UNIVERSIDAD DE GUANAJUATO

División de Ciencias Naturales y Exactas

EFFICIENT OBJECT CLASSIFICATION USING THE EULER CHARACTERISTIC

defendida por

Erik José Amézquita Morataya

Tesis para obtener el grado de

Licenciado en Matemáticas

Asesor de tesis

Dr. Antonio Rieser

Guanajuato, Gto., 28 de mayo de 2018

A mi papá, mi mamá y mi hermano.
Sin el apoyo, confianza y seguridad constante de ellos no sería
la persona que soy hoy en día.

Agradecimientos

Agradezco primero a mis papás y a mi hermano. Con su ejemplo entendí que yo soy el responsable principal de mis acciones y de todas las consecuencias que estas acciones conlleven. Ese mensaje, sencillo pero profundo, ahora forma uno de los pilares de mi vida. Este proyecto de investigación es una muestra más de este principio. También les agradezco su apoyo y confianza incondicional que me ha dado fuerza y seguridad para trazar los pasos en mi vida.

Agradezco a Antonio Rieser, mi asesor de tesis, por su apoyo constante desde aquel Verano de Investigación Científica AMC en el 2016 cuando surgió este proyecto. Con él aprendí habilidades para enfrentar de mejor manera el posgrado. También agradezco su paciencia al revisar cada una de las etapas de progreso del proyecto. Ninguno de los dos imaginó que la versión final sería tan extensa.

Agradezco a Mario Canul, quien me ayudó a manejar y entender los datos prehispánicos usados en la tesis. Él es también el responsable de mi uso diario actual de Linux.

Agradezco a Rogelio Hasimoto y a Diego Jiménez Badillo, quienes me proveyeron acceso a las mallas trianguladas de máscaras prehispánicas, los datos que sustentan la importancia y aplicabilidad de esta tesis. Asimismo agradezco a la Red Temática de Tecnologías Digitales para la Difusión del Patrimonio Cultural (RedTDPC), al Instituto Nacional de Antropología e Historia (INAH) y al Consejo Nacional de Ciencia y Tecnología (CONACYT).

Agradezco a Víctor Pérez-Abreu por todas las lecciones valiosas, tanto matemáticas como no matemáticas. Sus palabras de apoyo en noviembre 2015, cuando apenas cursaba 5to semestre, fueron vitales para aventurarme en el Análisis Topológico de Datos. Ese apoyo fue una constante con la que siempre conté a lo largo del desarrollo de esta tesis.

Agradezco a mis sinodales, Víctor Pérez-Abreu, Rogelio Hasimoto, y Miguel Naka-

mura, por el tiempo dedicado a la lectura de esta tesis y por sus comentarios valiosos para su mejora.

Agradezco a la Universidad de Arizona y a la Universidad de Tennessee por haberme dado la oportunidad de presentar parte de este proyecto en el extranjero. El taller CIMAT-UA en Tucson y las 47th Barrett Lectures en Knoxville me pusieron en contacto con públicos diversos de ideas igualmente diversas. Algunos de sus comentarios quedan aquí plasmados.

Agradezco a mis amigos de generación DEMAT 2013-2018. Con varios de ellos compartí varias experiencias maravillosas que me han formado como persona. El carrusel de emociones de la licenciatura no hubiese sido lo mismo estando solo.

Agradezco en general a la comunidad CIMAT-DEMAT, por haberme hecho sentir en casa a pesar de ser de un país distinto.

Agradezco al CIMAT como institución por haberme permitido conocer desde muy joven el significado de un ambiente de investigación constante. Agradezco su tierra fértil que permitió el desarrollo de mis proyectos personales como el CineClub CIMAT. Agradezco el haberme provisto de la Beca de Excelencia Académica, la cual fue vital para mi sustento económico durante mis estudios.

Agradezco a la Universidad de Guanajuato por tratarme de manera equitativa a pesar de no ser mexicano. Agradezco su educación y por la oportunidad de haber formado parte del Consejo Divisional y su Comisión de Docencia. Agradezco a su Cine Club por los numerosos países y épocas que me dejó explorar a través de la pantalla.

Agradezco al Sistema Nacional de Investigadores por el apoyo económico como ayudante de investigador nacional nivel III (Exp. Ayte. 13718).

Agradezco a Michigan State University y a su departamento de Computational Mathematics, Science and Engineering (CMSE) la oportunidad de continuar mis estudios de posgrado y ahondar en varios de los temas aquí tratados.

Introduction

This project aims to be an application of algebraic topology and machine learning to a current problem in archaeology. Archaeologists aim to identify and classify in an automated manner several ancient artifacts based on their shape. Since the topology of an object can convey valuable information regarding its shape, the project main direction is to understand the topology of the artifacts. In short, can Topological Data Analysis (TDA) tell us something about pre-Columbian masks?

Between 1978 and 1982, excavations were made in Templo Mayor, one of the epicenters of Aztec culture. Distributed among several crypts and chambers, 162 masks were found as offerings to Aztec authorities between the 14th and 15th centuries in today's modern Mexico City. These masks pose several questions, as their styles are unusual for the location and timeline. The archaeological community believes that the styles resemble those found across the Mexican state of Guerrero on the Pacific coast [JRM12]. Although Guerrero is roughly 300km away from Templo Mayor, the Aztecs had extensive routes of trade and tribute from coast to coast. What is more surprising is the fact that some the styles represented in the 162 mask collection resemble the styles present in Guerrero 500 or even 1000 years prior to the Templo Mayor offerings in the 14th century.

Archaeologists poise two possible hypotheses to explain such temporal discrepancy. Either the Aztecs were avid collectors of even more "ancient" artifacts, or somehow the ancient Guerrero styles were still practiced in more modern settings. Unfortunately, there is no additional evidence to support any of these theories. The problem is further complicated as there is no consensus within the archaeological community on the exact number of different styles from the Guerrero region. The problem posed by archaeologists is then to determine the number of actually different styles represented in the 162 mask collection and to identify the items accordingly. Such identification would further

the knowledge of archaeology in Guerrero and deepen the understanding of the Templo Mayor-Guerrero interactions during the 14th century. Due to the limited amount of artifacts from Guerrero, most of the proposed classifications of these 162 masks rely heavily on subjective perceptions which is reflected in several discrepancies among classifications given in the archaeological community.

The main goal of the thesis project is to define the morphology of each mask in a more objective manner based on its geometrical and topological invariants. Namely, we attempt to describe the morphology of a mask based on the Euler Characteristic. The idea is based on the Euler Characteristic Graphs (ECGs) which were proposed in the 2014 research paper by Richardson and Werman [RW14].

To obtain an ECG, we must establish beforehand a filtration function to systematically destroy part of the mesh in T steps and observe how its Euler characteristic changes with respect to such filtration. This procedure produces a T-dimensional vector for each mesh. We used then the ECGs of each mesh as descriptors to perform a classification of 128 pre-Columbian masks via supervised and unsupervised Support Vector Machines (SVMs). Our main hypothesis is that similar ECGs correspond to similar meshes. Albeit this hypothesis has not yet been mathematically proved, the data and experiments suggest its validity.

