affecting to the final results in the classification process.



(a) Original image

(b) Segmented image with centerlines.

(c) Segmented image with edges.

Figure 3.5: The segmentation for one abnormal image, which shows some mistakes.

3.3 BV norm computation with a histogram

As mentioned above, the total variation has been introduced in Image Processing first by Rudin, Osher and Fatemi [60], as a regularizing criterion for solving inverse problems. It has been proven to be quite efficient for regularizing images without smoothing the boundaries of the objects.

In our method, the total variation is used for measuring the change of image intensity. This method can be used because image is blurred in the same place of the blood vessel if the disease occurs (see Figure 3.1). We calculate BV norm at each point on centerlines in circle area or in rectangular area.

3.3.1 Computation of BV norm along centerline

After applying the algorithm, we have the full information about the vessels. In the next step, at each point P on centerline, we calculate the corresponding BV norm.

We assume for simplicity that the image u is squared with size $N \times N$. We note $X := \mathbb{R}^{N\times N}$ endowed with the usual inner product and the associated euclidean norm

$$\langle u, v\rangle_X := \sum_{1\leq i,j\leq N} u_{i,j}v_{i,j},\ \|u\|_X := \sqrt{\sum_{1\leq i,j\leq N} u_{i,j}^2}.$$

As presented in the Section 1.3.2, to define the discrete total variation, we first introduce the notation of the discrete gradient of the numerical image.

With an numerical image $u \in X$, the discrete gradient of u is $\nabla u \in X^2$ defined by

$$(\nabla u)_{i,j} = \big((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}\big)$$

where

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \text{ and } (\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}.$$

In our strategy, we calculate the total variation on two different domains (circle and rectangle), which is described in Figure 3.6 and in Figure 3.7 is the following.

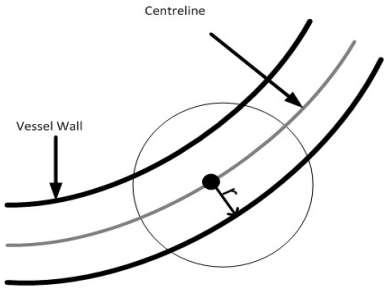


Figure 3.6: BV norm computation on circle domain

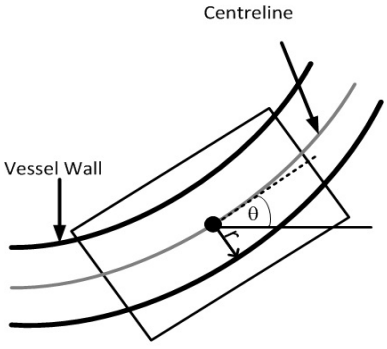


Figure 3.7: BV norm computation on rectangular domain

In the case of circular domain, at each point P on the centerlines, the total

variation is given by

$$J(P) = \sum_{(i,j)\in I} |(\nabla u)_{i,j}|\,, \tag{3.2}$$

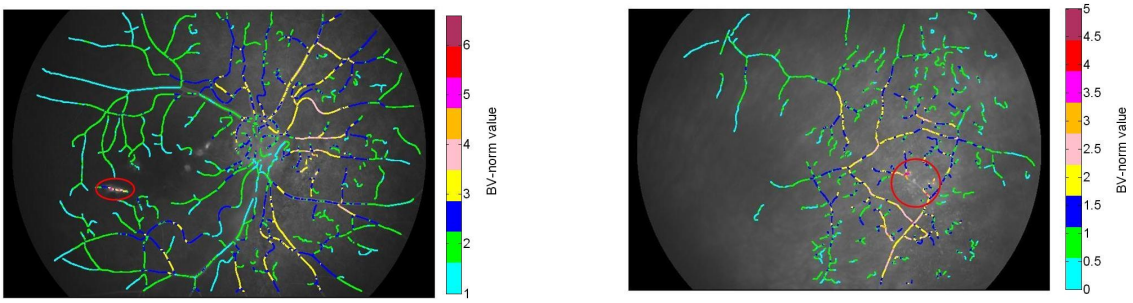
where I is the set of double indices corresponding to points in the disc of center P and radius R. We will choose $R \geq r$, where r is the radius of blood vessel at P. In the case of rectangular domain, the total variation is given by

$$J(P) = \sum_{(i,j)\in D} |(\nabla u)_{i,j}| \tag{3.3}$$

where D is set of double indices corresponding to point in the rectangle with center P, and width $w \geq 2r$. The domain D is chosen to follow the direction of the vessel (see Figure 3.7).

3.3.2 Using the BV norm to detect the diseased region

After computing BV norms at each point of the vessel centerlines, we normalize them via dividing by the area on which the BV norm is calculated. To display the results with color map, we choose the range of colors including [Cyan; Green; Blue; Yellow; Pink; Orange; Magenta; Red; Maroon]. This range is ordered from light to dark. The inflamed vessels branches then appear with darker color (see Figure 3.8 (a) and Figure 3.8 (b), diseased vessels are indicated by a red circle).



(a) (b)

Figure 3.8: The examples show the using BV norm to detect the diseased region.

In the following section, we construct histograms, which show the difference between diseased images and normal images.

3.3.3 Histogram Construction

From the experimental results by color map on diseased images in the previous section, we create histograms on both of the diseased images and non-diseased images. The range $[min_{BVnorm}, max_{BVnorm}]$, is divided equally into nb intervals, and then we count points on centerlines corresponding to BV norms in each interval. The number of points in each interval is normalized by dividing by the total points of centerlines. Those histogram will be used for classification between diseased cases

and non diseased cases. The next pictures (see Figure 3.9) are histograms which present the difference between diseased images and normal images. In histograms of normal case, the proportion of BV-norm which corresponds to the higher value is small, whereas the one in histograms of diseased case is bigger. We shall use the simple observation to classify the data set.

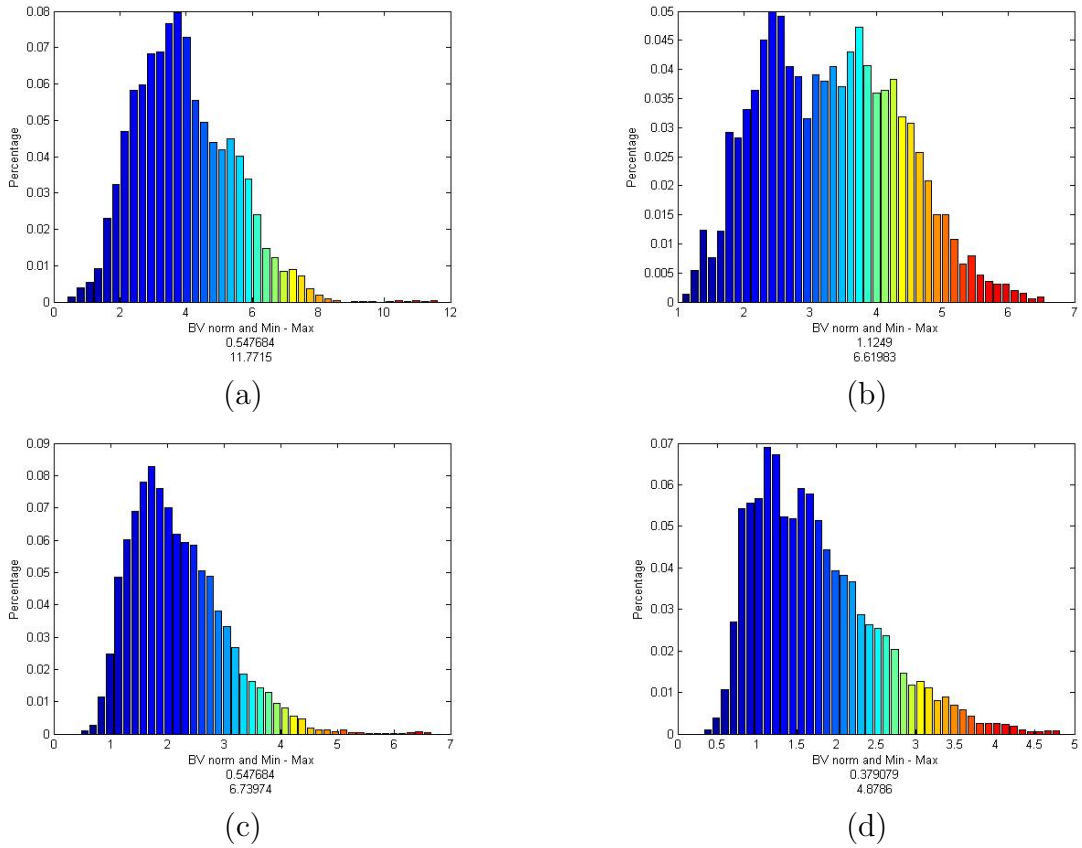


Figure 3.9: The histograms on the left correspond to healthy fundus image, while the histograms on the right correspond to pathological fundus image.

3.4 The classification algorithm

In this section, we present our experiments. All steps are described by means of a diagram in detail in Figure 3.10.

3.4.1 Generation of the training and the testing file for classification

In the first step, we construct training and testing data sets with manually labeled marks. We collect a total of 49 images. We get all images from Toulouse clinic department of ophthalmology and neurology. The pathological images are marked by our expert. After that we divide randomly it into two sets, one set for training and one set for testing. The training set contains 24 images and the test set contains 25 images.

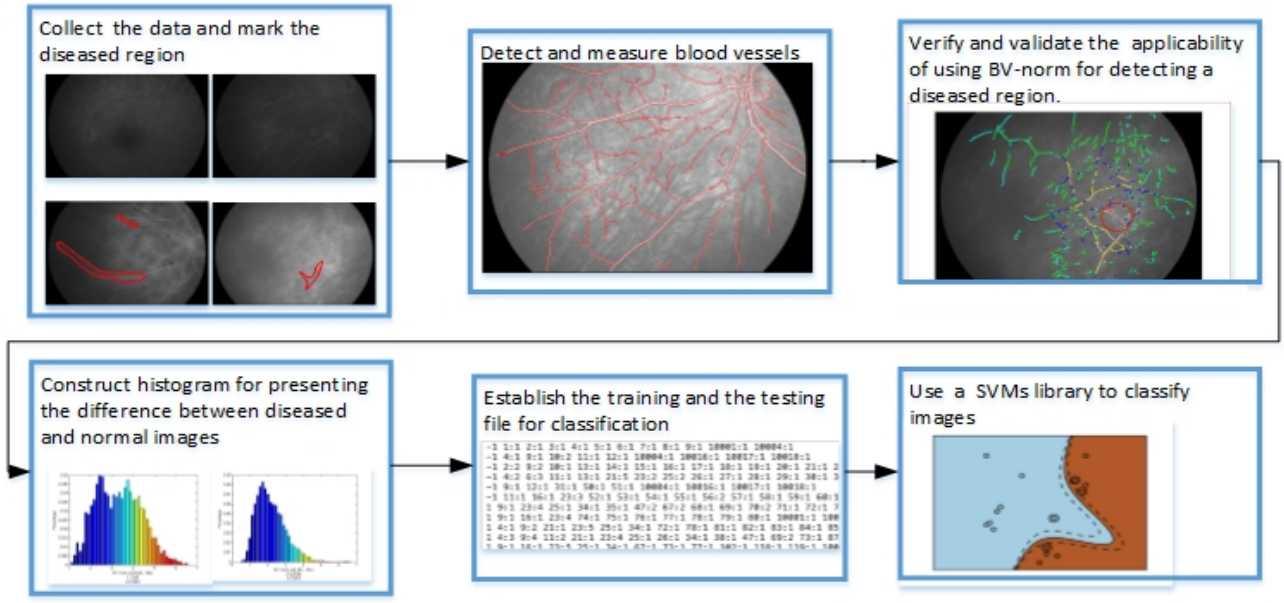


Figure 3.10: Diagram describing the different steps of the proposed method

When we have the data, we use the method proposed in Section 3.2.3. All pictures are processed by using the Bankhead algorithm to get characteristics of vessels (centerlines, radius, angles, ...). The results are stored in files (*.dat).

With each image, we compute the BV norm along the centerlines with various sizes of regions. BV norms at each point of the vessel centerlines are normalized by dividing by the area on which they were calculated. After that, we divide BV norm range $[min_{BVnorm}, max_{BVnorm}]$ into nb intervals, where nb is considered as a parameter. Points on centerlines corresponding to BV norm in each interval are counted and then are normalized by dividing the total points of centerlines. Now, from one image u, we create a vector $\mathbf{x} \in \mathbb{R}^{nb}$, where, the ith entries x_i of $\mathbf{x}$ is the percentage of points corresponding to BV norm in the ith interval.

The output consists of two files (test file and training file). Each line on the output file is target y_u and nb features of the image u.

3.4.2 Using SVM to classify the samples

We used two parameters in this experiment: Δr and nb where $\Delta r, nb \in \mathbf{N}$. If the radius of a vessel at the considered point is r, then $r + \Delta r$ is the radius of the domain on which the BV norm is calculated. The number of intervals between the minimum and the maximum values of the BV norm is represented by nb. The value of Δr ranges between 0 and 3. With one value of Δr and nb, we construct a pair of files: the training file and the test file. The training file serves as the input in the SVM procedure, and the output is the separating hypersurface. By using such a tool, we are able to separate healthy from pathological cases in the test file. Once the hypersurface has been computed, a new sample $\mathbf{x}_i$ can be easily classified as healthy or pathological. This feature is called *generalization* in the machine learning community.

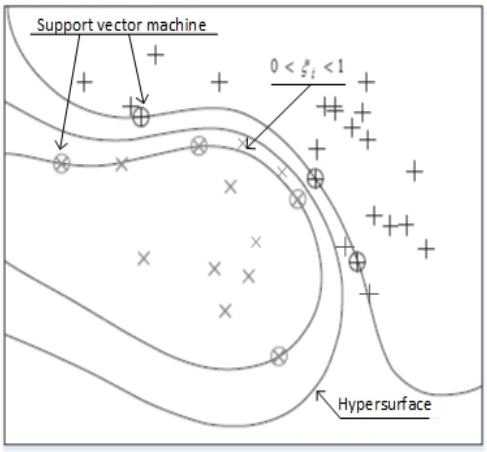


Figure 3.11: A separating hyperplane in the feature space may correspond to a non-linear boundary in the input space. The figure shows the classification boundary in a two-dimensional input space as well as the accompanying soft margins. The middle line is the decision surface; the outer lines precisely meet the constraint (where, the constraint in problem (P) becomes an equality with $\xi_i = 0$).

We used the libsvm (v 3.12) [73] library from Chih-Chung Chang and Chih-Jen Lin for classifying images. In this tool, we chose various parameters for the options to get the best result.

An important choice in the adoption of the support vector framework for problems such as these is that of the kernel K. In the experiments, we tried 4 commonly used kernels: linear, inhomogeneous polynomial, radial basis function (Gaussian kernel) and sigmoid. The results show that, using inhomogeneous polynomial kernel and Gaussian kernel gives out better results.

Scaling data before applying SVM is quite important. The main advantage of scaling is to avoid attributes with large numerical ranges to dominate over those with smaller numerical ranges. Another advantage is to avoid numerical difficulties during the computation. However, in our study it does not improve significantly the results since, for every image u, the corresponding feature vector $\mathbf{x}$ has entries in interval $[0, 1]$.

By changing various parameters, the algorithm achieved accuracy of 84% (21/25). The best result was achieved with $\Delta r = 1$. We also tested the influence of the parameter nb and we got better results when nb was in the range of 20 to 55.

3.5 Conclusion and perspectives

In this part, we have presented a method for the classification of retinal images. We have tested 49 images and the result we got is 84% of good classification. This is not perfect but quite acceptable, and the misclassification can easily be explained by experimental errors. For example, images sometime have light artifact or bad resolution. Furthermore, the segmentation for abnormal images has not been very good in some cases. In the experiments, we calculated the BV norm on different domains (circular and rectangular) but it is not clear which case will give the best

results. In the future, we hope to receive more pictures for testing and we expect that we can establish a complete data set which will be public. A larger data set would also allow us to evaluate the efficiency of the proposed algorithm more precisely. The complete database of this disease can be used for future research. Finally, it would also be interesting to explore alternative characteristics (to the BV norm) with the hope of improving image classification, as well as the classification of other diseases such as exudate, drusen, cotton wool spots, etc.

Part III

Air traffic complexity metric

In this part, we focus on the theme of air transport. Specifically, we introduce a new method for extracting the main flows of the air traffic. After that, we propose one method that calculates a complexity indicator for the air traffic.