Remember that the Euler characteristic of a given simplicial complex is simply an alternating sum. This fact makes the computation of ECGs quite efficient, even if the given simplicial complex has a large number of vertices. We propose the ECG as a powerful topological descriptor which can be computed immediately without the need of super-computing infrastructure. Thus the ECG may be a topological alternative to more computationally expensive approaches such as persistent homology.

We aim to provide the archaeological with an additional tool to evaluate the morphology of an artifact in a more objective manner. Although the results presented are based on pre-Columbian masks, the same approach may be used to distinguish and classify any other kind of artifacts, such as idols or flutes. This generalization could play a crucial role to establish an automatically sorted database of 3D digitized meshes, as the archaeological community in Mexico has gathered thousands of different ancient items throughout the decades. Current efforts face difficulties with the large number of

vertices of each mesh. The computational efficiency of the proposed ECGs might help to overcome such obstacles.

The thesis is written in 5 different, self-contained chapters which can be read independently. The first chapter is an introduction to simplicial homology. The main objective is to define and prove that the Euler characteristic is a topological invariant. In order to prove such result, several definitions and results are developed throughout the chapter. Such development also aims to familiarize the reader with definitions and results using concrete examples and appropriate figures. These results and definitions are commonly used when discussing other traditional topological data analysis techniques such as persistent homology.

The second chapter follows the 2014 paper by Richardson and Werman description in [RW14] of the Euler Characteristic Graph (ECG). In their paper, Richardson and Werman develop a computationally efficient algorithm to compute the ECG of a given simplicial complex. This algorithm depends on numerical values assigned to all the simplices of highest dimension. Their algorithm runs in linear time with respect to the number of vertices in the simplicial complex. We present a variation of Richardson and Werman algorithm that depends rather on numerical values given to the vertices of a simplicial complex. This variation also runs in linear time with respect to the number of vertices.

The third chapter is an introductory tutorial to well-known supervised linear, separable Support Vector Machines (SVMs). Rather than simply stating an algorithm, we aimed to state and prove the necessary theory on constrained optimization and convex programming problems, namely the Karush-Kuhn-Tucker conditions, in order to recast the SVM as a rigorous optimization problem. From this rigorous treatment, the SVM problem is later easily generalized to non-separable, nonlinear cases.

The fourth chapter develops theory on a possible unsupervised SVM procedure for the separable, binary, linear case. This chapter follows the proposed algorithm in the 2012 paper by Karnin *et al.* in [Kar+12]. To this end, general theory on the singular value decomposition of matrices is stated and proved. We mention some possible slight improvements of our own regarding the computing time of the algorithm. At the end of the chapter, we have a discussion on some negative effects that may arise when data lives in a high dimensional space, namely overfitting and concentration of measure.

Finally the fifth chapter discusses the origin of the archaeological data to be analyzed and the considered filter functions. We propose different assortments of the masks based on different ECGs. We observed that regardless of the filter function choice, the ECGs were sensitive enough to distinguish masks with holes in their eyes from masks without holes. Such distinction was observed in both the supervised and unsupervised cases. Some of the ECGs were also able to distinguish other physical features such as large noses or thickly squinted eyes. The supervised SVMs produced sensible results overall. It is also important to highlight that the unsupervised classification idea by Karnin *et al.* in [Kar+12] does not present results based on real datasets but is limited to discuss the theoretical implications of the algorithm. We present then some of the results obtained from this unsupervised procedure when it is employed with archaeological data. An appendix is located after the bibliography where renderings of the 128 analyzed masks are found.

Unless stated otherwise, we will assume our data and vertices lie in $\mathbb{R}^d$. The i-th vector in $\mathbb{R}^d$ will be written in bold typeface as $\mathbf{x}_i$ and its j-th component will be denoted in italics as $x_{i,j}$. Inner products will be denoted as $\langle \cdot, \cdot \rangle$, and it will refer to the usual dot product in $\mathbb{R}^d$ unless stated otherwise.

Contents

Acknowledgements **iii**
Introduction **v**
Contents **ix**
1 Simplicial Homology **1**
1.1 Triangulations . . . 1
1.2 Orientation . . . 3
1.3 Cycles and boundaries . . . 5
1.4 Homology Groups . . . 11
1.5 Euler-Poincaré Formula . . . 13
2 The ECG **17**
2.1 Construction of the ECG . . . 17
2.1.1 Filtering functions . . . 18
2.1.2 Thresholding . . . 19
2.2 Computation of the ECG . . . 19
3 SVMs **25**
3.1 Linear case for SVM . . . 26
3.1.1 The separable case . . . 26
3.2 Constrained Optimization . . . 28
3.2.1 First Order Conditions . . . 28
3.2.2 Convexity . . . 36
3.2.3 Duality . . . 39
3.3 SVM as an optimization problem . . . 40
3.3.1 The separable case . . . 40
3.3.2 The non-separable case . . . 43
3.4 Nonlinear SVM . . . 46
3.4.1 Examples and conditions of kernel functions . . . 47
3.5 Multiclass SVMs . . . 48
3.5.1 One-versus-All (OvA) . . . 49
3.5.2 All-versus-All (AvA) . . . 49
3.6 Discussion . . . 50
4 USVMs **51**
4.1 The Furthest Hyperplane Problem . . . 52
4.2 Singular Value Decompositions . . . 54
4.2.1 Singular vectors . . . 55
4.2.2 SVD . . . 58
4.2.3 Best k-rank approximations . . . 60

4.3 FHP Approximation . . . 62
4.4 MMP Aproximation . . . 67
4.5 Discussion . . . 69
4.6 Curse of dimensionality . . . 70
4.6.1 Overfitting and concentration . . . 70
4.6.2 A word on concentration of measure . . . 73
5 Results **77**
5.1 Background and motivation . . . 78
5.2 Archaeological data . . . 80
5.3 Euler Characteristic Graphs . . . 81
5.4 Supervised SVM . . . 84
5.5 Unsupervised SVM . . . 85
5.6 Computation of algorithms . . . 85
5.7 Nonlinear SVM Results . . . 86
5.7.1 Planar sweeps . . . 86
5.7.2 Cylindrical Sweeps . . . 97
5.7.3 Spherical Sweeps . . . 108
5.8 Unsupervised SVM Results . . . 117
5.8.1 Set 02 vs Set 05 . . . 119
5.8.2 Set 02 vs 09 . . . 121
5.8.3 Set 03 vs 04 . . . 123
5.8.4 Set 04 vs 05 . . . 123
5.9 Conclusions and future directions . . . 124
Bibliography **131**
A Original dataset **133**
A.1 Set 02 . . . 133
A.2 Set 03 . . . 134
A.3 Set 04 . . . 134
A.4 Set 05 . . . 135
A.5 Set 06 . . . 136
A.6 Set 07 . . . 136
A.7 Set 08 . . . 136
A.8 Set 09 . . . 137
A.9 Set 10 . . . 137

Chapter 1

Simplicial Homology

Our main descriptor, the Euler Characteristic Graph (ECG), hinges on the topological information encoded in the Euler characteristic. To understand it, it is necessary to first have some knowledge of simplicial homology. We aim to introduce the reader, who may not have a strong algebraic topology background, to some basic definitions and results in simplicial homology. We work out some concrete examples and compliment them with relevant, original figures to make the exposition as friendly as possible. Simplicial homology is a critical topic to understand several techniques relevant to topological data analysis aside from the already mentioned ECG, such as persistent homology. This chapter is based primarily on Chapters 6 through 9 of [Arm83].

1.1 Triangulations

We say the *hyperplane* spanned by points $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_q \in \mathbb{R}^d$ is defined as

$$\mathbf{H} := \left\{ \sum_{i=0}^{q} \lambda \mathbf{v}_i \ : \ \lambda_i \in \mathbb{R}, \ \sum_{i=0}^{q} \lambda_i = 1 \right\}. \tag{1.1.1}$$

We will say the points are in *general position* if any subset of theirs spans a hyperplane of smaller dimension. Notice that for any $i = 1, \ldots, n$, every point on the line $\lambda \mathbf{v}_0 + (1 - \lambda)\mathbf{v}_i$ is on $\mathbf{H}$. That way we see that the points are in general position if and only if the set $\{\mathbf{v}_1 - \mathbf{v}_0, \ldots, \mathbf{v}_q - \mathbf{v}_0\}$ is linearly independent.

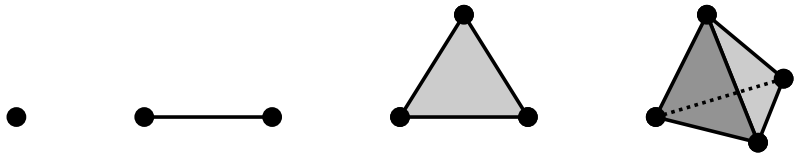
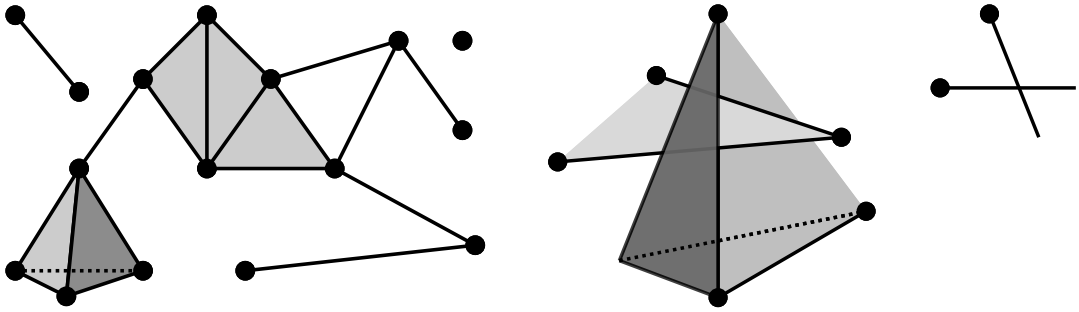


Figure 1.1: Examples of q-simplex for $q = 0, 1, 2, 3$ in $\mathbb{R}^3$



(a) Simplicial complex made of $0, 1, 2$-simplices **(b)** Conditions (i) and (ii) both fail

Figure 1.2: Example of a simplicial complex and a non-simplicial complex

Given $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_q$ points in general position, the smallest convex set containing them is referred as a *simplex of dimension* q or a q*-simplex*. The $\mathbf{v}_i$ will be called *vertices*. Refer to Figure 1.1 for some examples. In other words, the q-simplex is the convex hull of $q+1$ points in general position. That is

$$\sigma_q := \left\{ \sum_{i=0}^{q} \lambda_i \mathbf{v}_i \ : \ \lambda_i \geq 0, \ \sum_{i=0}^{q} \lambda_i = 1 \right\}. \tag{1.1.2}$$

If σ, τ are simplices, we say τ is a *face* of σ if and only if the vertices of τ are a subset of vertices of σ. This is written $\tau < \sigma$. A *simplicial complex* is a finite collection of simplices in $\mathbb{R}^d$ such that *(i)* whenever a simplex lies in the collection so it does each of its faces; and *(ii)* whenever two simplices of the collection intersect they do so in a common face. We will simply refer to it as a *complex* hereafter. The *dimension* of the complex will be the dimension of the the highest dimensional simplex it contains. Refer to Figure 1.2 for some examples. We may also understand a d-dimensional complex K as $\bigcup_{q=0}^{d} V_q$ where V_q is a finite set of q-dimensional simplices such that the previous conditions hold for the union and $V_d \neq \varnothing$.

Notice that the union of actual simplices that make up a complex K can be embed-

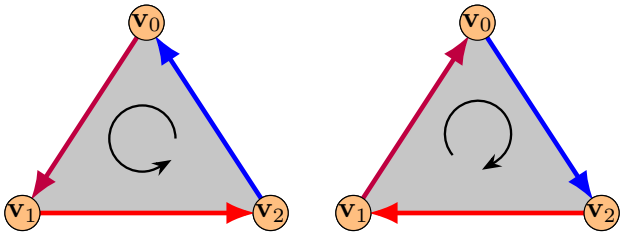


Figure 1.3: The only two possible orientations for a given 2-simplex

ded in some $\mathbb{R}^d$. Therefore it can be made into a topological space by giving it the subspace topology. When regarded this way, as a topological space, it will be denoted as $|K|$ and referred to as the *polyhedron* of the complex. A *triangulation of a topological space* X consists of a complex K and a homeomorphism $h : |K| \to X$.

1.2 Orientation

There are only two possible ways to orient a triangle: clockwise or counterclockwise, as shown by Figure 1.3. This statement can be made more precise by appropriately stating the order of the 2-simplex vertices. By $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$ we will denote a 2-simplex where we move from vertex $\mathbf{v}_0$ to $\mathbf{v}_1$, then from vertex $\mathbf{v}_1$ to $\mathbf{v}_2$ and finally from $\mathbf{v}_2$ to $\mathbf{v}_0$. We must agree that the cyclic permutations $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_0)$ and $(\mathbf{v}_2, \mathbf{v}_0, \mathbf{v}_1)$ represent the *same* orientation. At the same time, $(\mathbf{v}_2, \mathbf{v}_1, \mathbf{v}_0)$ denotes the same 2-simplex but with opposite orientation. This opposition will be denoted as $(\mathbf{v}_2, \mathbf{v}_1, \mathbf{v}_0) = -(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$.