Chapter 4

Air traffic complexity

4.1 Introduction

The operational capacity of a control sector is currently measured by the maximum number of aircraft able to traverse the sector in a given time period. This measurement does not take account of the orientation of traffic and considers geometrically structured and disordered traffic in the same manner. Thus, in certain situations, a controller may continue to accept traffic even if operational capacity has been exceeded (structured traffic); in other situations, controllers may be obliged to refuse additional traffic even though operational capacity has not yet been reached (disordered traffic). Thus, a measurement in terms of the number of aircraft per unit of time constitutes an insufficient metric for the representation of the difficulty level associated with a particular traffic situation.

In the context of operational control, the ideal would be to find a metric which precisely measures the level of mental effort needed to manage a set of aircraft. Without going quite so far, it is possible to find complexity metrics which go beyond a simple measurement of the number of aircraft. We shall begin by clarifying two essential notions for use in the rest of this chapter:

- **Control workload**: measurement of the difficulty for the traffic control system of treating a situation. This system may be a human operator or an automatic process. In the context of operational control, this workload is linked to the cognitive process of traffic situation management (conflict prediction and resolution, trajectory monitoring, etc.).

- **Traffic complexity**: intrinsic measurement of the complexity associated with a traffic situation. This measurement is independent of the system in charge of the traffic and is solely dependent on the geometry of trajectories. It is linked to sensitivity to initial conditions and to the interdependency of conflicts. Incertitude with respect to positions and speeds increases the difficulty of predicting future trajectories. In certain situations, this incertitude regarding future positions can increase exponentially, making the system extremely complex in that it is virtually impossible to reliably extrapolate a future situation. When a future conflict is detected, a resolution process is launched which,

in certain situations, may generate new conflicts. This inter-dependency between conflicts is linked to the level of mixing between trajectories.

Research into air traffic complexity metrics has attracted considerable attention in recent years, particularly in the United States and in Europe. The first projects were launched in Germany in the 1970s, and since then the subject has continued to develop. Currently, NASA, MIT and Georgia Tech are involved in work on the subject within the framework of the NextGen project. In Europe, the DSNA, the DLR and the NLR are involved in similar activities linked to SESAR.

4.2 State of the Art-air traffic complexity

The airspace complexity is related with both the structure of the traffic and the geometry of the airspace. Many works have focused on this problem in order to exhibit an efficient measure of airspace congestion. In this section, we review some air traffic complexity metrics which have been proposed in the literature. The paper [99] presents a summary of the main complexity metrics which have been developed in some previous related works.

(1) *Aircraft Density:* Observation of the positions of airplanes in a volume of airspace allows us to determine a level of aggregation known as *density* which is used to characterize the geographical distribution of aircraft. Density is used to identify spatial zones with high levels of aggregation in relation to their volume. Thus, for a constant number of airplanes in a sector, density is used to distinguish whether these aircraft are distributed homogeneously or in the form of clusters.

(2) *Dynamic density:* Laudeman *et al.* from NASA [108] have developed a metric called "Dynamic Density" which is based on the flow characteristics of the airspace. The "Dynamic Density" is a weighted sum of the traffic density (number of aircraft), the number of heading changes (>15 degrees), the number of speed changes (>0.02 Mach), the number of altitude changes (>750 ft), the number of aircraft with 3-D Euclidean distance between 0-25 nautical miles, the number of conflicts predicted in 25-40 nautical miles. The parameters of the sums have been adjusted by showing different situations of traffic to several controllers. Finally, B.Sridhar from NASA [101], has developed a model to predict the evolution of the metric in the near future. Efforts to define "Dynamic Density" have identified the importance of a wide range of potential complexity factors, including structural considerations.

(3) *Interval Complexity:* Interval Complexity developed by P. Flener *et al.* [102] estimates controller workload in a given sector. For a given sector s and a given time interval $[m, ..., m + kL]$, interval complexity is computed by averaging instantaneous complexities of s for the $k+1$ time samples $m+iL$, for $0 \leq i \leq k$. Instantaneous complexity of sector s at time m is a normalized weighted sum of the following terms:

$$C(s, m) = (w_1 N_{sec} + w_2 N_{cd} + w_3 N_{nsb}) \cdot S_{norm}$$

where N_{sec} is the number of flights in s at m; N_{cd} is the number of non-level (climbing or descending) flights in s at m; N_{nsb} is number of aircraft that fly close to the border of the sector; S_{norm} is a sector normalization constant. The parameters N_{sec} and N_{nsb} require special attention and procedures to be followed by the ATC.

(4) *Fractal Dimension:* Fractal dimension is thought to be an aggregate metric for measuring the geometrical complexity of a traffic pattern, which evaluates the number of degrees of freedom used in a given airspace. It has been proposed by S. Mondoloni and D, Liang in [103]. The fractal dimension is a ratio providing a statistical index of complexity comparing how detail in a pattern (strictly speaking, a fractal pattern) changes with the scale at which it is measured. The block count approach is used to compute the fractal dimension of a geometrical entity. In this approach, the entire volume is subdivided into a collection of blocks with dimension d and the number of blocks contained in this entity is counted (N). The fractal dimension D_0 is then given by :

$$D_0 = \lim_{d \to 0} \frac{\log N}{\log d}$$

The authors show a relation between fractal dimension and conflict rate (number of conflicts per hour for a given aircraft). A higher fractal dimension indicates a higher degrees of freedom in the airspace. This information is independent of sector and does not scale with traffic volume.

(5) *Input-Output Approach:* In [104] and [105], it is demonstrated that, air traffic complexity can be measured by the control activity require to avoid the occurrence of conflicts along some reference time horizon when an additional aircraft enters the traffic. Based on a conflict free situation, a new aircraft in included for which some maneuvers of the former aircraft are required. Based on the number of maneuver and their associated extension, a complexity metric can be introduced. A complexity map can also be derived from such a metric. This approach is highly dependent on the kind of algorithm used to solve conflicts.

(6) *Intrinsic Complexity Metrics:* Intrinsic complexity metrics were introduced with the purpose of capturing the level of disorder as well the organization of a set of aircraft trajectories. In [106], two approaches have been proposed, both based on the measurements of the aircraft velocities and positions. The first one describes an air traffic complexity indicator based on the structure and the geometry of the traffic.

The second approach is based on the dynamic system theory used to models air traffic. Based on positions and speeds of aircraft, a regression of the non-linear dynamic system is carried out using the least squares method. This model is used to build a regular field which is perfectly fitted to the observations. Using this model, we can then apply Lyapunov's exponent theory in order to quantify the local level of organization of the vector field. The principle of Lyapunov exponents consists of measuring the sensitivity of the reconstituted vector

field to initial conditions. When an exponent has a high value, it shows a high sensitivity to initial conditions. The future situation is thus very difficult to predict in the zone of calculation of this exponent. On the other hand, a Lyapunov exponent with a low value shows a well-organized situation which is easy to predict. The map of the Lyapunov exponents allows us to identify zones of the airspace where traffic is well organized (requires little monitoring) and zones of disordered traffic. In organized zones, the relative distances between aircraft remain stable over time, giving a stable situation with no modifications in the near future. More details about such complexity metric can be found in [99, 107].

The next section presents a new approach for air traffic complexity metric based on image processing.

4.3 An image processing approach for air traffic complexity metric

4.3.1 Introduction

As for previous metrics, the objective of the our metric is to measure the level of complexity of given traffic situation. Our approach is based on the notion of dominant trajectory also called major flow or main flow. In [109], the definition of major flow is given as follow :

When radar tracks are observed over a long period of time in a dense area, it is very easy to identify major flows connecting major airports. The expression "major flows "is often used but never rigorously defined. Based on an exact trajectory distance and a learning classifier, it is possible to answer the following questions: Given a set of observed trajectories, can we split it into "similar "trajectory classes? If yes, classes with highest number of elements will rigorously define the major flows. Given those classes and a new trajectory, can we tell if it belongs to a major flow and which one? The principle of the major flows definition is to use shape space to represent trajectory shapes as points and to use a shape distance (the shape of a trajectory is the path followed by an aircraft, that is the projection in the 3D space of its 4D trajectory. The speed on the path has no impact).

In order to successfully plan and accommodate the increased number of flights, one must be able to identify major flows in the airspace. In [110, 111], Histon, J.M. *et al.* indicated the importance of the standard flows crossing a sector. They also showed that complex sectors have many entry points and exit points with many interacting flows. The major flows and their interactions constitute the basis for air traffic controllers to build their abstraction of a sector. In the study [112], the authors consider that sector capacity should be based on the geometric distribution of major flows in sectors. A list of flow features is then used to describe traffic flow patterns. Based on such features, they proposed a method to compute the sector capacity. The method avoids measuring controller's workload directly and predefining controller's workload threshold. In [113], major flows are used to study the impact of severe weather. Although analyzing aircraft trajectories is a vital

component of such tools, individual flights scale can either be prohibitively expensive due to the large number of operations, or inappropriate for macroscopic features or trends in big airspace. Hence, it is usual that analysis tools include algorithms that capture and aggregate flights behavior while preserving an appropriate level of fidelity [114].

There are several algorithms used to extract the major flows in a set of trajectories. Some algorithms use traditional data reduction methods (e.g. Principal Component Analysis (PCA)) with clustering methods (e.g. k-means). In [115], Eckstein used PCA and k-means to build a flight taxonomy. Then, Gariel *et al.* have improved this method by increasing data dimensionality (by adding heading, angular position, etc.) and have used the DBSCAN clustering algorithm. The advantage of this algorithm is that it does not require a-priori selection of cluster size and features outlier identification. Marzouli *et al.* [116] also used PCA and DBSCAN to identify flows, from which a mathematical graph (network) was created. Recently, Enriquez and Kurcz [114] proposed another approach based on spectral clustering to identify flows in terminal and en-route airspaces. In [117], Enriquez extended this method to identify flows by including the temporal dimension. Based on the project called *FromDaDy* (which stands for FROM DAta to DisplaY), Marzouli *et al.* [116] have developed a visualization tool which is used to display and extract specific recorded trajectories. A two step algorithm is proposed. The first step uses KDEEB algorithm (which stand for Kernel Density Estimation-based Edge Bundling) to bundle the trajectories into a less cluttered graph. Once this step is implemented, a given graph drawing is transformed into a density map using kernel density estimation. The second step collects flows through a succession of brushing, picking, dropping algorithms.

In this sections, we proposed a different approach which is based on image processing.

4.3.2 Trajectory reconstruction

Before presenting our complexity metric, a brief introduction of trajectory reconstruction is given, which is the first step of our algorithm. This will enable the building of density matrix.

Many interpolation methods have been presented in order to reconstruct trajectory. Delahaye *et al.* [109] have introduced a survey of several reduction models for trajectories.

Given a set of way points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, with $a = x_0 < x_1 < x_2 < \cdots < x_n = b.$, the purpose of interpolation is to construct a shape within the interval $[a, b]$.

The following method aims at solving such problem :

Straight line Segments

One of the easiest way to design trajectory is to use way points connected by straight lines (see Figure 4.1). This easy principle ensures continuity for the trajectory but not for its derivatives. If one want to approximate trajectory with many

shape turns, one have to increase the number of way points in order to reduce the error between the model of the real trajectory.

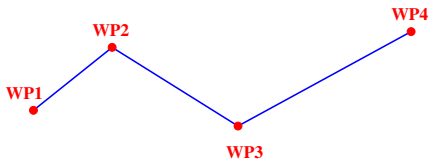


Figure 4.1: Trajectory defined by four way points connected by straight lines.

In order to improve concept Lagrange interpolation process adjust a polynomial function to a given set of way points.

Lagrange interpolation

Given $n+1$ real numbers y_i,$0 \leq i \leq n$, and $n+1$ distinct real numbers $x_0 < x_1 < ... < x_n$, *Lagrange polynomial [120] of degree* n ($L_n(x)$) associated with $\{x_i\}$ and $\{y_i\}$ is a polynomial of degree n solving the interpolation problem :

$$L_n(x_i) = y_i, \quad 0 \leq i \leq n \tag{4.1}$$

$$L_n(x) = \sum_{i=0}^{n} y_i.l_i(x) \tag{4.2}$$

where

$$l_i(x) = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} \tag{4.3}$$

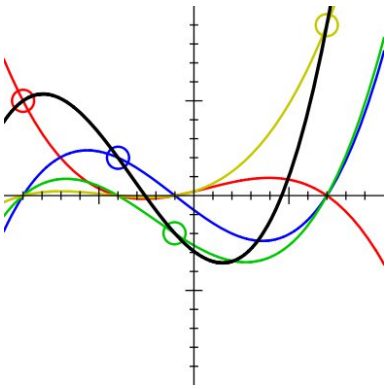


Figure 4.2: $L_n(x)$ is represented by the black curve. The others curves are the polynomials $l_i(x)$.

An example of Lagrange interpolation is given on Figure 4.2 for which four points are interpolated by the black curve which represents $L_4(x)$. The four polynomial functions $\{l_0(x), l_1(x), l_2(x), l_3(x)\}$ are also given by the red, blue, green and yellow curves.

When derivatives have also to be interpolated, Hermite interpolation has to be used.

Hermite interpolation

Hermite interpolation [121] generalizes Lagrange interpolation by fitting a polynomial ($H(x)$) to a function f that not only interpolates f at each knot but also interpolates a given number of consecutive derivatives of f at each knot. This means that the first derivative of the polynomial $H(x)$ have to fit the first derivatives of the function $f(x)$:

$$\left[\frac{\partial^j H(x)}{\partial x^j}\right]_{x=x_i} = \left[\frac{\partial^j f(x)}{\partial x^j}\right]_{x=x_i} \tag{4.4}$$

for all $j = 0, 1, ..., m$ and $i = 1, 2, ..., k$

This means that $n(m+1)$ values

$$\begin{array}{llll} (x_0, y_0), & (x_1, y_1), & \ldots, & (x_{n-1}, y_{n-1}), \\ (x_0, y_0'), & (x_1, y_1'), & \ldots, & (x_{n-1}, y_{n-1}'), \\ \vdots & \vdots & & \vdots \\ (x_0, y_0^{(m)}), & (x_1, y_1^{(m)}), & \ldots, & (x_{n-1}, y_{n-1}^{(m)}) \end{array} \tag{4.5}$$

must be known, rather than just the first n values required for Lagrange interpolation. The resulting polynomial may have degree at most $n(m+1)-1$, whereas the Lagrange polynomial has maximum degree $n-1$.

These interpolation polynomials seem attractive but they both induce oscillations between interpolation points (*Runge 's phenomenon*). Runge's phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree (which is the case for Lagrange and Hermite interpolation). An example of such Runge's phenomenon is given on Figure 4.3 for which Lagrange interpolation has been used.

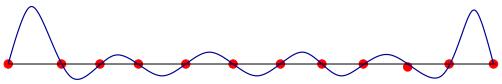


Figure 4.3: Lagrange interpolation result for a set of aligned points.

We can conclude that interpolation with high degree polynomial is risky. In order to avoid this drawback of high degree polynomial interpolation one must use piecewise interpolation.

Piecewise linear interpolation

This is the simplest piecewise interpolation method.

Given $n+1$ real numbers y_i,$0 \leq i \leq n$, and $n+1$ distinct real numbers $x_0 < x_1 < ... < x_n$, we consider the n linear curves $lin_i(x) = a_i x + b_i$ on the intervals $[x_i, x_{i+1}]$ for $i = 0, ...n-1$ ($lin_i(x)$ represent linear functions for which a_i is the slope and b_i a constant).

Each $l_i(x)$ has to connect two points (x_i, y_i), (x_{i+1}, y_{i+1})

$$y_i = a_i x_i + b_i x_i \text{ and } y_{i+1} = a_i x_{i+1} + b_i x_{i+1} \tag{4.6}$$

In order to associate a piecewise formulation of this interpolation method, the following "tent" functions are defined :

$$\psi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{if } x \in [x_{i-1}, x_i] \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{if } x \in [x_i, x_{i+1}] \\ 0 & \text{elsewhere} \end{cases} \tag{4.7}$$

Then,

$$f(x) = \sum_{i=0}^{n} y_i.\psi_i(x). \tag{4.8}$$

An example of such a linear piecewise interpolation is given on Figure 4.4

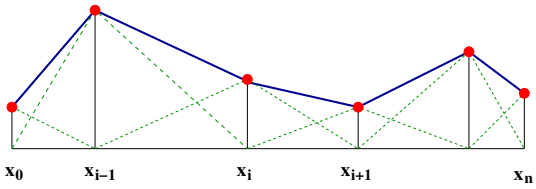


Figure 4.4: Piecewise linear interpolation.