This notion can be generalized to any q-simplex σ when $q > 0$. Consider two orderings of vertices equivalent if they differ by an even permutation. Thus there are only two possible orientations. In case we use an odd permutation, we consider the permuted simplex to have the opposite orientation and we denote it with a minus sign. In other words, for every permutation $\theta : \{0, 1, \ldots, n\} \to \{0, 1, \ldots, n\}$ we have that

$$(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n) = \operatorname{sgn}(\theta)\,(\mathbf{v}_{\theta(0)}, \mathbf{v}_{\theta(1)}, \ldots, \mathbf{v}_{\theta(n)}), \tag{1.2.1}$$

with *sgn* the sign function where $\operatorname{sgn}(\theta)$ is 1 in case θ is an even permutation and -1 otherwise. In case $n = 0$, that is, if we are dealing with a single vertex, there is only one possible orientation. Whenever we fix the orientation of σ, we will say that σ is an *oriented simplex*.

Now consider the face σ_i defined by the same vertices of σ omitting $\mathbf{v}_i$. If i is even, the orientation of σ_i specified by this ordering is called the *orientation induced* by σ. If i is odd, the orientation induced by σ will be the opposite to the one defined by the ordering of vertices in σ. In other words, if σ_i is an oriented simplex with the induced orientation then

$$\sigma_i = (-1)^i(\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_n). \tag{1.2.2}$$

It is important to remark that the induced orientation on σ_i depends solely on the orientation of σ and not on the index i of the removed vertex. To see this, consider a permutation $\theta : \{0, 1, \ldots, n\} \to \{0, 1, \ldots, n\}$ and a new n-simplex $\sigma^\theta := (\mathbf{v}_{\theta(0)}, \mathbf{v}_{\theta(1)}, \ldots, \mathbf{v}_{\theta(n)})$ made by the same vertices of σ. Suppose $\theta^{-1}(i) = k$ and consider the face σ^θ_k defined by the same vertices of σ^θ except for $\mathbf{v}_{\theta(k)} = \mathbf{v}_i$. We have to show that

$$\sigma^\theta_k = \operatorname{sgn}(\theta)\, \sigma_i. \tag{1.2.3}$$

If $\theta^{-1}(i) = i$, then the previous equation is just restating (1.2.1) and there is nothing else to show. If $\theta^{-1}(i) = k \neq i$, then θ can be seen as composition of disjoint cycles. Assume i happens to be in the cycle $\sigma = (i\ j_1\ \cdots\ j_l\ k)$. Without loss of generality we can also assume that $i < k$. Next rewrite $\theta = \sigma \circ \psi$ where ψ is a disjoint permutation to σ. Now consider the permutation cycle $\sigma' := (k\ j_l\ \cdots\ j_1)$ of $\{0, 1, \ldots, n\} \setminus \{i\}$ and define $\theta' := \sigma' \circ \psi^{-1}$. As σ' and ψ^{-1} are disjoint we have that $\operatorname{sgn}(\theta') = \operatorname{sgn}(\sigma')\operatorname{sgn}(\psi^{-1}) = -\operatorname{sgn}(\sigma)\operatorname{sgn}(\psi) = -\operatorname{sgn}(\theta)$. The orientation of σ^θ_k induced from σ^θ is then

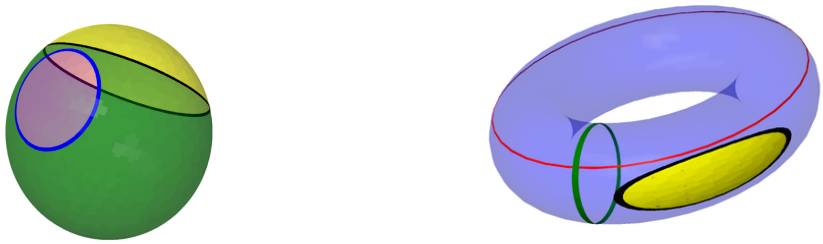
$$\begin{aligned}
\sigma^\theta_k &= (-1)^k(\mathbf{v}_{\theta(0)}, \ldots, \mathbf{v}_{\theta(i-1)}, \mathbf{v}_{\theta(i)}, \mathbf{v}_{\theta(i+1)}, \ldots, \mathbf{v}_{\theta(k-1)}, \mathbf{v}_{\theta(k+1)}, \ldots, \mathbf{v}_{\theta(n)}) \\
&= \operatorname{sgn}(\theta')(-1)^k(\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_k, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_{k-1}, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_n) \\
&= (-1)^{k-i-1}(-1)\operatorname{sgn}(\theta)(-1)^k(\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_{k-1}, \mathbf{v}_k, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_n) \\
&= \operatorname{sgn}(\theta)\, \sigma_i.
\end{aligned}$$

The third equality is due to the fact that we need $k-i-1$ transpositions (each of them an odd permutation) to move $\mathbf{v}_k$ from the i-th place to the $(k-1)$-th place. Thus we proved (1.2.3). This result can be stated as a lemma.

Lemma 1.2.1. *The induced orientation on σ_i by σ depends only on the orientation of σ and not on the particular ordering of the vertices of σ chosen to represent this orientation.*

(a) Compatibly oriented triangles **(b)** Incompatibly oriented triangles

Figure 1.4: Compatible and incompatible orientations of triangles



(a) Blue and black curves split the surface **(b)** Red and green curves do not split the surface

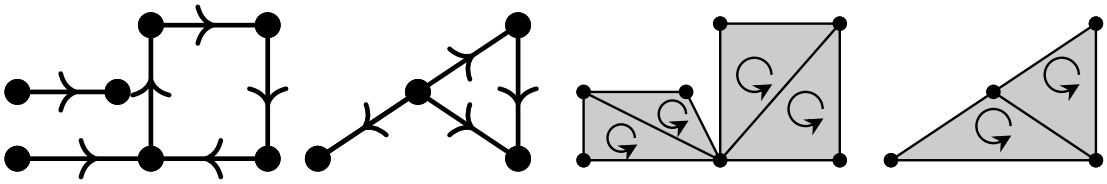
Figure 1.5: Closed curves on a sphere, a torus and a punctured torus

Finally, we say that a complex K is orientable if it is possible to orient all the simplices of K in a compatible way. That is to say, two adjacent simplices induce *opposite* orientations on their common face. Refer to Figure 1.4.

1.3 Cycles and boundaries

On a sphere, any closed curve divides its surface in two different connected components. This is not be the case for the torus. As we see in Figure 1.5b, although the black curve splits the torus' surface into yellow and blue regions, the green and red curves don't split the surface into two different connected regions.

We would like to ignore those curves that enclose a piece of contractible surface in order to recognize the holes of the torus. We will work then in a general setting with a topological space X and some fixed triangulation (K, h) of it. Consider $C_1(K)$ the free Abelian group generated by all the 1-dimensional simplices of K. This group will be the 1*st simplicial chain group* and its elements will be referred as 1-*dimensional chains*. That



(a) 1-chain **(b)** 2-chain

Figure 1.6: Example of oriented simplicial chains

is, a 1-chain is of the form $c := \sum_{i=0}^{n} \lambda_i(\mathbf{u}_i, \mathbf{v}_i)$ for $\lambda_i \in \mathbb{Z}$ where $\mathbf{u}_i, \mathbf{v}_i$ are vertices in K. We will also consider $(\mathbf{u}, \mathbf{v}) + (\mathbf{v}, \mathbf{u}) = 0$ and $\lambda[-(\mathbf{u}, \mathbf{v})] = (-\lambda)(\mathbf{u}, \mathbf{v})$ for every pair of vertices $\mathbf{u}, \mathbf{v} \in K$. Refer to Figure 1.6a This notions can be generalized to higher dimensions.