The derivative of the resulting curve is not continuous. In order to fix this drawback, one can use piecewise quadratic interpolation.

Piecewise Quadratic Interpolation

We consider the n quadratic curves $\psi_i(x) = q_i(x) = a_i x^2 + b_i x + c_i$ on the intervals $[x_i, x_{i+1}]$ for $i = 0, ...n-1$. Each $q_i(x)$ has to connect two points (x_i, y_i),(x_{i+1}, y_{i+1}); $\Rightarrow y_i = a_i x_i^2 + b_i x_i + c_i$ and $y_{i+1} = a_i x_{i+1}^2 + b_i x_{i+1} + c_i$. Furthermore, on each point, the derivative of the previous quadratic has to be equal to the derivative of the next one; $\Rightarrow 2a_i + b_i = 2a_{i-1} + b_{i-1}$. For the first segment the term $2a_{i-1} + b_{i-1}$ is arbitrarily chosen (this will affects the rest of the curve). An example of piecewise quadratic interpolation is given on Figure 4.5.

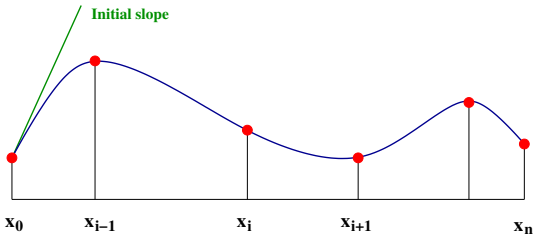


Figure 4.5: Piecewise quadratic interpolation. The shape of the entire curve depend of the choice of the initial slope. Between two points, a quadratic polynomial is fitted.

The main drawback of piecewise quadratic interpolation is linked to the effect induced on the curve by moving on point. As a matter of fact moving one point may totally change the shape of the interpolating curve. The piecewise cubic interpolation avoid this drawback.

Piecewise cubic interpolation

This interpolation is also called Hermite cubic interpolation [122]. For this interpolation :

$$\psi_i(x) = C_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \tag{4.9}$$

and we have the following constraints :

$$C_i(x_i) = y_i \qquad C_i(x_{i+1}) = y_{i+1} \tag{4.10}$$

$$C_i'(x_i) = y_i' = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \quad C_i'(x_{i+1}) = y_{i+1}' = \frac{y_{i+2} - y_i}{x_{i+2} - x_i} \tag{4.11}$$

An example of piecewise cubic interpolation is given on Figure 4.6.

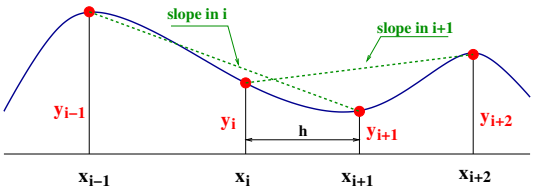


Figure 4.6: Piecewise cubic interpolation. The derivative at point x_i is given by line joining the point (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}). Between two points, a cubic polynomial is fitted. The term h represents the distance two consecutive points.

Moving a point do not affect all the curve which is the main advantage of this interpolation. The resulting curve is C^1 but not C^2 (the second derivative is not continuous). The curvature radius of a curve may be expressed by the following expression :

$$R = \frac{1 + \left(\frac{df(x)}{dx}\right)^{\frac{3}{2}}}{\left|\left(\frac{d^2 f(x)}{dx^2}\right)\right|} \tag{4.12}$$

The piecewise cubic interpolation do not insure that trajectory curvature is continuous which is not adapted for aircraft trajectory mainly in TMA [1] areas and cubic spline interpolation has to be used.

Cubic Spline Interpolation

This method has been developed by General Motor in 1964 [123]. For this piecewise interpolation $\psi_i(x) = S_i(x)$ with the following constraints :

[1] TMA : "Terminal Maneuvering Area"

$$\begin{array}{ll} S_i(x_i) = y_i & S_i(x_{i+1}) = y_{i+1} \\ S_i'(x_i) = S_{i-1}'(x_{i+1}) & S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) \\ S_i''(x_i) = S_{i-1}''(x_{i+1}) & S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}) \end{array} \tag{4.13}$$

One can show that $S_i(x)$ for $x \in [x_i, x_{i+1}]$ is given by :

$$\begin{array}{rcl} S_i(x) & = & \frac{\sigma_i}{6}.\frac{(x_{i+1}-x)^3}{x_{i+1}-x_i} + \frac{\sigma_{i+1}}{6}.\frac{(x-x_i)^3}{x_{i+1}-x_i} \\ & + & y_i.\frac{x_{i+1}-x}{x_{i+1}-x_i} - \frac{\sigma_i}{6}.(x_{i+1}-x_i)(x_{i+1}-x) \\ & + & y_{i+1}.\frac{x-x_i}{x_{i+1}-x_i} - \frac{\sigma_{i+1}}{6}.(x_{i+1}-x_i)(x-x_i) \end{array} \tag{4.14}$$

where

$$\sigma_i = \frac{d^2 S_i(x)}{dx^2}. \tag{4.15}$$

An example of such interpolation is given on Figure 4.7.

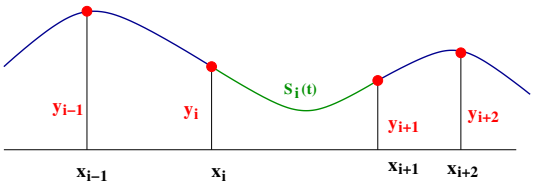


Figure 4.7: Cubic Spline Interpolation.

Such spline is also called natural spline because it represents the curve of a metal spline constrained to interpolate some given points.

When interpolation is not a hard constraint, one can use some control points which change the shape of a given trajectory without forcing this trajectory to go through such control point; such approach is called approximation for which one of the famous methods is the Bézier curve.

Bézier approximation curve

Bézier curves[124] were widely publicized in 1962 by the French engineer Pierre Bézier, who used them to design automobile bodies. But the study of these curves was first developed in 1959 by mathematician Paul de Casteljau using de Casteljau's algorithm [125], a numerically stable method to evaluate Bézier curves. A Bézier curve is defined by a set of control points $\vec{P}_0$ through $\vec{P}_n$, where n is called its order ($n = 1$ for linear, 2 for quadratic, etc.). The first and last control points are always the end points of the curve; however, the intermediate control points (if any) generally do not lie on the curve. Given points $\vec{P}_0$ and $\vec{P}_1$, a linear Bézier curve $\vec{B}(t)$ is simply a straight line between those two points (see Figure 4.8). The curve is given by :

$$\vec{B}(t) = \vec{P}_0 + t(\vec{P}_1 - \vec{P}_0) = (1-t)\vec{P}_0 + t\vec{P}_1 \ , \ t \in [0,1] \tag{4.16}$$

With four points ($\vec{P}_0$, $\vec{P}_1$, $\vec{P}_2$, $\vec{P}_3$), a Bézier curve of degree three can be built. The curve starts at $\vec{P}_0$ going towards $\vec{P}_1$ and arrives at $\vec{P}_3$ coming from the direction of $\vec{P}_2$. Usually, it will not pass through $\vec{P}_1$ or $\vec{P}_2$; these points are only there to provide directional information (see Figure 4.9).

Figure 4.8: Bézier Curve with 2 points.

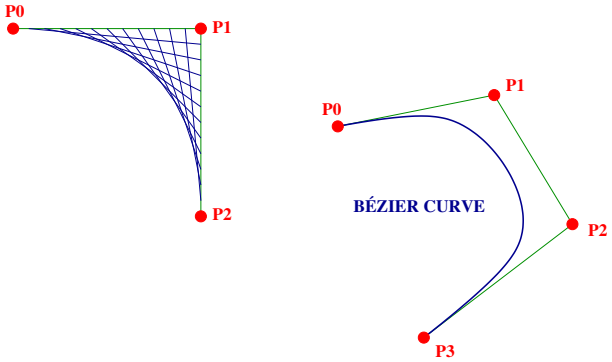


Figure 4.9: Cubic Bézier curve.

Basis spline

B-spline [126] is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. B-splines were investigated as early as the nineteenth century by Nikolai Lobachevsky. A fundamental theorem states that every spline function of a given degree, smoothness, and domain partition, can be uniquely represented as a linear combination of B-splines of that same degree and smoothness, and over that same partition. It is a powerful tool for generating curves with many control points, B stands for basis. A single B-spline can specify a long complicated curve and B-splines can be designed with sharp bends and even "corners". B-Spline interpolation is preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline. Furthermore, spline interpolation avoids the problem of Runge's phenomenon which occurs when interpolating between equidistant points with high degree polynomials.

4.3.2.1 Uniform B-Splines of Degree Zero

We consider a node vector $\vec{T} = \{t_0, t_1, \ldots, t_n\}$ with $t_0 \leq t_1 \leq \ldots \leq t_n$ and n points $\vec{P_i}$. One want to build a curve $\vec{X_0}(t)$ such that :

$$\vec{X_0}(t_i) = \vec{P_i} \tag{4.17}$$

$\Rightarrow \vec{X_0}(t) = \vec{P_i}\ \forall t \in [t_i, t_{i+1}]$.

$$\vec{X_0}(t) = \sum_i B_{i,0}(t).\vec{P_i} \tag{4.18}$$

where

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{elsewhere} \end{cases} \tag{4.19}$$

The shape of the $\vec{X}_0(t)$ function in one dimension is given on Figure 4.10.

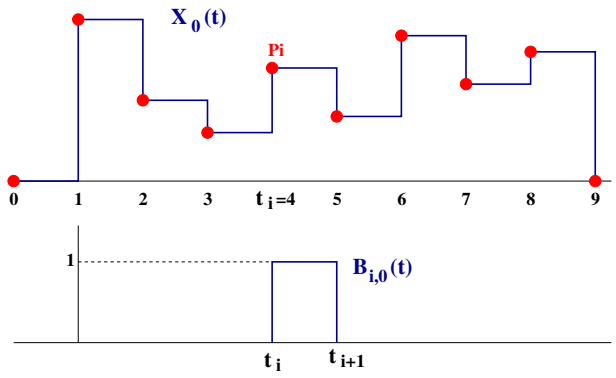


Figure 4.10: Uniform B-Splines of Degree Zero

4.3.2.2 Uniform B-Splines of Degree One

We are searching for a piecewise linear approximation $\vec{X}_1(t)$ for which :

$$\vec{X}_1(t) = \left(1 - \frac{t - t_i}{t_{i+1} - t_i}\right) \vec{P}_{i-1} + \left(1 - \frac{t - t_i}{t_{i+1} - t_i}\right) \vec{P}_i \quad \forall t \in [t_i, t_{i+1}] \tag{4.20}$$

One can write $\vec{X}_1(t)$:

$$\vec{X}_1(t) = \sum_i B_{i,1}(t).\vec{P}_i \tag{4.21}$$

where

$$B_{i,1}(t) = \begin{cases} \frac{t - t_{i-1}}{t_i - t_{i-1}} & \text{if } t \in [t_{i-1}, t_i] \\ \frac{t_{i+1} - t}{t_{i+1} - t_i} & \text{if } t \in [t_i, t_{i+1}] \\ 0 & \text{elsewhere} \end{cases} \tag{4.22}$$

The shape of the $\vec{X}_1(t)$ function in one dimension is given on Figure 4.11.

4.3.2.3 Uniform B-Splines of Degree Three

Those B-Splines have been developed at Boeing in the 70s and represent one of the simplest and most useful cases of B-splines. Degree 3 B-Spline with $n+1$ control points is given by :

$$\vec{X}_3(t) = \sum_{i=0}^{n} B_{i,3}(t).\vec{P}_i \quad 3 \leq t \leq n+1 \tag{4.23}$$

where $B_{i,3}(t) = 0$ if $t \leq t_i$ or $t \geq t_{i+4}$.

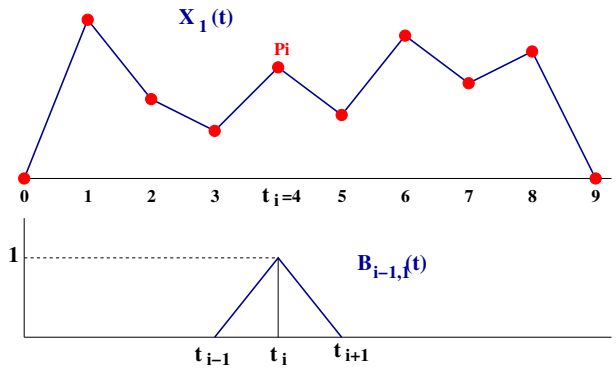


Figure 4.11: Uniform B-Splines of Degree One

$$\vec{X}_3(t) = \sum_{i=j-3}^{j} P_i.B_{i,3}(t) \;\; t \in [j, j+1], \;\; 3 \leq j \leq n \tag{4.24}$$

When a single control point $\vec{P}_i$ is moved, only the portion of the curve $\vec{X}_3(t)$ is changed (with $t_i < t < t_{i+4}$) insuring local control property. The basis functions have the following properties :

- They are translates of each other i.e $B_{i,3}(t) = B_{0,3}(t-i)$
- They are piecewise degree three polynomial
- Partition of unity $\sum_i B_{i,3}(t) = 1$ for $3 \leq t \leq n+1$
- The functions $\vec{X}_i(t)$ are of degree 3 for any set of control points

$$B_{i-2,3}(t) = \frac{1}{h} \begin{cases} (t-t_{i-2})^3 \;\; \text{if } t \in [t_{i-2}, t_{i-1}] \\ h^3 + 3h^2(t-t_{i-1}) + 3h(t-t_{i-1})^2 - 3(t-t_{i-1})^3 \\ \;\; \text{if } t \in [t_{i-1}, t_i] \\ h^3 + 3h^2(t_{i+1}-t) + 3h(t_{i+1}-t)^2 - 3(t_{i+1}-t)^3 \\ \;\; \text{if } t \in [t_i, t_{i+1}] \\ (t_{i+2}-t)^3 \text{ if } t \in [t_{i+1}, t_{i+2}] \\ 0 \text{ otherwise} \end{cases} \tag{4.25}$$

where h is the distance between two consecutive points.

Those basis functions are shown on Figure 4.12.

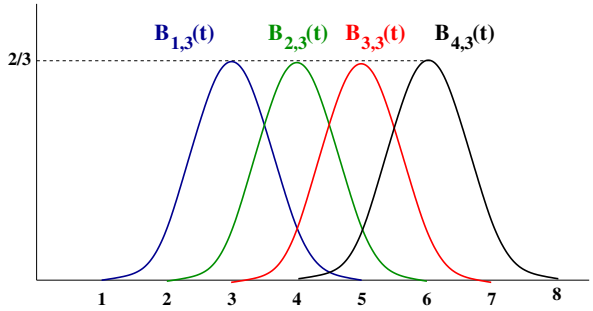


Figure 4.12: Order 3 basis function

4.3.3 Density map generation

This section describes how density map is built. The initial data set consists in a set of trajectories for several days of traffic in the french airspace. Each trajectory is represented by a set of samples gathering positions, time, speed vector and id which are accumulated on a map. We set the size of the map to 800×1000 pixels. In order to increase efficiency of the algorithm, we first interpolate data. Some the previous interpolation models have been tried and compared. For our purposes, cubic spline interpolation has been chosen as it produces the best results in term of error between models and observation. Cubic spline interpolation is easy to implement and produce a curve that appears to be seamless. Furthermore, it is efficient and numerically stable method for determining smooth curves from a set of points. After interpolating, we scale trajectory as a mapping between $[0, 1]$ into $\mathbb{R}^2$ (2D image). Based on this mapping, it is quite simple for building a matrix from such a 2D map. This matrix (Map) will represent the aircraft density of the given airspace and is built with the following process.

Suppose that (x, y) is a grid point of some trajectory. Then

$$\text{Map}[i][j] = \text{Map}[i][j] + 1, \text{ with } i = ceil(y * length), \ j = ceil(x * width),$$

here $length \times width$ is the size of the map. The function *ceil* extract the integer part of a given real number. The Figure 4.13 illustrates how we can build the matrix of density map.

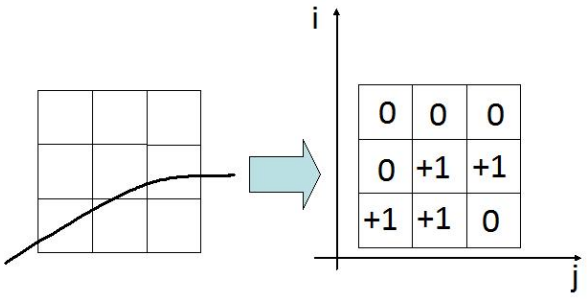


Figure 4.13: Establishing the matrix of density map

4.3.4 Medial axis extraction

After getting the map from the above section it is possible to generate a traffic picture. Figure 4.14 shows the density map of one day of traffic over France. Major air traffic flows clearly appear in the airspace. This image is quite clear but not sufficient identify major flows.

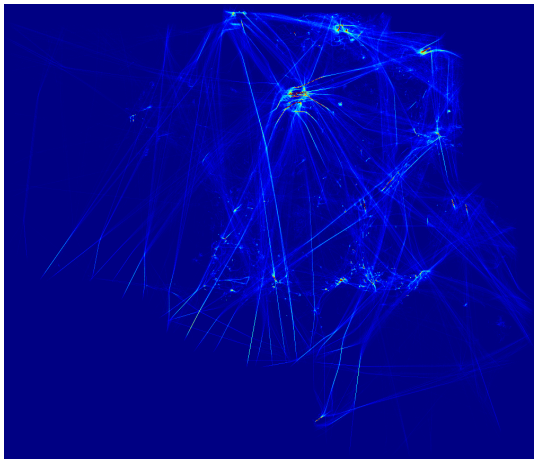
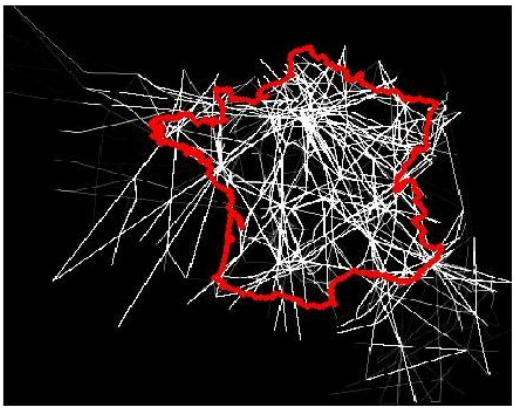


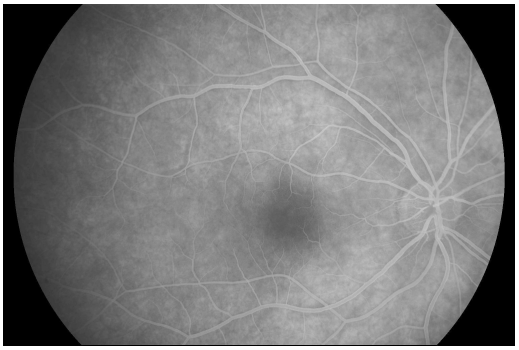
Figure 4.14: Traffic over France generated from the density map

To produce images which are easier to understand and sharper, we scale density map to a gray scale matrix. This allows to extract the flows without losing the structure of the airspace. The Figure 4.15 (a) shows a grey scale image of the traffic over France.

The key idea of our approach is to consider such gray scale image as our former retinal image and to used our previous "vessel" extraction algorithm to identify major flows in the airspace (See Figure 4.15 (b)).



(a) An Image which was created from the gray matrix.



(b) An example of a retinal image

Figure 4.15: Pictures illustrate similarities between the air traffic map and the retinal image

We consider the main flows in the air traffic map as vessels in the eye image. So, the remaining task of medial axis extraction has changed to the task of detection

and measurement of blood vessels in retinal images. We chose the algorithm of the authors Bankhead P *et al.* [59] to get the features of the major flows (the centerlines, the diameters of flow, the directions of flow, etc.). It is based on a flow representation, similar to the vessel extraction in the previous sections.

The input data set contains 104072 trajectories which were collected during one week, between $21st$ and $27th$ October 2013, in France. We applied the algorithm to each day of traffic.

The Figure 4.16 shows that we can effectively apply the method for getting medial axes. It also shows that, almost all main flows in the air traffic were totally extracted.

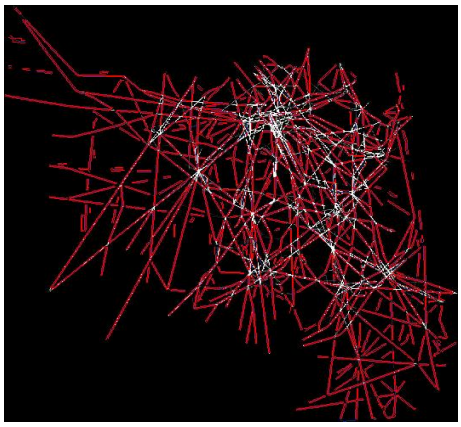
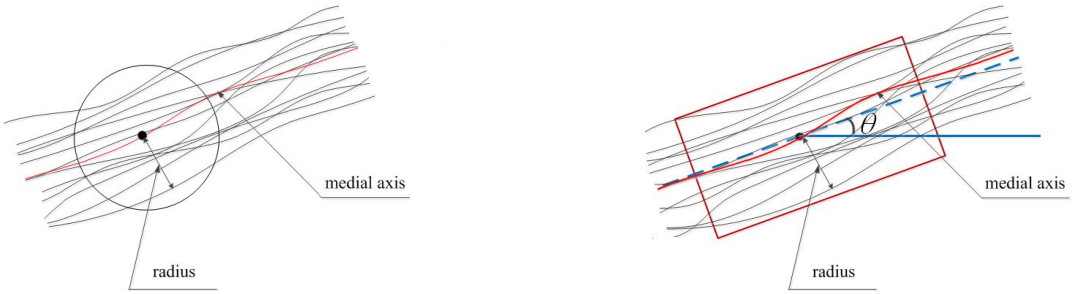


Figure 4.16: Major flows extraction in the French airspace

4.3.5 Application of BV norm to airspace complexity

Based on the algorithms used in the first part of this thesis, we calculate the BV norm along the main flows and use it as a metric of the complexity in the airspace.

After extracting all major flows and their features by using the algorithm described in [59], we calculate the total variation on two different domains (circle and rectangular) (see Figure 4.17 (a) and Figure 4.17 (b)).



(a) BV norm computation on circle domain.

(b) BV norm computation on rectangular domain

Figure 4.17: Pictures illustrate the domain on which BV norm is computed

In the case of a circular domain, the total variation at each point P is given by

$$J(\mathbf{u}) = \sum_{(i,j)\in I} |(\nabla \mathbf{u})_{i,j}| \tag{4.26}$$

where I is the set of double indices corresponding to points in the disc of center P and radius R. Note that $R \geq r$, where r is the radius of flow at P (which is determined by experiment).

In the case of a rectangular domain, the total variation is given by

$$J(\mathbf{u}) = \sum_{(i,j)\in D} |(\nabla \mathbf{u})_{i,j}| \tag{4.27}$$

where D is set of double indices corresponding to points in the rectangle with center P, and width $w \geq 2r$. The domain D is rotated with angle θ, which is the direction of the flow.

We used the color map to represent the value of BV norm at points located along centerlines. The figures below illustrate the results when we calculate the BV norm with different domains and the associated density map.

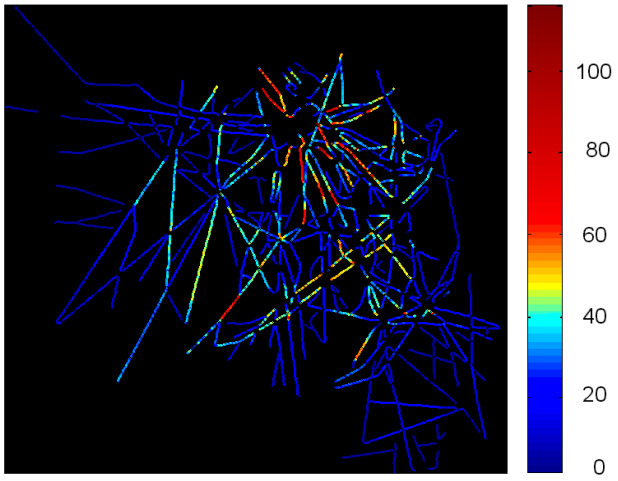


Figure 4.18: BV-norm values computed with circular domain

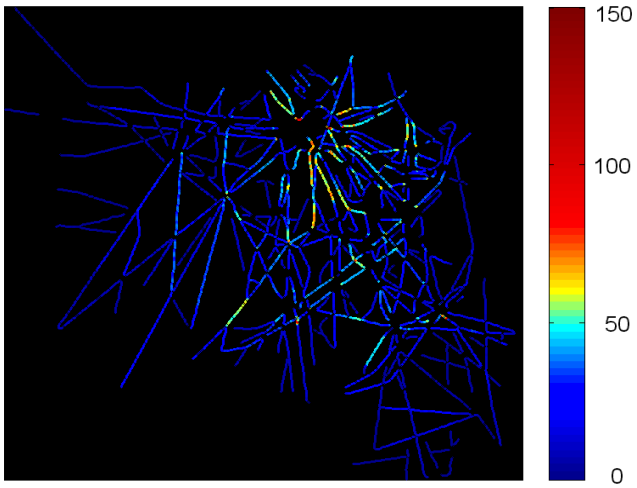


Figure 4.19: Density map

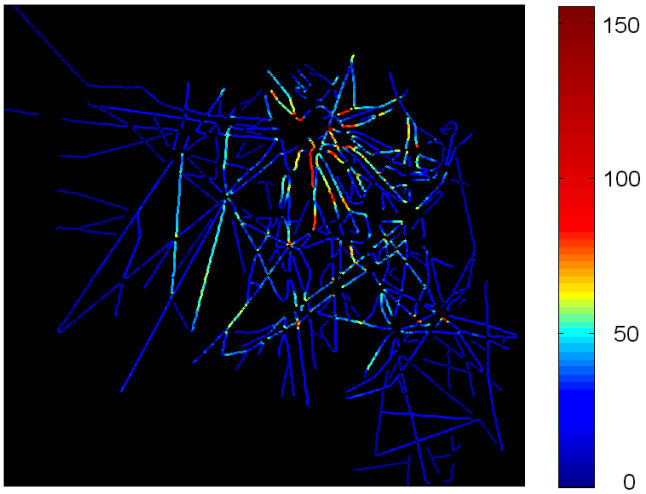


Figure 4.20: BV-norm values computed with rectangular domain

As expected, in both cases, the main complexity is located around Paris area which is known to be the most complex airspace. BV-norm gives more information than density in terms of complexity. In the next section we propose to extend the BV-norm approach on the vector field associated the air traffic. Such vector field will be computed by the mean of aircraft observations (positions and speeds) and a dynamical system model regression.

4.4 Model of air traffic based on dynamical system

The modeling of the set of trajectories by a dynamical system was proposed by Delahaye *et al.* in [118]. The aim is to find a dynamical system which modelizes a vector field as close as possible to the observations given by aircraft positions and

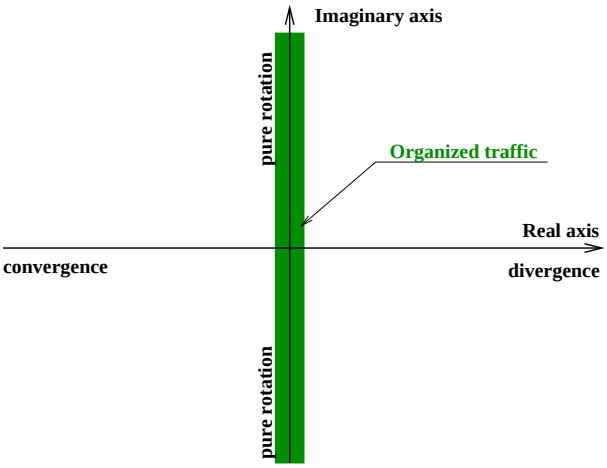


Figure 4.21: Location of the eigenvalues of matrix A. The central rectangle corresponds to organized traffic situations (in pure rotation or in translation).

speeds. It helps to build some metrics of the intrinsic complexity of the distribution of traffic in the airspace.

4.4.1 Linear dynamical systems

This approach consists of modeling a set of trajectories using a linear dynamical system with the following equation:

$$\dot{\vec{X}} = \mathbf{A} \cdot \vec{X} + \vec{B} \tag{4.28}$$

where $\vec{X}$ represents the state vector of the system.

$$\vec{X} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.29}$$

This equation associates a speed vector $\dot{\vec{X}}$ with each point in the state space $\vec{X}$.

The coefficients of matrix A determine the mode of evolution of the system in relation to its dynamics. More precisely, the eigenvalues of this matrix will determine the behavior of the system. Thus, the real part of the eigenvalues indicates whether the system is convergent or divergent. An eigenvalue with a positive real part produces a divergence, and an eigenvalue with a negative real part results in convergence. The absolute value of these real parts is proportional to the level of contraction or expansion of the system. The imaginary part of the eigenvalues shows the tendency of the system to organize itself following a global rotation movement associated with each of the eigen axes.

In the complex plane, it is then possible to identify the locus of the eigenvalues of matrix A associated with organized traffic situations (see Figure 4.21).

Our problem therefore consists of determining the dynamic model which is closest to the observations we have available at a given instant. The least squares method is applied in order to adjust the model to the observations.

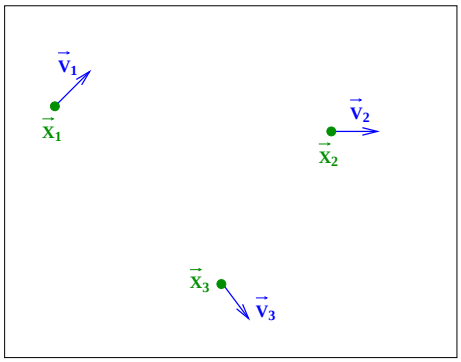


Figure 4.22: Radar captures associated with three aircraft

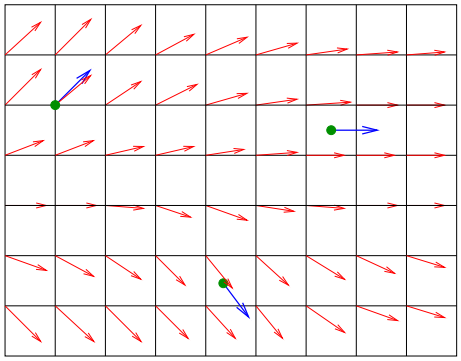


Figure 4.23: Vector field produced by the linear dynamic system

Let N be the number of observations at a given instant (number of airplanes present in a sector at a given instant).

For each of these observations, we have a position measurement (see Figure 4.22):

$$X_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

and a speed measurement:

$$V_i = \begin{bmatrix} vx_i \\ vy_i \\ vz_i \end{bmatrix}$$

We thus wish to find the vector field described by a linear equation ($\dot{\vec{X}} = A.\vec{X} + \vec{B}$) which is best fitted to our observations. To illustrate this aspect, we construct a grid over the airspace (see Figure 4.23) on which we carry out regression of a vector field in such a way as to minimize the error between the model and the observation.

We then construct an error criterion E based on a norm (Euclidean, in our case) which should be minimized in relation to matrix A and vector $\vec{B}$, which represent the parameters of the model:

$$E = \sqrt{\sum_{i=1}^{i=N} \left\| \vec{V_i} - \left(A.\vec{X_i} + \vec{B} \right) \right\|^2}$$

We then introduce the following matrices:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & ... & x_N \\ y_1 & y_2 & y_3 & ... & y_N \\ z_1 & z_2 & z_3 & ... & z_N \\ 1 & 1 & 1 & ... & 1 \end{bmatrix}$$

$$V = \begin{bmatrix} vx_1 & vx_2 & vx_3 & ... & vx_N \\ vy_1 & vy_2 & vy_3 & ... & vy_N \\ vz_1 & vz_2 & vz_3 & ... & vz_N \end{bmatrix}$$

$$C = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix}$$

Criterion E may then be written in the following form:

$$E = \|V - C.X\|_F$$

where $\|\cdot\|_F$ represents the Frobenius norm ($\|A\|_F = \sum_i \sum_j A_{ij}^2$).