Let $C_q(K)$ be the free Abelian group generated by all the q-simplexes of K. The group will be referred as the q-th *simplicial chain group* and its elements will be referred as q-*dimensional chains*. In both groups it happens that $\sigma + \theta = 0$ whenever σ and θ represent the same simplex with opposite orientation. We can then define homomorphisms φ on these chain groups by defining φ for every q-simplex and later extend it linearly to the rest of the q-chain. We only have to care that the relationship $\varphi(\sigma) + \varphi(-\sigma) = 0$ holds.

One such homomorphism is the *boundary homomorphism* $\partial_q : C_q(K) \to C_{q-1}(K)$. The boundary of a q-simplex σ is defined as $(q-1)-$chain determined by the sum of its $(q-1)$-dimensional oriented faces, each taken with the orientation induced by σ. That is

$$\partial_q(\sigma) = \partial_q(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_q) = \sum_{i=0}^{q} (-1)^i (\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_q). \tag{1.3.1}$$

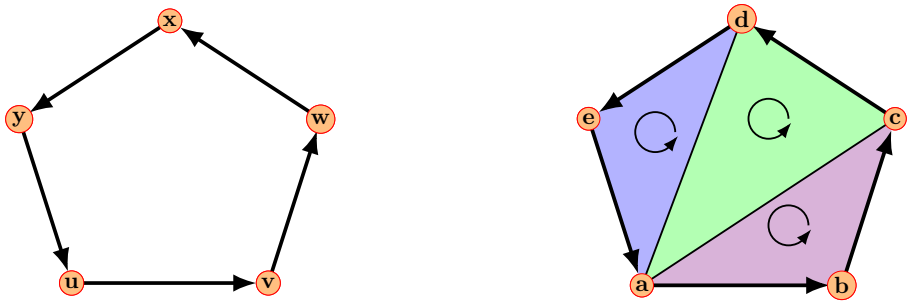
We also check that $\partial(\sigma) + \partial(-\sigma) = 0$, as changing the orientation of σ changes the induced orientation of each of its $(q-1)$-dimensional faces. Keep in mind that the actual indexing of vertices in σ does not affect the boundary due to Lemma 1.2.1.

From (1.3.1), the boundary of an oriented edge is

$$\partial(\mathbf{u}, \mathbf{v}) = \mathbf{v} - \mathbf{u}. \tag{1.3.2}$$

And for an oriented triangle its boundary is

$$\partial(\mathbf{u}, \mathbf{v}, \mathbf{w}) = (\mathbf{u}, \mathbf{v}) + (\mathbf{v}, \mathbf{w}) + (\mathbf{w}, \mathbf{u}). \tag{1.3.3}$$



(a) $c_1 \in Z_1(K)$ **(b)** $c_2 \in B_1(K)$

Figure 1.7: Boundaries and cycles

Think of the oriented 1-chain $c_1 = (\mathbf{u},\mathbf{v}) + (\mathbf{v},\mathbf{w}) + (\mathbf{w},\mathbf{x}) + (\mathbf{x},\mathbf{y}) + (\mathbf{y},\mathbf{u})$ as shown in Figure 1.7a. We define its boundary linearly, that is the sum of boundaries of edges is the boundary of the whole edge sum. That is,

$$\begin{aligned}\partial c_1 &= \partial(\mathbf{u},\mathbf{v}) + \partial(\mathbf{v},\mathbf{w}) + \partial(\mathbf{w},\mathbf{x}) + \partial(\mathbf{x},\mathbf{y}) + \partial(\mathbf{y},\mathbf{u}) \\ &= (\mathbf{v}-\mathbf{u}) + (\mathbf{v}-\mathbf{w}) + (\mathbf{x}-\mathbf{w}) + (\mathbf{y}-\mathbf{x}) + (\mathbf{u}-\mathbf{y}) = 0.\end{aligned}$$

Consequently we say that c_1 *has no boundary*. Now think of the oriented 1-chain $c_2 = (\mathbf{a},\mathbf{b}) + (\mathbf{b},\mathbf{c}) + (\mathbf{c},\mathbf{d}) + (\mathbf{d},\mathbf{e}) + (\mathbf{e},\mathbf{a})$ and the triangles $(\mathbf{a},\mathbf{b},\mathbf{c})$, $(\mathbf{a},\mathbf{c},\mathbf{d})$ and $(\mathbf{a},\mathbf{d},\mathbf{e})$ of K which are all oriented compatibly as in Figure 1.7b. Consider the boundary of the sum of these three triangles.

$$\begin{aligned}\partial((\mathbf{a},\mathbf{b},\mathbf{c}) + (\mathbf{a},\mathbf{c},\mathbf{d}) + (\mathbf{a},\mathbf{d},\mathbf{e})) &= \partial(\mathbf{a},\mathbf{b},\mathbf{c}) + \partial(\mathbf{a},\mathbf{c},\mathbf{d}) + \partial(\mathbf{a},\mathbf{d},\mathbf{e}) \\ &= ((\mathbf{a},\mathbf{b}) + (\mathbf{b},\mathbf{c}) + (\mathbf{c},\mathbf{a})) + ((\mathbf{a},\mathbf{c}) + (\mathbf{c},\mathbf{d}) + (\mathbf{d},\mathbf{a})) \\ &\quad + ((\mathbf{a},\mathbf{d}) + (\mathbf{d},\mathbf{e}) + (\mathbf{e},\mathbf{a})) \\ &= ((\mathbf{a},\mathbf{b}) + (\mathbf{b},\mathbf{c}) - \cancel{(\mathbf{a},\mathbf{c})}) + (\cancel{(\mathbf{a},\mathbf{c})} + (\mathbf{c},\mathbf{d}) - \cancel{(\mathbf{a},\mathbf{d})}) \\ &\quad + (\cancel{(\mathbf{a},\mathbf{d})} + (\mathbf{d},\mathbf{e}) + (\mathbf{e},\mathbf{a})) = c_2.\end{aligned}$$

Thus we effectively say that c_2 *bounds* the three triangles in K above.

Now consider the subgroup of $C_1(K)$ consisting of all 1-chains $z = \sum_{i=1}^{n} \lambda_i(\mathbf{u}_i,\mathbf{v}_i)$ such that they have no boundary, that is, $\partial(z) = 0$. Such expression will be called a 1-*dimensional cycle* of K. Since the coefficients are integers and ∂ is lineal, the 1-dimensional cycles form an Abelian subgroup. This subgroup will be denoted as $Z_1(K)$.