Minimizing E is equivalent to minimizing $E^2 = \|V - C.X\|_F^2$. We have

$$E^2 = \|V - C.X\|_F^2 = \mathrm{Tr}(CX - V)(CX - V)^T,$$

where, $\mathrm{Tr}(A)$ is trace of matrix A and it is defined by ($\mathrm{Tr}(A) = \sum_i A_{ii}$). By applying the formula

$$\nabla_M \mathrm{Tr}(AMB + D)(AMB + D)^T = 2A^T(AMB + D)B^T \tag{4.30}$$

and replacing A by I; M by C; B by X; and D by $-V$, we can calculate the gradient of E^2 in relation to matrix C as following:

$$\nabla_C E^2 = 2.(C.X - V).X^T$$

By canceling the above, we obtain: $\nabla_C E^2 = 0 \Leftrightarrow C.X.X^T = VX^T$, which then allows us to calculate C_{opt}:

$$C_{opt} = V.X^T.(X.X^T)^{-1} \tag{4.31}$$

The expression $X^T.(X.X^T)^{-1}$ is the pseudo-inverse of matrix X for which the singular values decomposition is given by :

$$X^T.(X.X^T)^{-1} = L^T.S^{-1}.R$$

where S is the diagonal matrix of the singular values. This decomposition allows us to control conditioning by only inverting singular values which are sufficiently distant from zero. Matrix C is thus given by:

$$C = V.L^T.S^{-1}.R$$

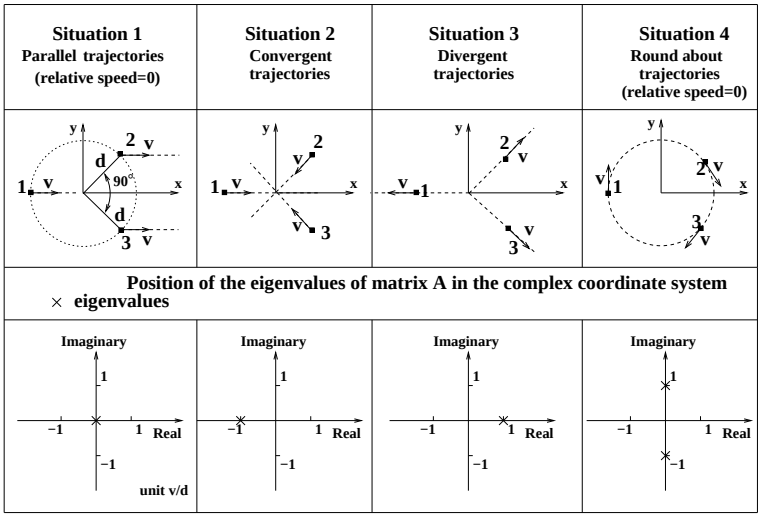


Figure 4.24: Representation of the eigenvalues of matrix A associated with 4 traffic situations.

We then extract matrix A, for which we calculate the associated eigenvalues:

$$A = L.D.U^T.$$

As an example (see Figure 4.24), the eigenvalues of matrix A have been calculated for a situation with three airplanes located on a circle, for which only the orientation of the speed vectors is modified in order to create four traffic situations (organized traffic, convergence, divergence and rotation).

As we see in Figure 4.24, the two organized traffic situations have eigenvalues in the central band.

This approach, based on linear dynamic systems, thus produces a global measurement of the level of organization of a set of trajectories. As the number of degrees of freedom of the linear model is reduced, an error remains between the model and the observation when we increase the number of measurements. In another study [119], D. Delahaye *et al.* proposed an approach which consists of using a local linearization of the underlying dynamic system. This allows to directly extract the gradient of the associated vector field. This approach uses local linear model.

4.4.2 Local linear models

A local approximation method has been developed, which uses only observations close to the evaluation point when calculating regression in order to accelerate the computation of the vector field on the airspace grid. The computation starts by determining the global linear part of the vector field ($\dot{\vec{X}} = A.\vec{X} + \vec{B}$) and by subtracting it from each observations :

$$\vec{v_i} = \vec{V_i} - (A.\vec{X_i} + \vec{B})$$

$i \in \{1, N\}$ where N is the number of observations. $\vec{v_i}$ represents the deviation of the observation from the average field.

The local approximation then consists of seeking a local linear model adjusted to each of the deviations $\vec{v}_i$.

The first order approximation of the spatio-temporal field $\vec{f}$ is given by the following expression:

$$\begin{aligned}\vec{f}(t_0, \vec{X}_0) =& \vec{f}(t, \vec{X}) + \frac{\partial \vec{f}(t, \vec{X})}{\partial t}(t_0 - t) + \\ &+ \frac{\partial \vec{f}(t, \vec{X})}{\partial \vec{X}}(\vec{X}_0 - \vec{X}) + O(|t_0 - t| + \|\vec{X}_0 - \vec{X}\|)\end{aligned}$$

where $\frac{\partial \vec{f}(t,\vec{X})}{\partial t}$ is the temporal derivative of the field $\vec{f}$ and $\frac{\partial \vec{f}(t,\vec{X})}{\partial \vec{X}}$ the associated spatial derivative.

This equation represents a local linear model of field $\vec{f}$ in the vicinity of point $(t, \vec{X})$.

We shall now use this model to compute an approximation of the field based on a set of local observations.

Let us consider a grid point $(t, \vec{X})$ in the state space and look for observations located in its vicinity. The field is regressed in such a way as to minimize the error between the relative deviation $\vec{v}_i(t_i, \vec{X}_i)$ and the local linear model associated with field $\vec{f}$ at the grid point:

$$\begin{aligned}\vec{v}_i(t_i, \vec{X}_i) &\simeq \vec{f}(t, \vec{X}) + \frac{\partial \vec{f}(t, \vec{X})}{\partial t}(t_i - t) + \frac{\partial \vec{f}(t, \vec{X})}{\partial \vec{X}}(\vec{X}_i - \vec{X}) \\ &= \vec{a} + \vec{b}.(t_i - t) + C.(\vec{X}_i - \vec{X})\end{aligned}$$

with $\vec{a} = \vec{f}(t, \vec{X})$, $\vec{b} = \frac{\partial \vec{f}(t,\vec{X})}{\partial t}$ and $C = \frac{\partial \vec{f}(t,\vec{X})}{\partial \vec{X}}$

We now wish to find the vectors $\vec{a}$, $\vec{b}$ and the matrix C to minimize criterion J:

$$\min_{\vec{a},\vec{b},C} J = \sum_{i=1}^{N} \|\vec{v}_i(\vec{X}_i, t_i) - \left\{\vec{a} + \vec{b}(t_i - t) + C.(\vec{X}_i - \vec{X})\right\}\|^2.\psi_i \tag{4.32}$$

where $\psi_i = \psi(t_i - t, \vec{X}_i - \vec{X})$ is a spatio-temporal weighting window used to select observations in the vicinity of a given grid point.

Noting:

$$X = \begin{bmatrix} 1 & 1 & 1 & ... & 1 \\ (t_1 - t) & (t_2 - t) & (t_3 - t) & ... & (t_N - t) \\ (x_1 - x) & (x_2 - x) & (x_3 - x) & ... & (x_N - x) \\ (y_1 - y) & (y_2 - y) & (y_3 - y) & ... & (y_N - y) \\ (z_1 - z) & (z_2 - z) & (z_3 - z) & ... & (z_N - z) \end{bmatrix}$$

$$V = \begin{bmatrix} vx_1 & vx_2 & vx_3 & ... & vx_N \\ vy_1 & vy_2 & vy_3 & ... & vy_N \\ vz_1 & vz_2 & vz_3 & ... & vz_N \end{bmatrix}$$

$$M = \begin{bmatrix} a_x & b_x & C_{xx} & C_{xy} & C_{xz} \\ a_y & b_y & C_{yx} & C_{yy} & C_{yz} \\ a_z & b_z & C_{zx} & C_{zy} & C_{zz} \end{bmatrix}$$

and

$$\Psi = \begin{bmatrix} \sqrt{\psi_1} & 0 & 0 & ... & 0 \\ 0 & \sqrt{\psi_2} & 0 & ... & 0 \\ 0 & 0 & \sqrt{\psi_3} & ... & 0 \\ 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & ... & \sqrt{\psi_N} \end{bmatrix}$$

as $\|A\|_F^2 = \mathrm{Tr}(AA^T)$, then criterion J takes the form:

$$\begin{aligned} J &= \|(M \cdot X - V)\Psi\|_F^2 \\ &= \mathrm{Tr}(M \cdot X - V)\Psi \cdot (M \cdot X - V) \cdot \Psi)^T \\ &= \mathrm{Tr}(MX\Psi^2X^TM^T) - \mathrm{Tr}(MX\Psi^2V^T) - \mathrm{Tr}(V\Psi^2X^TM^T) + \mathrm{Tr}(V\Psi^2V^T) \\ &= J_1 - J_2 - J_3 + J_4, \end{aligned} \tag{4.33}$$

where, $J_1 = \mathrm{Tr}[MX\Psi^2X^TM^T]$, $J_2 = \mathrm{Tr}[MX\Psi^2V^T]$, $J_3 = \mathrm{Tr}[V\Psi^2X^TM^T]$, and $J_4 = \mathrm{Tr}[V\Psi^2V^T]$.

We will now calculate each term of (4.33).

Using the formula

$$\nabla_X \mathrm{Tr}(XBX^T) = XB^T + XB,$$

we have

$$\begin{aligned} \nabla_M J_1 = \nabla_M \mathrm{Tr}(MX\Psi^2X^TM^T) &= M(X\Psi^2X^T)^T + M(X\Psi^2X^T) \\ &= 2M(X\Psi^2X^T). \end{aligned} \tag{4.34}$$

Applying the formula

$$\nabla_X \mathrm{Tr}(XA) = A^T,$$

it follows that

$$\nabla_M J_2 = \nabla_M \mathrm{Tr}(MX\Psi^2V^T) = (X\Psi^2V^T)^T = V\Psi^2X^T. \tag{4.35}$$

As

$$\nabla_X \mathrm{Tr}(AX^T) = A,$$

then

$$\nabla_M J_3 = \nabla_M (V\Psi^2X^TM^T) = V\Psi^2X^T. \tag{4.36}$$

Finally, taking the derivative of a constant

$$\nabla_M J_4 = \nabla_M \mathrm{Tr}(V\Psi^2V^T) = \mathbf{0} \tag{4.37}$$

By taking now the derivative of J with respect to matrix M, and taking into consideration (4.34)-(4.37), we obtain:

$$\begin{aligned} \nabla_M J &= \nabla_M J_1 - \nabla_M J_2 - \nabla_M J_3 + \nabla_M J_4 \\ &= 2M(X\Psi^2X^T) - V\Psi^2X^T - V\Psi^2X^T + \mathbf{0} \\ &= 2[M(X\Psi^2X^T) - V\Psi^2X^T] \end{aligned} \tag{4.38}$$

By setting the derivative of J vanish, we get

$$M_{opt} = V \cdot \Psi^2 \cdot X^T (X \cdot \Psi^2 \cdot X^T)^{-1} \tag{4.39}$$

Having M_{opt} it is possible to extract the parameters of the local models: vectors $\vec{a}$, $\vec{b}$ and matrix C.

Based on this algorithm, we can easily compute a smooth vector field which fit exactly with observations. Such vector field will now be used to build air traffic complexity metric.

4.4.3 Computation of local vectorial total variation norm of vector field.

We now propose to compute a traffic complexity metric by using local vectorial total variation norm of the relative deviation vector field which is expressed in the previous Section 4.4.2. In what follows, we introduce a complete method to compute the local vectorial total variation norm of relative deviation vector field. This indicator will provide a local measurement of disorder of the field, taking into account the relative deviation.

As mentioned in Section 1.2, several methods to define the vectorial total variation norm of vector-valued function have been proposed. As TV_J can be derived from the generalized Jacobians from geometric measure theory, within the context of this theory, it is the most natural form of a vectorial total variation. It can be shown that, in the case of differentiable $\mathbf{u}$, the vectorial total variation can be obtained by computing the integral over the largest singular value of the derivative matrix. However, when different methods are compared in the context of numerical analysis (in terms of computational complexity), TV_F is preferable to another methods. Because of its good performance, TV_F has emerged as a favorite candidate for vectorial TV. Therefore, in our experiments, we chose the approach based on the Frobenius Norm to calculate the vectorial total variation norm of vector field. TV_F can be computed by using the following integral :

$$TV_F = \int_{\Omega_{X^*}} \|D\mathbf{u}(\mathbf{x})\| \mathrm{d}\mathbf{x}, \tag{4.40}$$

where, $X^* = [x^*, y^*, z^*]^T$ is the considered point, and Ω_{X^*} is the neighborhood of X^*.

In order to compute such integral (4.40),we will now develop numerical method.

Let $\tau = V_k{}_{i=1}^k$ be a partition of Ω_{X^*} i.e. a system of measurable sets V_i such that $\cup_{i=1}^k V_i = \Omega_{X^*}$ and $m(V_i \cap V_j) = 0$, $\forall i \neq j$, $i, j = 1, \ldots, k$, $m(V)$ is a measure of a set V. For simplicity , we use the regular partition, $m(V_i) = m(V_j) = m \; \forall i, j = 1, \ldots, k$. The integral sum is built as follow:

$$S_I = \sum_{i=1}^{k} \|D\mathbf{u}(X_i)\|_F \cdot m, \tag{4.41}$$

where $X_i \in V_i$ is an arbitrary point. The summation (4.41) can be used to approximate the integral (4.40). In practice, we set $m = 1$.

Let us now describe the practical algorithm for computing complexity maps.

1) Regression of the global linear dynamic system $(A \cdot \vec{X}) + \vec{B}$. (as in Subsection 4.4.1)

2) Computation of relative observations $(\vec{V_i} - (A \cdot \vec{X} + \vec{B}))$.

3) For each grid point $\vec{X}(t)$ of a cube of airspace, carry out:

   i) Computation of the local linear model $(\vec{a}, \vec{b}, C)$ (as in the Subsection 4.4.2)

   ii) Computation vectorial total variation norm (as expression above).

We have tested the proposed method in three simulated traffic situations. In each situation, the performance can be described as follow. First, at the time t, we filter the data set to get data in the interval $[t - t_{minus}; t + t_{plus}]$, in the experiment, we chose $t_{minus} = 25s$ and $t_{plus} = 15s$. In our artificial data, the time step period is set to $3s$.

We then solve Equation (4.31) to obtain the global linear model $(A, \vec{B})$. Next, we solve the the problem (4.32) to get the vector field and its derivative at grid points. In this step, a space-time weight function has been used to select the observations which are close to the grid point. Because the data set has been filtered by time, the weight function now only depends on space. This function is chosen as below

$$\Psi_i(t_i - t, \vec{X_i} - \vec{X}) = \begin{cases} 1 - \frac{\|\vec{X_i} - \vec{X}\|}{R} & \text{if } \frac{\|\vec{X_i} - \vec{X}\|}{R} < 1 \\ 0 & \text{otherwise} \end{cases}$$

where R is the radius of the grid point neighborhood.

The main task now is to compute the optimal solution which has been given in (4.39). Finally, the complexity indicator is obtained by calculating vectorial total variation norm of the vector field which has been expressed above. We chose the vicinity of the given point $(\vec{X})$ as a square, which is divided by a partition of 7×7 small boxes. Here, we consider the problem in the two-dimensional (2D) context. A three-dimensional $(3D)$ form can easily be extended from this algorithm. The points X_i in the integral sum (4.41) are the centroids of the small squares. Figure 4.25 depicts the vicinity of a given point.

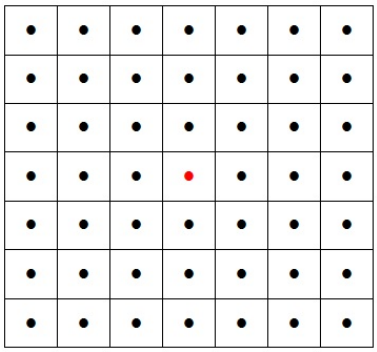


Figure 4.25: The vicinity of a given point which is marked by red color

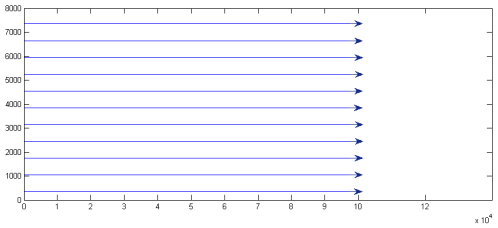
In the next subsection, we present some very good results to affirm the fitness of the model.

4.4.4 Results

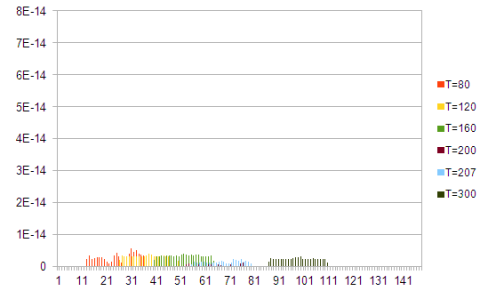
We investigated three traffic situation. Those are *parallel case*, *face to face case* and convergent flows case. The results are computed at the flow centerline.

Parallel case

In this case, we created 44 aircraft. Those aircraft form a parallel flow without conflicts. This is depicted in the Figure 4.26 (a).



(a) A parallel flow with no conflict



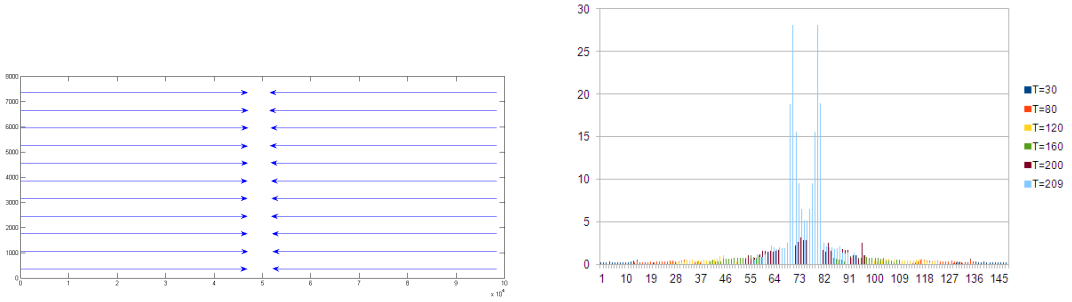
(b) Complexity of the parallel case. The vertical axis shows complexity values and the horizontal axis gives the distance along the flow.

Figure 4.26: Parallel situation

The Figure 4.26 (b) shows that, the parallel flow case does not generate complexity at all. All values are smaller than $1E-14$. Only the remaining computational error is shown on the figure.

Face to face case

In the second situation, we created 2 opposite flows. Each flow contains 22 aircraft that are set parallel without conflict as shown on Figure 4.27 (a). The Figure 4.27 (b) shows the complexity which is highest at time when the conflict occurs. The complexity value gradually increases when the flow moves towards the conflict points (the two flows will collide at the time $t = 225s$).

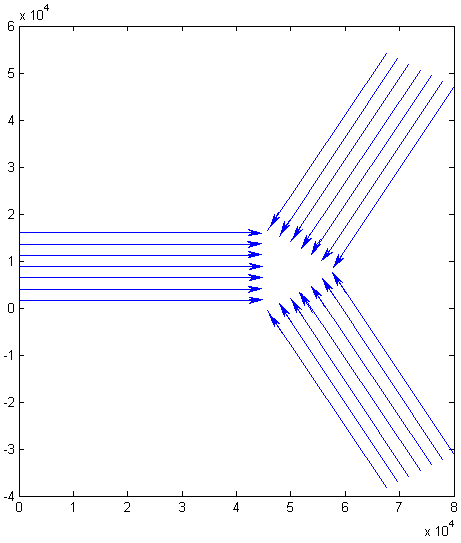


(a) Two opposite flows (b) Complexity of face to face case

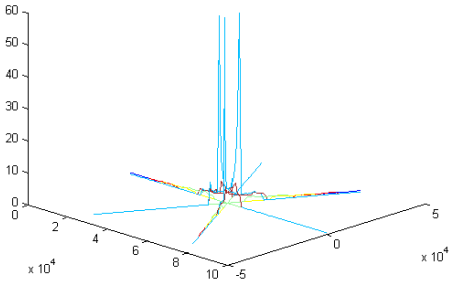
Figure 4.27: Face to face situation

Convergent 3 flows case

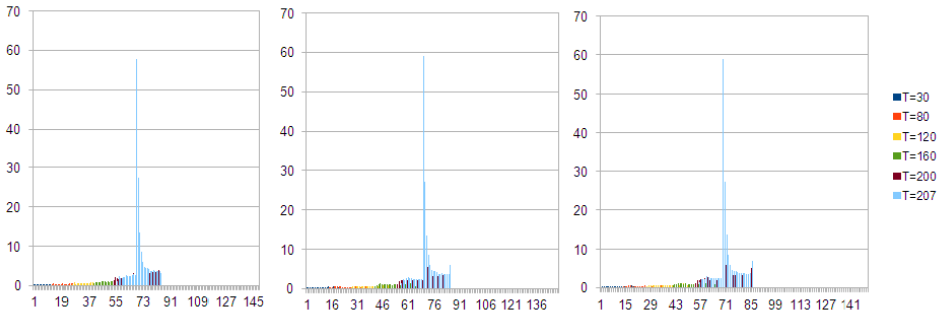
In this case, we created 3 flows that move towards the same point. Each flow includes 21 aircraft with no conflict. The Figure 4.28 (a) depicts the convergent situation. The result of this case shows that, the value of complexity increases when the aircraft fly towards the conflict points. This result also indicates that the complexity of three convergent flows is higher than the face to face case.



(a) Three convergent flows



(b) 3D complexty representation



(c) Complexities of three convergent flows

Figure 4.28: Convergent situation

Conclusion

This thesis has two main contributions. The first one establish an automatic algorithm for classification of vasculitis in multiple sclerosis fundus angiography. In the second one, we have extracted an airspace complexity indicator based previous results on image processing.

The thesis includes three parts. It starts with the background that provides us preliminary knowledge required for the part 2 and part 3.

In chapter 1, we introduce an overview of total variation as well as its applications in image processing. A discrete version of TV used in our algorithm has been presented. Then, we introduced an extension of scalar total variation norm to vector-valued functions which has been applied in the air traffic management application (part 3).

In chapter 2, a tutorial on Support Vector Machines has been presented. It starts with the concepts of VC dimension and structural risk minimization. Then, SVMs for separable and non-separable data have been introduced based on structural risk minimization. At the end of the chapter, we described kernel methods in details which are used for non-separable data. Kernel methods are very efficient for real world data analysis problems requiring nonlinear methods. It allows us to avoid computing dot product in the high-dimensional feature space.

In the second part of this thesis, we propose a method for classification of retinal images. Our method helps ophthalmologists for the diagnosis of vasculitis in multiple sclerosis fundus angiography. First, we provide a review of methods for segmentation and measurement of blood vessel in retinal image that is an important step in our method. Based on BV norm calculated at each point along centerline, we detect the diseased region in the pathological images. A feature extraction strategy was introduced to be used in SVMs model. The resulting set of features was then used to represent the input image set in terms of feature vectors. Standard SVM classifier was applied to classify images. The reported evaluation indicated that the proposed method worked well and also produce good results.

The third part addresses an Air Traffic Management application. Based on the ideas developed in the second part, we introduced a method to extract the main flows in the airspace which critical for ATM application. Based on algorithms used in the second part, we developed an airspace complexity indicator which could be used at macroscopic level. The results show that, such indicator is better than the regular density metric which is computed just by counting the number of aircraft.

Finally, by using a dynamical system model of air traffic, we propose a method for developing a new traffic complexity metric based on the computation of the local vectorial total variation norm of the relative deviation vector field. By investigating

three different traffic situations, the obtained results are quite relevant with what is expected in operation.

Appendix A

Applications of Total Variation

Although there are some applications of total variation in the field of differential equations (eg. the TVD scheme introduce in [18]), we just focus here on the applications in images processing.

A.1 The ROF model

The use of TV as a regularizer has been shown to be very effective for processing images because of its ability to preserve edges. The denoising unconstrained model is defined by Rudin et *al.* as follows.

$$\inf_{u \in L^2(\Omega)} \int_\Omega |\nabla u| + \mu \int_\Omega (u - f)^2 d\mathbf{x} \tag{A.1}$$

Here, Ω is the image domain, $f : \Omega \longrightarrow \mathbb{R}$ is the observed noisy image, $u : \Omega \longrightarrow \mathbb{R}$ is the denoised image, and $\mu \geq 0$ is a parameter depending on the noise level.

The first term is the total variation which is a measure of the amount of oscillation in the resulting image u. It is given by

$$TV(u) = \int_\Omega |\nabla u| \tag{A.2}$$

Its minimization would reduce the amount of oscillation which presumably reduces noise.

The second term is the L^2 distance between u and f, which encourages the denoised image to inherit most features from the observed data. Thus the model trades off the closeness to f by gaining the regularity of u. The noise is assumed to be additive and Gaussian with zero mean. If the noise variance level σ^2 is known, then the parameter μ can be treated as the Lagrange multiplier. Being introduced for different reasons, several variants of TV can be found in the literature, restraining the resulting image to be consistent with the known noise level, i.e., $\int_\Omega (u - f)^2 = |\Omega|\sigma^2$.

A.2 Total variation based image deblurring

Image deblurring is fundamental in making pictures sharp and useful. Like denoising, it frequently arises in imaging sciences and technologies, including optical,

medical, and astronomical applications, and is often a crucial step towards successful detection of important patterns such as abnormal tissues or the surface details of some distant planets [20]. Image deblurring can be extended from ROF model as follows.

$$\min_u \left\{ \int_\Omega |Du| + \frac{1}{2} \int_\Omega (Au - f)^2 d\mathbf{x} \right\}, \tag{A.3}$$

where $\Omega \subset \mathbb{R}$ is the domain of the image and A is a linear operator and is called the blurring kernel. In this model, f is formulated as the sum of a Gaussian noise v and a blurry image Au resulting from the linear blurring operator A acting on the clean image $\bar{u}$, i.e., $f = Au + v$. The existence and uniqueness of the optimal deblurred estimation is studied by Chambolle and Lions [22]. The following conditions will be assumed for the study of existence.

a) Observation $f \in L^2(\mathbb{R}^2) \bigcap L^\infty(\mathbb{R}^2)$.

b) Image $u \in BV_2(\mathbb{R}^2)$, and $\|u\|_{L^\infty} \leq \|f\|_{L^\infty}$

c) Kernel $A \in BV(\mathbb{R}^2)$, nonnegative, and satisfies the DC-condition $A[1] \equiv 1$, treating $1 \in L^\infty(\mathbb{R})$. Here, DC stands for *direct current* since the Fourier transform of a constant contains no nonzero frequencies.

Computationally, model (A.3) can be implemented via many different algorithms [22, 23]. Some numerical results which demonstrate the performance of the above model show that the model is very competitive.

A.3 Total Variation Based Inpainting

Inpainting technique has found use in many applications such as restoration of old films, object removal in digital photos, red eye correction, super resolution, compression, image coding and transmission. The word inpainting is an artistic synonym for image interpolation and has been used for quite a while among museum restoration artists [24]. It was first transplanted into digital image processing in the remarkable work by Bertalmio *et al.* [25], which has stimulated the recent wave of interest in numerous problems related to image interpolation, including the works by Chan and Shen and their collaborators [20]. Image inpainting refers to the filling-in of missing or occluded regions in an image based on information available on the observed regions. The aim of inpainting might be to use the background information for restoring damaged portions of an image or for removing unwanted elements that are presented in the image.

Several successful inpainting models have been proposed that are based upon the Bayesian, variational, PDE, wavelet approach, such as Masnou and Morel [26] and Bertalmio *et al.* [25]. The TV inpainting model which uses variational methods in inpainting is proposed by Chan and Shen in [27]. The TV inpainting model is to find the solution of the the boundary value problem:

$$\min_u \int_\Omega |\nabla u| \text{ subject to } \frac{1}{\text{Area}(\Omega \setminus D)} \int_{\Omega \setminus D} |u - u_0|^2 d\mathbf{x} = \sigma^2. \tag{A.4}$$

Here, D is the inpainting region with piecewise smooth boundary Γ, u_0 is the observed image whose value in D is missing, we assume that $u_0|_{\Omega\setminus D}$ is contaminated by homogeneous white noise (modeled by the Gaussian distribution), σ is the standard deviation of the white noise.

Thus, the TV inpainting method simply fills-in the missing region such that the TV in Ω is minimized. The use of TV-norm is desirable because it has the effect of extending level sets into D without smearing discontinuities along the tangential direction of the boundary of D [15].

As practiced in the variational methodology, it is more convenient to solve the unconstrained TV inpainting problem

$$\min_u \int_\Omega |\nabla u| + \lambda \int_{\Omega\setminus D} |u - u_0|^2 d\mathbf{x}, \tag{A.5}$$

where λ plays the role of the Lagrange multiplier for the constrained variational problem (A.4).

By defining the masked Lagrange multiplier

$$\lambda_D(\mathbf{x}) = \lambda \cdot \mathbb{1}_{\Omega\setminus D}(\mathbf{x}), \tag{A.6}$$

the Euler-Lagrange equation for (A.5) can be written as

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + 2\lambda_D(u - u_0) = 0 \tag{A.7}$$

which has the same form as that in the ROF model, except that the regularization is switching between 0 and λ in different regions.

Compared with all the other variational inpainting schemes, the TV model has the lowest complexity and easiest digital implementation. It works remarkable well for all local inpainting problems such as digital zoom-in and text removal [21].

A.4 Image Segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels). Its goal is to partition a given image into a collection of "objects", built upon which other high-level tasks such as object detection, recognition, and tracking can be further performed. In this section, we cite the survey by Chan *et al.* [29]. It presents in detail the recent results in Total Variation based image segmentation.

TV minimization problems also arise from image segmentation. When one seeks for a partition of the image into homogeneous segments, it is often helpful to regularize the shape of the segments. This can increase the robustness of the algorithm against noise and avoid spurious segments. It may also allow the selection of features of different scales. In the classical Mumford-Shah model [28], the regularization is done by minimizing the total length of the boundary of the segments. In this case, if a segment is represented by its characteristic function, then the length of its boundary is exactly the TV of the characteristic function. Therefore, the minimization of length becomes the minimization of TV of characteristic functions.

Given an observed image u on an image domain Ω, the piecewise constant Mumford-Shah model seeks a set of curves C and a set of constants $\mathbf{c} = (c_1, c_2, \ldots, c_L)$ which minimize the energy functional given by:

$$F^{MS}(C, \mathbf{c}) = \sum_{l=1}^{L} \int_{\Omega_l} [u(\mathbf{x}) - c_l]^2 d\mathbf{x} + \beta \cdot \text{Length}(C). \tag{A.8}$$

The curves in C partition the image into L mutually exclusive segments Ω_l for $l = 1, 2, \ldots, L$. The idea is to partition the image, so that the intensity of u in each segment Ω_l is well approximated by a constant c_l. The goodness-of-fit is measured by the L^2 difference between u and c_l. On the other hand, a minimum description length principle is employed which requires the curves C to be as short as possible. This increases the robustness to noise and avoids spurious segments. The parameter $\beta > 0$ controls the trade-off between the goodness-of-fit and the length of the curves C.

The Mumford-Shah objective is non-trivial to optimize especially when the curves need to be split and merged. Chan *et al.* [30] proposed a level set-based method which can handle topological changes effectively. In the two-phase version of this method, the curves are represented by the zero level set of a Lipschitz level set function ϕ defined on the image domain. The objective function then becomes

$$\begin{aligned} F^{CV}(\phi, c_1, c_2) = &\int_{\Omega} H(\phi(\mathbf{x}))[u(\mathbf{x}) - c_1]^2 d\mathbf{x} \\ &+ \int_{\Omega} [1 - H(\phi(\mathbf{x}))][u(\mathbf{x}) - c_2]^2 d\mathbf{x} + \beta \int_{\Omega} |\nabla H((\phi))| \end{aligned} \tag{A.9}$$

The function H is the Heaviside function defined by $H(\mathbf{x}) = 1$ if $x \geq 0$, $H(\mathbf{x}) = 0$ otherwise. In practice, we replace H by a smooth approximation H_ε, e.g.,

$$H_\varepsilon = \frac{1}{2}\left[1 + \frac{2}{\pi} \arctan\left(\frac{x}{\varepsilon}\right)\right].$$

Although this method simplifies splitting and merging of curves, the energy functional is non-convex which possesses many local minima. These local minima may correspond to undesirable segmentations [31].

Interestingly, for fixed c_1 and c_2, the above non-convex objective can be reformulated as a convex problem, so that a global minimum can be easily computed [32, 33]. The globalized objective is given by

$$F^{CEN} = \int_{\Omega} \left((u(\mathbf{x}) - c_1)^2 - (u(\mathbf{x}) - c_2)^2\right) \phi(\mathbf{x}) d\mathbf{x} + \beta \int_{\Omega} |\nabla \phi| \tag{A.10}$$

which is minimized over all ϕ satisfying the bilateral constraints $0 \leq \phi \leq 1$, and all scalars c_1 and c_2. After a solution ϕ is obtained, a global solution to the original two-phase Mumford-Shah objective can be obtained by thresholding ϕ with μ for almost every $\mu \in [0, 1]$, see [32, 33]. This problem is exactly a TV denoising problem with bound constraints. Some other proposals for computing global solutions can be found in [31].