A 1-chain $z = \sum_{i=1}^{n} \lambda_i(\mathbf{u}_i, \mathbf{v}_i)$ will be considered *simple* if each of its vertices appears exactly twice in the sum with coefficient 1, as c_2 in Figure 1.7b. Notice that an oriented, simple closed polygonal curve in K can be thought of as the sum of oriented edges and it is hence a 1-cycle, which will be considered an *elementary* 1-cycle.

Lemma 1.3.1. *$Z_1(K)$ is generated by these elementary 1-cycles.*

Proof. Say $z \in \sum_{i=1}^{k} \lambda_i(\mathbf{u}_i, \mathbf{v}_i) \in Z_i(K)$. Say $V(K) = \{\mathbf{u}_1, \mathbf{v}_1, \ldots, \mathbf{u}_k, \mathbf{v}_k\}$ is the set of all different vertices in z. Assume that every λ_i is positive and that if $(\mathbf{u}_i, \mathbf{v}_i) = (\mathbf{u}_j, \mathbf{v}_j)$ then $i = j$. In other words, z is non trivial and the summands of the form $(\mathbf{u}, \mathbf{v}) - (\mathbf{u}, \mathbf{v})$ vanish. For every oriented edge $(\mathbf{u}, \mathbf{v})$, we will refer to $\mathbf{u}$ as the posterior vertex and $\mathbf{v}$ as the anterior vertex. Additionally, we know that

$$\partial z = \sum_{i=0}^{k} \lambda_i \partial(\mathbf{u}_i, \mathbf{v}_i) = \sum_{i=0}^{k} \lambda_i[\mathbf{v}_i - \mathbf{u}_i] = \sum_{i=0}^{k} \mathbf{v}_i - \sum_{i=0}^{k} \mathbf{u}_i = 0. \tag{1.3.4}$$

Observe from the previous equality states every vertex $\mathbf{v}$ is the anterior of some edge and the posterior of some other edge. As $\partial z = 0$, k must be larger than 1. If $k = 2$, then it must be $z = \lambda((\mathbf{u}, \mathbf{v}) + (\mathbf{v}, \mathbf{u}))$, which is a null chain and hence elementary. Now assume by hypothesis of induction that if z has less than k different oriented edges, then z can be rewritten as sum of elementary 1-cycles. To understand the case of k different oriented edges is better to use language of multi-digraphs, that is, graphs where two vertices can be connected by more than one directed edge. Consider a multi-digraph $G(V, E)$. Each vertex $\mathbf{v}$ in z corresponds to a vertex in $\tilde{\mathbf{v}}$ in V (the tilde will distinguish between simplicial vertices and digraph vertices). The set of edges is constructed as follows: for a fixed vertex $\mathbf{v}$, the edge $(\tilde{\mathbf{v}}, \tilde{\mathbf{w}}) \in E$ if and only if $(\mathbf{v}, \mathbf{w}) = (\mathbf{u}_i, \mathbf{v}_i)$ is part of the z chain for some $i \in \{1, \ldots, k\}$. The number of edges connecting $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{w}}$ will be exactly λ_i. Refer to Figure 1.8 for an example. As observed after (1.3.4), the number of incoming edges equals the number of departing edges from every vertex $\tilde{\mathbf{v}}$. Hence the vertex $\mathbf{v}$ must be the anterior of some oriented 1-simplex. Thus there is an edge $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ in E.

The idea is that we can now read the cycle z as a graph $G(V, E)$ where we are able to split graph cycles into simple cycles, that is, cycles that don't traverse any vertex twice. Consider all the possible simple cycles $\tilde{A}_1, \ldots, \tilde{A}_n$ that start and end at $\tilde{\mathbf{v}}$. Say $\tilde{A}_i :=$

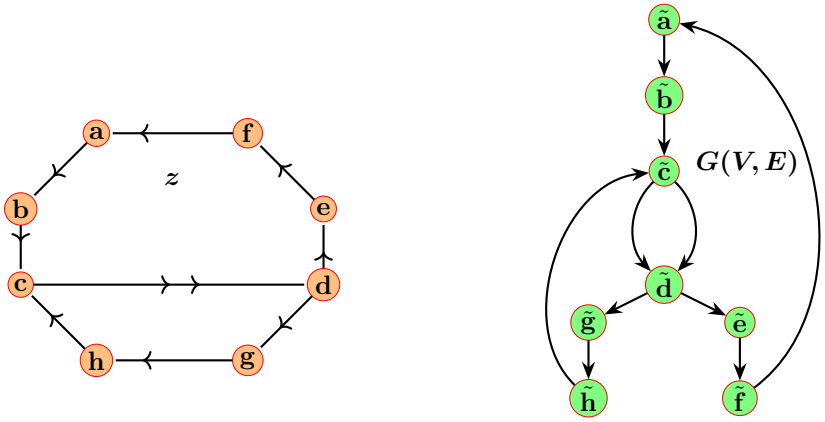


Figure 1.8: Example of 1-cycle z and its corresponding multi-digraph $G(V, E)$.

$(\tilde{\mathbf{v}}, \tilde{\mathbf{w}}^i_1, \ldots, \tilde{\mathbf{w}}^i_{m_i}, \tilde{\mathbf{u}}, \tilde{\mathbf{v}})$. Then the 1-chain $y_i := (\mathbf{v}, \mathbf{w}^i_1) + \ldots + (\mathbf{w}^i_{m_i}, \mathbf{u})$ is a simple 1-cycle. If all the y_i's are removed, that is, we now consider the 1-cycle $z' := \sigma - \sum_{i=1}^n \tau_i$, then we have effectively removed all the elementary 1-cycles within z that contain the vertex $\mathbf{v}$. Hence z' has less than k different edges and it can be decomposed by induction hypothesis as sum of elementary 1-cycles. As $\sum_{i=1}^n \tau_i$ is in itself a sum of elementary 1-cycles, we conclude that z is a sum of elementary 1-cycles.

□

We say that a 1-cycle is a 1-*dimensional boundary cycle* if there is an oriented 2-chain (a linear combination of oriented triangles) whose boundary is the given cycle. The subset of bounding cycles is referred as $B_1(K)$. For example, c_2 is a 1-boundary in Figure 1.7b.

Observe that for any 2-chain c we have that $\partial^2 c = 0$. Since the boundary homomorphism is linear, it is enough to see that $\partial^2(\mathbf{u}, \mathbf{v}, \mathbf{w}) = 0$ for any 2-simplex $(\mathbf{u}, \mathbf{v}, \mathbf{w})$. This is a simple computation.

$$\begin{aligned}\partial^2(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \partial(\mathbf{u}, \mathbf{v}) + \partial(\mathbf{v}, \mathbf{w}) + \partial(\mathbf{w}, \mathbf{u}) \\ &= (\mathbf{v} - \mathbf{u}) + (\mathbf{w} - \mathbf{v}) + (\mathbf{u} - \mathbf{w}) = 0.\end{aligned}$$

From the last equality we see that $B_1(K) \subset Z_1(K)$. Moreover, $B_1(K)$ is a normal subgroup of $Z_1(K)$.