Appendix B

Karush-Kuhn-Tucker conditions

B.1 Notation

Definition B.1.1 (Convex Set) A set X in a vector space is called convex if for any $\mathbf{x}, \mathbf{x}' \in X$ and any $\lambda \in [0,1]$, we have

$$\lambda\mathbf{x} + (1-\lambda)\mathbf{x}' \in X \tag{B.1}$$

Definition B.1.2 (Convex Function) A function f defined on a set X (note that X need not be convex itself) is called convex if, for any $\mathbf{x}$, $\mathbf{x}' \in X$ any $\lambda \in [0,1]$ such that $\lambda\mathbf{x} + (1|\lambda)\mathbf{x}' \in X$, we have

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{x}') \leq \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{x}') \tag{B.2}$$

A function f is called strictly convex if for $\mathbf{x} \neq \mathbf{x}'$ and $\lambda(0,1)$ (B.2) is a strict inequality.

The following lemma shows the relationship between convex set and convex function.

Lemma B.1 *Denote by $f : \mathcal{X} \longrightarrow \mathbb{R}$ a convex function on a convex set $\mathcal{X}$. Then the set*

$$X := \{\mathbf{x} | \mathbf{x} \in \mathcal{X} \text{ and } f(x) \leq c\} \, \forall c \in \mathbb{R} \tag{B.3}$$

is convex.

Proof. For any $\mathbf{x}$ and $\mathbf{x}' \in X$ we have $f(\mathbf{x}) \leq c$ and $f(\mathbf{x}') \leq c$, since $\mathcal{X}$ is convex so $\lambda\mathbf{x} + (1-\lambda)\mathbf{x}' \in \mathcal{X}$ for any $\lambda \in [0,1]$. Moreover,

$$\begin{aligned} f(\lambda\mathbf{x} + (1-\lambda)\mathbf{x}') &\leq \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{x}') \\ &\leq \lambda c + (1-\lambda)c = c. \end{aligned}$$

Hence we have $\lambda\mathbf{x} + (1-\lambda)\mathbf{x}' \in X$. □

Theorem B.2 (Minima on Convex Sets) *If the convex function $f : \mathcal{X} \longrightarrow \mathbb{R}$ has a minimum on a convex set $X \subset \mathcal{X}$, then its arguments $\boldsymbol{x} \in X$, for which the minimum value is attained, form a convex set. Moreover, if f is strictly convex, then this set will contain only one element.*

Corollary (Constrained Convex Minimization) *Given the set of convex functions f, $c_1, ..., c_k$ on the convex set $\mathcal{X}$, the problem*

$$\begin{aligned} &\min_{\mathbf{x}} \ f(\mathbf{x}) \\ &\text{subject to } c_i(\mathbf{x}) \leq 0 \\ &\qquad\qquad i = 1, \ldots, k. \end{aligned} \tag{B.4}$$

has as its solution a convex set, if a solution exists. This solution is unique if f is strictly convex.

The Support Vector problem has the same formulation as this problem. However, in practice c_i is written as positivity constraints by using concave functions and this can be fixed by a sign change [9]. In some cases, we additionally have equality constraints $e(x) = 0$ for some $j = 1, \ldots, k'$. Then the optimization problem can be written as

$$\begin{aligned} &\min_{\mathbf{x}} \ f(\mathbf{x}) \\ &\text{subject to } c_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \ldots, k, \\ &\qquad\qquad e_j(\mathbf{x}) = 0 \text{ for } j = 1, \ldots, k'. \end{aligned} \tag{B.5}$$

B.2 Optimization conditions

Theorem B.3 (Equality Constraints) *Assume an optimization problem of the form (B.5), where $f, c_i, ej : \mathbb{R}^n \longrightarrow \mathbb{R}$ for $i = 1, \ldots, k$ and $j = 1, \ldots, k'$ are arbitrary functions, and a Lagrangian*

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_i c_i(\mathbf{x}) + \sum_{j=1}^{k'} \beta_j e_j(\mathbf{x}) \text{ for } \alpha_i \geq 0 \text{ and } \beta_j \in \mathbb{R} \tag{B.6}$$

If a triple of variables $(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}})$ with $\bar{\mathbf{x}} \in \mathbb{R}^n$, $\bar{\boldsymbol{\alpha}} \in [0, \infty)^k$, and $\bar{\boldsymbol{\beta}} \in \mathbb{R}^{k'}$ exists such that for all $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\alpha} \in [0, \infty)^k$, and $\boldsymbol{\beta} \in \mathbb{R}^{k'}$,

$$L(\mathbf{x}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}) \leq L(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}) \leq L(\bar{\mathbf{x}}, \boldsymbol{\alpha}, \boldsymbol{\beta}), \text{ (Saddle point)} \tag{B.7}$$

then $\bar{\mathbf{x}}$ is a solution to (B.5)

Definition B.2.1 Function

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_i c_i(\mathbf{x}) + \sum_{j=1}^{k'} \beta_j e_j(\mathbf{x}) \text{ for } \alpha_i \geq 0 \text{ and } \beta_j \in \mathbb{R} \tag{B.8}$$

is called the Lagrangian of the problem B.5,

$$g(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\mathbf{x}} \ L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{B.9}$$

is Lagrange dual function and

$$\begin{aligned} &\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \ g(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &\text{subject to } \alpha_i \geq 0 \\ &\qquad\qquad i = 1, \ldots, k. \end{aligned} \tag{B.10}$$

is dual problem.

B.2.0.1 Some important properties

- The dual problem is always convex.
- The primal and dual optimal values, f^* and g^*, always satisfy weak duality: $f^* \geq g^*$. Indeed, Let $\bar{\mathbf{x}}$ be a primal feasible solution and $\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}$ be a dual feasible solution. Then

$$\begin{aligned} g(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}) &= \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \sum_{i=1}^{k} \bar{\alpha}_i c_i(x) + \sum_{j=1}^{k'} \bar{\beta}_j e_j(\mathbf{x}) \right\} \\ &\leq f(\bar{\mathbf{x}}) + \sum_{i=1}^{k} \bar{\alpha}_i c_i(\bar{\mathbf{x}}) + \sum_{j=1}^{k'} \bar{\beta}_i e_j(\bar{\mathbf{x}}) \\ &\leq f(\bar{\mathbf{x}}). \end{aligned}$$

  Definition B.2.2 Given primal feasible $\bar{\mathbf{x}}$ and dual feasible $\bar{\boldsymbol{\alpha}}; \bar{\boldsymbol{\beta}}$, the quantity $f(\bar{\mathbf{x}}) - g(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}})$ is called the duality gap between $\bar{\mathbf{x}}$ and pair $(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}})$.
- Slater's condition: for convex primal, if there is an $\mathbf{x}$ such that $c_i(\mathbf{x}) < 0$ for $i = 1, \ldots, k$ and $e_j(\mathbf{x}) = 0$ for $j = 1, \ldots, k'$ then strong duality holds: $f^* = g^*$.

From now on we make the assumption: the Primal problem (B.5) is convex and its objective function is differentiable.

Definition B.2.3 (Karush-Kuhn-Tucker conditions) Given general problem

$$\begin{aligned} &\min_{\mathbf{x}} \ f(\mathbf{x}) \\ &\text{subject to } c_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \ldots, k, \\ &\qquad\qquad e_j(\mathbf{x}) = 0 \text{ for } j = 1, \ldots, k'. \end{aligned}$$

The Karush-Kuhn-Tucker conditions or KKT conditions are:

$$\partial_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \partial_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_i \partial_{\mathbf{x}} c_i(x) + \sum_{j=1}^{k'} \beta_j \partial_{\mathbf{x}} e_j(\mathbf{x}) = 0 \text{ (stationarity)} \quad \text{(B.11a)}$$

$$\alpha_i c_i(x) = 0 \text{ for all } i \text{ (complementary slackness)} \quad \text{(B.11b)}$$

$$c_i(x) \geq 0; \ e_j(x) = 0 \text{ for all } i; j \text{ (primal feasibility)} \quad \text{(B.11c)}$$

$$\alpha_i \geq 0 \text{ for all } i \text{ (dual feasibility)} \quad \text{(B.11d)}$$

Theorem B.4 (**Necessary optimality condition**) *If $\boldsymbol{x}^*$ and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ are primal and dual solutions, with zero duality gap, then $\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ satisfy the KKT conditions.*

Proof. Let $\mathbf{x}^*$ and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ be primal and dual solutions with zero duality gap (strong duality holds, e.g., under Slater's condition). Then,

$$\begin{aligned} f(\mathbf{x}^*) &= g(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \\ &= \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_i^* c_i(\mathbf{x}) + \sum_{j=1}^{k'} \beta_j^* e_j(\mathbf{x}) \\ &\leq f(\mathbf{x}^*) + \sum_{i=1}^{k} \alpha_i^* c_i(\mathbf{x}^*) + \sum_{j=1}^{k'} \beta_j^* e_j(\mathbf{x}^*) \\ &\leq f(\mathbf{x}^*) \end{aligned}$$

In other words, all these inequalities are equalities. So

$$\sum_{i=1}^{k} \alpha_i^* c_i(\mathbf{x}^*) + \sum_{j=1}^{k'} \beta_j^* e_j(\mathbf{x}^*) = 0$$
$$\Longrightarrow \sum_{i=1}^{k} \alpha_i^* c_i(\mathbf{x}^*) = 0,$$

this follows from $\sum_{j=1}^{k'} \beta_j^* e_j(\mathbf{x}^*) = 0$.

Note that $\alpha_i^* \geq 0$ and $c_i(\mathbf{x}^*) \leq 0$ we deduce $\alpha_i^* c_i(\mathbf{x}^*) = 0$ for all $i = 1, \ldots, k$. This is complementary slackness.

Primal and dual feasibility evidently hold. Finally, because point $\mathbf{x}^*$ minimizes $L(\mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ over $\mathbf{x} \in \mathbb{R}^n$. Hence the derivative of $L(\mathbf{x}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ with respect to $\mathbf{x}$ must be 0 at $\mathbf{x} = \mathbf{x}^*$. This is exactly the stationary condition. □

Theorem B.5 (The sufficient optimality condition) *If $\boldsymbol{x}^*$ and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ satisfy the KKT conditions, then $\boldsymbol{x}^*$ and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ are primal and dual solutions.*

Proof. If there exists $\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ that satisfy the KKT conditions, from stationary condition (B.11a), we have

$$g(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) = f(\mathbf{x}^*) + \sum_{i=1}^{k} \alpha_i^* c_i(\mathbf{x}^*) + \sum_{j=1}^{k'} \beta_j^* e_j(\mathbf{x}^*) \tag{B.12}$$

On the other hand, from complementary slackness condition (B.11b) and feasible conditions, we get

$$f(\mathbf{x}^*) + \sum_{i=1}^{k} \alpha_i^* c_i(\mathbf{x}^*) + \sum_{j=1}^{k'} \beta_j^* e_j(\mathbf{x}^*) = f(\mathbf{x}^*) \tag{B.13}$$

Comparison of (B.12) and (B.13) shows that

$$f(\mathbf{x}^*) = g(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*). \tag{B.14}$$

Therefore duality gap is zero (and $\mathbf{x}^*$ and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ are primal and dual feasible) so $\mathbf{x}^*$ and $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$ are primal and dual optimal. □

Bibliography

[1] T. Mitchell. *Machine Learning*. McGraw-Hill International, 1997 Vol . 177.

[2] LipoWang(Ed.) *Support Vector Machines: Theory and Applications*, (2005) ISBN 3-540-24388-7

[3] Burbidge, Robert, and Bernard Buxton. *An introduction to support vector machines for data mining.* Keynote papers, young OR12 (2001): 3-15.

[4] Vapnik, Vladimir N., and A. Ya Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities." *Theory of Probability & Its Applications 16*, no. 2 (1971): 264-280.

[5] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[6] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995

[7] Christopher J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 1998.

[8] Lin, Keng-Pei, and Ming-Syan Chen. "Efficient kernel approximation for large-scale support vector machine classification." *Proceedings of the Eleventh SIAM International Conference on Data Mining*. 2011.

[9] B. Schölkopf and A. Smola. *Learning with kernels*. MIT Press, Cambridge, MA, 2002.

[10] Anthony, Martin, and Norman Biggs. *PAC learning and artificial neural networks.* MIT Press, 1995.

[11] Oliver Chapelle, *Support Vector Machines: Induction Principles, Adaptive Tuning and Prior Knowledge*, Thesis, 2004.

[12] Cucker, Felipe, and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint.* No. 24. Cambridge University Press, 2007.

[13] Sridhar, Banavar, Kapil S. Sheth, and Shon Grabbe. "Airspace complexity and its application in air traffic management." *2nd USA/Europe Air Traffic Management R&D Seminar*. 1998.

[14] D. L. Cohn, *Measure theory*. Birkhäuser, Boston 1993.

[15] T. Chan, *et al. Recent Developments in Total Variation Image Restoration*, Springer Verlag, In *Mathematical Models of Computer Vision*, 2005.

[16] E. Giusti, *Minimal surfaces and functions of bounded variation.* Birkhäuser, Boston, 1984.

[17] Rudin, Leonid I., Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms." *Physica D: Nonlinear Phenomena 60,* no. 1 (1992): 259-268.

[18] Harten, Ami. "High resolution schemes for hyperbolic conservation laws." *Journal of computational physics* 49.3 (1983): 357-393.

[19] Goldluecke, Bastian, *et al.* "The natural vectorial total variation which arises from geometric measure theory." *SIAM Journal on Imaging Sciences* 5.2 (2012): 537-563.

[20] Chan Tony F, Shen Jianhong (Jackie), *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*, Society of Industrial and Applied Mathematics. (2005) ISBN 0-89871-589-X

[21] Chan, Tony F., and Jianhong Jackie Shen. "Variational image inpainting." *Communications on pure and applied mathematics* 58.5 (2005): 579-619.

[22] Chambolle, Antonin, and Pierre-Louis Lions. "Image recovery via total variation minimization and related problems." *Numerische Mathematik* 76.2 (1997): 167-188.

[23] Acar, Robert, and Curtis R. Vogel. "Analysis of bounded variation penalty methods for ill-posed problems." *Inverse problems* 10.6 (1994): 1217.

[24] G. Wahba. *The Ravished Image.* St. Martin's Press, New York, 1985.

[25] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. *Image Inpainting.* In computer Graphics (SIGGRAPH 2000), 2000.

[26] Masnou, Simon, and J-M. Morel. "Level lines based disocclusion." *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on.* IEEE, 1998.

[27] Shen, Jianhong, and Tony F. Chan. "Mathematical models for local nontexture inpaintings". *SIAM Journal on Applied Mathematics* 62.3 (2002): 1019-1043.

[28] Mumford, David, and Jayant Shah. "Optimal approximations by piecewise smooth functions and associated variational problems." *Communications on pure and applied mathematics* 42.5 (1989): 577-685.

[29] Chan R., Chan T., Yip A."Numerical Methods and Applications in Total Variation Image Restoration". *Handbook of Mathematical Methods in Imaging*, Springer-Verlag Berlin Heidelberg, 2011.

[30] Chan, Tony F., *et al.* "Superresolution image reconstruction using fast inpainting algorithms." *Applied and Computational Harmonic Analysis* 23.1 (2007): 3-24.

[31] Law Y, Lee H, Yip A. "A multi-resolution stochastic level set method for Mumford-Shah image segmentation". *IEEE Trans Image Process* 17(12):2289-2300, 2008.

[32] Chan T, Esedoḡlu S, Nikolova M. "Algorithms for finding global minimizers of image segmentation and denoising models". *SIAM J Appl Math* 66(5):1632-1648, 2006

[33] Strang, Gilbert. "Maximal flow through a domain." *Mathematical Programming* 26.2 (1983): 123-143.

[34] Cumming, Christine. "The Total Variation Approach to Approximate Hyperbolic Wave Equations".

[35] Bresson, Xavier, and Tony F. Chan. "Fast dual minimization of the vectorial total variation norm and applications to color image processing." *Inverse Problems and Imaging* 2.4 (2008): 455-484.

[36] Di Zenzo, Silvano, "A note on the gradient of a multi-image." *Computer vision, graphics, and image processing* 33.1 (1986): 116-125.

[37] Sapiro, Guillermo. "Vector-valued active contours." *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on.* IEEE, 1996.

[38] Julie Le Scanff *et al.*, "Uveitis associated with multiple sclerosis", *Multiple sclerosis* 2007; 00:1-3

[39] Marchomichelakis N . "Multiple sclerosis in Foster S and itale A," (eds). *Diagnosis and treatment of uveitis. philadelphia: WB. Sanders company*, 2002.

[40] Jordan JF *et al.*, "Intermediate uveitis in childhood preceding the diagnosis of multiple sclerosis," *Am ophthalmol* 2003; 135:885-6.

[41] BirchMK *et al.*, "Retinal venous sheathing and blood-retinal barrier in multiple sclerosis", *Arch Ophtalmol* 1996; 114:34-9.

[42] Bachman DM *et al.*, "Granulomatous uveitis in neurological disease", *Br J Ophtalmol* 1985; 69:192-6.

[43] Patt M *et al.*, "Vascularites rétiniennes proliférantes et slérose en plaque", *J Fr Ophtalmmol* 2003; 26,4:381-5.

[44] Pederson JE *et al.*, "Pathology of parsplanitis", *Am J Ophtalmol* 1986:762-774.

[45] Vine AK *et al.*, "Severe periphlebitis, peripheral retina ischemia, and preretinal neovascularization in patients with ultiple sclerosis", *Am . J Ophtalmol* 1992; 113:28-32.

[46] Nussenblatt RB *et al.*, "Fundamentals and clinical practice", 2 ed . St Louis: Mosby, 199; 354-63

[47] Jabs, D. A., R. B. Nussenblatt, and J. T. Rosenbaum. "Standardization of uveitis nomenclature for reporting clinical data. Results of the First International Workshop." *American journal of ophthalmology* 140.3 (2005): 509-516.

[48] Miller, David, *et al.*, "Clinically isolated syndromes suggestive of multiple sclerosis, part I: natural history, pathogenesis, diagnosis, and prognosis." *The Lancet Neurology* 4.5 (2005): 281-288.

[49] Wild, Sarah, *et al.*, "Global prevalence of diabetes estimates for the year 2000 and projections for 2030." *Diabetes care* 27.5 (2004): 1047-1053.

[50] Eye Diseases Prevalence Research Group. "The prevalence of diabetic retinopathy among adults in the United States." *Archives of Ophthalmology* 122.4 (2004): 552.

[51] S.P Harding, D.M Broadbent, C. Neoh, *et al.*, "Sensitivity and specificity of photography and direct ophthalmology in screening for sight threatening eye disease: the Liverpool Diabetic Eye study", *British Medical Journal*, 311, 1131, 1995.

[52] Herbert F. Jelinek, Michael J. Cree, "Automated image detection of retinal pathology" *New York : CRC Press*, 2010.

[53] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating the blood vessel in retinal images by piecewise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 203-310, 2000.

[54] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum, "Detection of blood vessels in retinal images using two-dimensional matched filters," *IEEE Transactions on Medical Imaging*, vol, 8, pp. 263-369, 1989.

[55] O. Brinchmann-Hansen and O. Engvold, "Microphotometry of the blood column and light streak on retinal vessels in fundus photographs," *Acta Ophthalmologica Supplement*, vol. 179, pp. 9-19, 1986.

[56] O. Brinchmann-Hansen and H. Heier, "Theoretical relationships between light streak characteristics and optical properties of retinal vessels," *Acta Ophthalmological Supplement*, vol. 179, pp. 33-37, 1986

[57] P.H. Gregson, Z. Shen, R.C. Scott, and V. Kozousek, "Automated grading of venous beading," *Computers and Biomedical Research*, vol, 28, pp. 291-304, 2000.

[58] N. Chapman, N. Witt, X. Goa, A. Bharath, A. V. Stanton, S. A. Thom, and A. D. Hughes, "Computer algorithms for the automated measurements of retinal arteriolar diameters," *British Journal of Ophthalmology*, vol. 85, pp. 75-79, 2001

[59] Bankhead P, Scholfield CN, McGeown JG, Curtis TM "Fast Retinal vessel Detection and Measurement Using Wavelets and Edge Location Refinement," *FloS ONE* 7(3): e32435. doi: 10.1371/journal.pone.0032435, 2012

[60] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, Vol. 60, pp. 259-268, 1992

[61] Starch JL, Murtagh F, "Astronomical image and signal processing," *IEEE Signal Process Mag* 18: 30-34,2001

[62] Olivo-Marin JC, "Extraction of spots in biological images using multiscale products," *Pattern Recognit* 35: 1989-1996, 2002

[63] Starck JL, Fadili J, Murtagh F, "The undecimated wavelet decomposition and its reconstruction," *IEEE Trans Signal Process* 16: 297-309, 2007.

[64] Vermeer KA, Vos FM, Lemij HG, Vossepoel AM, "A model based method for retinal blood vessel detection," *Comput Biol Med* 34: 209-219, 2004.

[65] Lee ETY, "Choosing nodes in parametric curve interpolation," *Computer-Aided Design* 21: 363-370, 1989.

[66] Chapman N, Witt N, Gao X, Bharath AA, Stanton AV, *et al.*, "Computer algorithms for the automated measurement of retinal arteriolar diameters," *Br J Ophthalmol* 85: 74-79, 2001.

[67] Herier H, Brinchmann-Hansen O "Releable measurements from fundus photographs in the presence of focusing errors," *Invest Ophthalmol Vis Sci*, 30: 674-677, 1989

[68] Kaushik S, Tan AG, Mitchell P, Wang JJ "Prevalence and associations of enhanced retinal arteriolar light reflex: A new look at an old sign," *Ophthalmologica*, 114: 113-120, 2007.

[69] Metelitsina TI, Grunwald JE, DuPont JC, Ying GS, Liu C "Effect of viagra on retinal vein diameter in AMD patients," *Exp Eye Res* 83: 128-132.

[70] Brinchmann-Hansen O, Heier H, "Theoretical relations between light streak characteristics and optical properties of retinal vessels," *Ophthalmologica supplement*, Vol. 179: 33-37, 1986

[71] Li H, Hsu W, Lee ML, Wong TY, "Automatic grading of retinal vessel caliber," *IEEE Trans Biomed Eng* 52: 1352-1355, 2005.

[72] Lowell J, Hunter A, Steel D, Basu A, Ryder R, *et al.*, "Measurement of retinal vessel widths from fundus images based on 2-D model," *IEEE Trans Med Imaging* 23: 1196-1204, 2004

[73] Chang, Chih-Chung and Lin, Chih-Jen,"A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, pp: 1-27, 2011 (Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm)

[74] M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. Rudnicka, C. Owen, S. Barman, "Blood vessel segmentation methodologies in retinal images—a survey", *Computer Methods and Programs in Biomedicine, 108.1* (2012): 407-433.

[75] E. Ricci and R. Perfetti, "Retinal blood vessel segmentation using line operators and support vector classification", *IEEE Trans. Med. Image., vol. 26*, no. 10, Oct. 2007.

[76] L. Gang, O. Chutatape, S.M. Krishnan, "Detection and measurement of retinal vessels in fundus images using amplitude modified second-order Gaussian filter", *IEEE Transactions on Biomedical Engineering 49* (2002) 168–172.

[77] B. Zhang, L. Zhang, L. Zhang, F. Karray, "Retinal vessel extraction by matched filter with first-order derivative of Gaussian", *Computers in Biology and Medicine 40* (2010) 438–445.

[78] M. Al-Rawi, M. Qutaishat, M. Arrar, "An improved matched filter for blood vessel detection of digital retinal images", *Computers in Biology and Medicine 37* (2007) 262–267.

[79] M. Amin, H. Yan, *High speed detection of retinal blood vessels in fundus image using phase congruency*, Soft Computing – A Fusion of Foundations, Methodologies and Applications (2010) 1–14.

[80] I. Liu, Y. Sun, Recursive tracking o vascular networks in angiograms based on the detection-deletion scheme, *IEEE Transactions on Medical Imaging 12* (1993) 334–341.

[81] Z. Liang, M.S. Rzeszotarski, L.J. Singerman, J.M. Chokreff, The detection and quantification of retinopathy usingdigita angiograms, *IEEE Transactions on Medical Imaging13* (1994) 619–626.

[82] Delibasis, Konstantinos K., *et al.*, Automatic model-based tracing algorithm for vessel segmentation and diameter estimation. *Computer methods and programs in biomedicine 100.2* (2010): 108-122.

[83] O. Chutatape, Z. Liu, S.M. Krishnan, Retinal blood vessel detection and tracking by matched Gaussian and Kalman filters, in: *Engineering in Medicine and Biology Society,1998. Proceedings of the 20th Annual International Conference of the IEEE, vol.* 3146, 1998, pp. 3144–3149.

[84] P. Kelvin, H. Ghassan, A. Rafeef, Live-vessel: extending live wire for simultaneous extraction of optimal medial andboundary paths in vascular images, in: *Proceedings of the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer-Verlag, Brisbane, Australia, 2007.

[85] F. Zana,, and J-C. Klein. "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation." *Image Processing, IEEE Transactions on 10.7* (2001): 1010-1019.

[86] A.M. Mendonca, A. Campilho, Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction, *IEEE Transactions on Medical Imaging 25* (2006) 1200–1213.

[87] Y. Yang, S. Huang, N. Rao, "An automatic hybrid method for retinal blood vessel extraction" 2008, *International Journal of Applied Mathematics and Computer Science 18* 399–407.

[88] K. Sun, Z. Chen, S. Jiang, Y. Wang, "Morphological multiscale enhancement, fuzzy filter and watershed for vascular tree extraction in angiogram" 2010, *Journal of Medical Systems.*

[89] M.S. Miri, A. Mahloojifar, "Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction" (2011), *IEEE Transactions on Biomedical Engineering 58* 1183–1192.

[90] K. Akita, H. Kuga, "A computer method of understanding ocular fundus images", *Pattern Recognition 15* (1982)431–443.

[91] R. Nekovei and Y. Sun, "Back-propagation network and its configuration for blood vessel detection in angiograms," *IEEE Trans. Neural Net-works*, vol. 6, pp. 64–72, Jan. 1995.

[92] Sinthanayothin, C., *et al.*, "Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images". *British Journal of Ophthalmology*, 1999. 83(8): p. 902-910.

[93] Staal, J., **et al.**, "Ridge-based vessel segmentation in color images of the retina." *IEEE Transactions on Medical Imaging*, 2004. 23(4): p. 501-509.

[94] STARE: STructured Analysis of the Retina, *http://www.ces.clemson.edu/ ahoover/stare/.* 2000

[95] DRIVE: Digital Retinal Images for Vessel Extraction *http://www.isi.uu.nl/Research/Databases/DRIVE/*

[96] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, Jr., H. F. Jelinek, and M.J. Cree,"Retinal vessel segmentation using the 2D Gabor wavelet and supervised classification," *IEEE Trans. Med. Image.*, vol. 25, no. 9, pp.1214–1222, Sep. 2006.

[97] Salem, S., N. Salem, and A. Nandi, "Segmentation of retinal blood vessels using a novel clustering algorithm (RACAL) with a partial supervision strategy." *Medical and Biological Engineering and Computing,* 2007. 45(3): p. 261-273.

[98] D. Marin, A. Aquino, ME. Gegundez-Arias and JM. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using grey-level and moment invariants-based features," *IEEE Trans. Med. Imaging*, vol. 30, no. 1, pp. 146-158, 2011.

[99] M. Prandini *et al.*, "Toward Air Traffic Complexity Assessment in New Generation Air Traffic Management Systems", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12 , NO. 3, pp. 809-818, Sep. 2011. tems." *Journal of Aerospace Operations 1*, no. 3 (2012): 281-299.

[100] Ing. Andrea Ranieri *Combinatorial Exchange Models for a User-Driven Air Traffic Flow Management in Europe*, PhD thesis, University of Trieste.

[101] Sridhar, B., Seth, K.S., Grabbe, S., "Airspace complexity and its application in air traffic management metric", 2nd *USA/EUROPE ATM R&D seminar*, Orlando, December 1998.

[102] P. Flener, J. Pearson, M. Agren, C. Garcia-Avello1, M. Celiktin, and S. Dissing, "Air traffic complexity resolution in multi-sector planning using constraint programming". In *Air Traffic Management R&D Seminar*, 2007.

[103] S. Mondoloni and D. Liang. "Airspace fractal dimension and applications". In Eurocontrol-FAA, editor, *Fourth USA/EUROPE Air Traffic Management R&D Seminar*, 2001.

[104] K. Lee, E. Feron, and A. Pritchett, "Describing airspace complexity: Airspace response to disturbances," *J. Guid. Control Dyn.*, vol. 32, no. 1, pp. 210–222, Jan./Feb. 2009.

[105] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed-integer programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 3–11, Mar. 2002.

[106] D. Delahaye and S. Puechmorel, "Air traffic complexity: Towards intrinsic metrics," *in Proc. 3rd FAA/Eurocontrol Air Traffic Manag. R&D Semin.*, Napoli, Italy, Jun. 2000

[107] S. Puechmorel and D. Delahaye. "New trends in air traffic complexity". In *ENRI International Workshop on ATM/CNS (EIWAC 2009)*, Tokyo, Japan, March 2009.

[108] I.V Laudeman, S.G Shelden, R Branstrom, and C.L Brasil, *Dynamic density : an air traffic management metric*, Tech. report, NASA TM-1998-112226, 1998.

[109] Delahaye, Daniel, *et al.*, "Mathematical models for aircraft trajectory design: a survey." *EIWAC 2013, 3rd ENRI International Workshop on ATM/CNS.*

[110] Histon, J. M., R. J. Hansman, B. Gottlieb, H. Kleinwaks, S. Yenson, D. Delahaye, and S. Puechmorel. "Structural considerations and cognitive complexity in air traffic control." In *Digital Avionics Systems Conference, 2002. Proceedings.* The 21st, vol. 1, pp. 1C2-1. IEEE, 2002.

[111] Histon, Jonathan M., R. John Hansman, Guillaume Aigoin, Daniel Delahaye, and Stephane Puechmorel. "Introducing structural considerations into complexity metrics." (2002).

[112] L. Song, D. Greenbaum, and C. Wanke, "Predicting sector capacity for traffic flow management decision support,"in *6 th AIAA Aviation Technology, Integration and Operations Conference*, 2006.

[113] L. Song, D. Greenbaum, and C. Wanke, "The impact of severe weather on sector capacity," in *8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*, Napa, California, USA, 2009.

[114] Enriquez, Marco, and Christopher Kurcz. "A simple and robust flow detection algorithm based on spectral clustering." *ICRAT Conference*. 2012.

[115] Eckstein, Adric. "Automated flight track taxonomy for measuring benefits from performance based navigation." *Integrated Communications, Navigation and Surveillance Conference, 2009. ICNS'09.* IEEE, 2009.

[116] Marzuoli, Aude, Vlad Popescu, and Eric Feron. "Two perspectives on graph-based traffic flow management." *First SESAR Innovation Days* (2011).

[117] Enriquez, Marco. "Identifying Temporally Persistent Flows in the Terminal Airspace via Spectral Clustering." *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)*. 2013.

[118] Delahaye, Daniel, *et al.* "Air traffic complexity map based on non linear dynamical systems." *Air Traffic Control Quarterly* 12.4 (2004).

[119] Delahaye, Daniel, and Stephane Puechmorel. "Air traffic complexity based on dynamical systems." *Decision and Control (CDC), 2010 49th IEEE Conference on.* IEEE, 2010.

[120] H. Jeffreys and B. S. Jeffreys, *Methods of mathematical physics*, Cambridge University Press, 1988.

[121] R.H Bartels, J.C Beatty, and B.A Barskyn, *An introduction to splines for use in computer graphics and geometric modeling*, Computer graphics, Morgan Kaufmann, 1998.

[122] T Hearth, M, *Scientific computing, an introductory survey*, Computer graphics, McGraw-Hill, 2002.

[123] Birkhoff and C de Boor, *Piecewise polynomial interpolation and approximation*, Proceeding of the General Motors Symposium of 1964, General Motors, 1964.

[124] Farin, Gerald, and Hansfordand Dianne, *The essentials of cagd*, A K Peters, Ltd, 2000.

[125] G Farin, *Curves and surfaces for computer aided geometric design. a practical guide*, Academic Press, 1993.

[126] C de Boor, *A practical guide to splines*, Springer-Verlag, 1978.