Lemma 1.3.2. *$B_1(K)$ is a normal subgroup of $Z_1(K)$.*

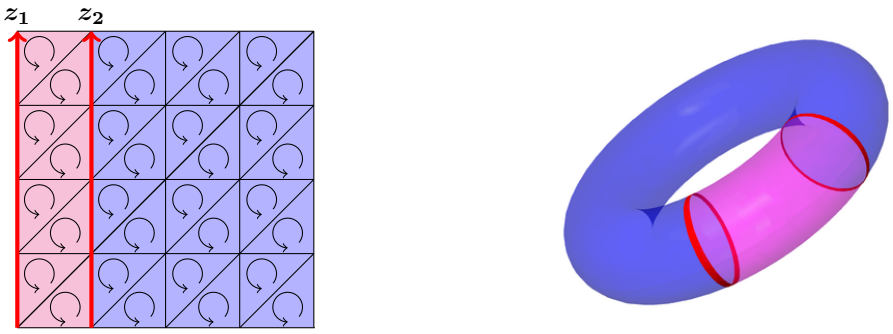


Figure 1.9: Flat torus on the left and joining the correspondences on the right.

Proof. Take any two 1-cycles $z_1, z_2 \in B_1(K)$. Then there exist 2-dimensional chains $y_i = \sum_{k=1}^{n} \lambda_k^i(\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i)$ such that $\partial y_i = z_i$ for $i = 1, 2$. Consider then the 2-dimensional chain $y := y_1 + y_2$. Due to linearity $\partial y = z_1 - z_2$. That is, $z_1 - z_2 \in B_1(K)$ and thus $B_1(K)$ is a subgroup. Finally, remember $Z_1(K)$ is Abelian, so all its subgroups are normal. □

Based on the previous lemma we may define the quotient

$$H_1(K) := Z_1(K)/B_1(K), \tag{1.3.5}$$

which we call the *first homology group* of K. Thus two cycles whose difference is a bounding cycle will be in the same coset of $H_1(K)$, and they will be referred as *homologous* cycles.

For example, consider the flat torus. That is, consider the unit interval $I := [0, 1]$ and the unit square I^2 along the equivalence relations

$$\begin{aligned} (x, 0) &\sim (x, 1), \quad \forall\, x \in I \\ (0, y) &\sim (1, y), \quad \forall\, y \in I \\ (x, y) &\sim (x, y), \quad 0 < x, y < 1. \end{aligned}$$

Then the *flat torus* is defined as the quotient $T^2 := I^2/\!\sim$. Consider now the 1-cycle z_1 defined by the curve $\gamma_1 : I \to T^2$ given by $t \mapsto [(0, t)]$. Similarly z_2 is given by $\gamma_2 : t \mapsto [(\frac{1}{4}, t)]$. Refer to Figure 1.9. Observe that $z_2 - z_1$ is the boundary of the magenta colored oriented triangles in the figure. That is, these 1-cycles enclose a region of the torus surface. Thus z_1 and z_2 are homologous 1-cycles.

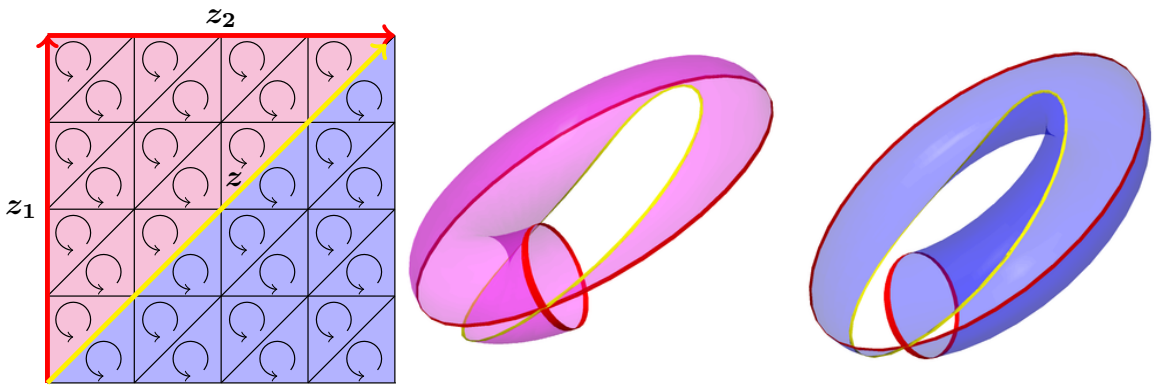


Figure 1.10: Flat torus on the left and joining the correspondences on the right.

Moreover, it may be shown that any 1-cycle on the torus is homologous to a linear combination of a cycle z_1 given by $\gamma_1 : t \mapsto [(a,t)]$ and a cycle z_2 given by $\gamma_2 : t \mapsto [(t,b)]$ for some fixed $a, b \in I$. Thus $H_1(T) = \mathbb{Z} \oplus \mathbb{Z}$ and $\beta_1 = 2$. For example, in Figure 1.10, the diagonal yellow 1-cycle z is homologous to the 1-cycle $z_1 + z_2$. Observe that $z_1 + z_2 - z$ are the boundary of the collection of oriented magenta triangles.

1.4 Homology Groups

The previous section can be generalized to higher dimensions. Say $K = \bigcup_{q=0}^{d} V_q(K)$ is a finite d-dimensional simplicial complex where each $V_i(K)$ is a finite set of i-dimensional simplices. Remember from Section 1.2 that each of simplex from $V_q(K)$ can be oriented in exactly one of two possible ways. Let $C_q(K)$ be as in the previous section. Notice that the rank of $C_q(K)$, that is, the number of generators, is exactly $|V_q(K)|$.

Remember that we shall define homomorphisms φ on these groups by defining such homomorphism for each generator and require that $\varphi(\sigma) + \varphi(-\sigma) = 0$ holds. Next we extend linearly for the rest of the q-chain. Remember the *boundary homomorphism* $\partial_q : C_q(K) \to C_{q-1}(K)$ defined as

$$\partial_q(\sigma) = \partial_q(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_q) = \sum_{i=0}^{q} (-1)^i (\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_q).$$

We also say the boundary of a vertex is 0 and the set $C_{-1}(K) := \{0\}$. That way $\partial_0 \equiv 0$. Finally we extended ∂_q linearly to obtain a homomorphism. We define the subgroup of

q-dimensional cycles as

$$Z_q(K) := \ker \partial_q, \tag{1.4.1}$$

which is a finitely generated Abelian group.

Lemma 1.4.1. *The composition $C_{q+1}(K) \xrightarrow{\partial_{q+1}} C_q(K) \xrightarrow{\partial_q} C_{q-1}(K)$ is the zero homomorphism.*

Proof. By construction of ∂, we need only to show that $\partial_{q+1} \circ \partial_q(\sigma) = 0$ for any oriented q-simplex σ. From the definition

$$\begin{aligned}
\partial_q(\partial_{q+1}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{q+1})) &= \partial_q \left(\sum_{i=0}^{q+1} (-1)^i (\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_{q+1}) \right) \\
&= \sum_{i=0}^{q+1} (-1)^i \sum_{j=0}^{i-1} (-1)^j \quad (\mathbf{v}_0, \ldots, \mathbf{v}_{j-1}, \mathbf{v}_{j+1}, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_{q+1}) \\
&+ \sum_{i=0}^{q+1} (-1)^i \sum_{j=i+1}^{q+1} (-1)^{j-1} (\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_{j-1}, \mathbf{v}_{j+1}, \ldots, \mathbf{v}_{q+1}).
\end{aligned}$$

The second sum has a $(-1)^{j-1}$ coefficient as the actual position of the vertices $\mathbf{v}_{i+1}, \ldots, \mathbf{v}_{q+1}$ was reduced by 1. In other words, the vertex $\mathbf{v}_{i+k}$ occupies the $(i+k-1)$-th position. Now, each oriented simplex $(\mathbf{v}_0, \ldots, \widehat{\mathbf{v}_j}, \ldots, \widehat{\mathbf{v}_i}, \ldots, \mathbf{v}_{q+1})$ appears twice, once in each sum. That is, it appears once with a $(-1)^i(-1)^j$ coefficient and later with a $(-1)^i(-1)^{j-1}$ coefficient. Hence, the whole sum vanishes. □

We define then the set of *q-dimensional boundaries* as

$$B_q(K) := \text{im}\,(\partial_{q+1}) \subset Z_q(K). \tag{1.4.2}$$

Analogous to Lemma 1.3.2, $B_q(K)$ is a normal finitely generated Abelian subgroup of $Z_q(K)$ and we define *the* q-th *homology group* of K as the quotient

$$H_q(K) := Z_q(K)/B_q(K). \tag{1.4.3}$$

Thus, each q-cycle determines an element $[z] \in H_q(K)$ called the *homology class of* z. Two q-cyles whose difference is a q-boundary have the same homology class and will be said to be *homologous* cycles.

By construction, $H_q(K)$ is a finitely generated Abelian group. Due to the Fundamental Theorem of Finitely Generated Abelian Groups (cf. [Rot99, Ch. 9]), we can decompose it as $H_q(K) \cong F \oplus T$ where F is a finitely generated free Abelian group (specifically it is isomorphic to a direct sum of a finite number of copies of $\mathbb{Z}$) and T is a finite Abelian group. The rank of F (that is, the number of copies of $\mathbb{Z}$) is called the q-th *Betti number* of K. It is denoted as β_q.

1.5 The Euler-Poincaré Formula

With K as in the last section, we define its *Euler characteristic* as

$$\chi(K) := \sum_{q=0}^{d} (-1)^q |V_q(K)|. \tag{1.5.1}$$

Before proceeding, it will be convenient to reinterpret the Betti numbers as follows.

Suppose we now consider *rational numbers* as coefficients when we form all possible linear combinations of oriented simplices

$$r_1\sigma_1 + \ldots + r_s\sigma_s, \quad r_i \in \mathbb{Q}, \;\; \sigma_i \in V_q(K).$$

The set of all possible combinations is now a vector space V over $\mathbb{Q}$. Let W be the subspace spanned by the elements of the form $\sigma + \tau$ where σ, τ are the same simplex in $V_q(K)$ but with opposite orientations. The quotient $C_q(K, \mathbb{Q}) := V/W$ will be referred as the vector space of *rational q-chains*. In other words, we take $\sigma + \tau = 0$ when they represent the same simplex with different orientations. We can observe that $\dim C_q(K, \mathbb{Q}) = |V_q(K)|$.

The boundary homomorphism is a linear map of vector spaces over $\mathbb{Q}$ which is defined as in the integer case in (1.3.1). We define the rational q-cycles in $Z_q(K, \mathbb{Q})$ and the bounding rational q-cycles in $B_q(K, \mathbb{Q})$ analogous to the definitions in (1.4.1) and (1.4.2). Notice that both $Z_q(K, \mathbb{Q})$ and $B_q(K, \mathbb{Q})$ are subspaces of $C_q(K, \mathbb{Q})$. The quotient $H_q(K, \mathbb{Q}) := Z_q(K, \mathbb{Q})/B_q(K, \mathbb{Q})$ will be referred as the q-th *homology group of K with rational coefficients*.

The following lemma will help us to translate the Betti numbers from its original language of free Abelian groups to the new construction with linear spaces.

Lemma 1.5.1. *β_q is the dimension of $H_q(K,\mathbb{Q})$ when regarded as a vector space over $\mathbb{Q}$.*

This result actually follows from a stronger proposition stated in [Hat02, Corollary 3A.6.(a)] as follows.

Corollary 1.5.2. *$H_q(K;\mathbb{Q}) \cong H_q(K)\otimes\mathbb{Q}$, so when $H_q(K)$ is finitely generated, the dimension of $H_q(K;\mathbb{Q})$ as a vector space over $\mathbb{Q}$ equals the rank of $H_q(K;\mathbb{Z})$.*

Its proof via a even more general framework relies heavily on elements of commutative algebra which we will omit in this work. Corollary 1.5.2 states that the vector space $H_q(K;\mathbb{Q})$ can actually be seen as $H_q(K)$ up to some rational coefficient. We can present a more intuitive albeit not completely rigorous proof of the lemma above as follows.

Intuitive proof of Lemma 1.5.1. Suppose $H_q(K) \cong F \oplus T$, where $F \cong \bigoplus_{i=1}^{\beta_q}\mathbb{Z}$ and T is a finite Abelian group. Choose a minimal set of generators $[z_1],\dots,[z_{\beta_q}],[w_1],\dots,[w_{\gamma_q}]$ for $H_q(K)$ where the $[z_i]$'s generate the free part F and the $[w_i]$'s all have finite order. For any rational q-cycle we can rewrite it as

$$\frac{a_1}{b_1}\sigma_1+\cdots+\frac{a_s}{b_s}\sigma_s = \frac{1}{b_1\cdots b_s}(\text{cycle with integer coefficients}) \tag{1.5.2}$$

$$= \frac{1}{b_1\cdots b_s}(\text{linear integer combination of } z_i\text{'s and } w_j\text{'s}). \tag{1.5.3}$$

Notice that every q-cycle class $[z]$ with integer coefficients can be regarded as a q-cycle class $\{z\}$ with rational coefficients. Hence $\{z_1\},\dots,\{z_{\beta_q}\},\{w_1\},\dots,\{w_{\gamma_q}\}$ span the whole $H_q(K,\mathbb{Q})$.

If $[w]\in T$, then the q-cycle w must have a finite order, say m. Then there must exist a $(q+1)$-chain w' with integer coefficients such that $\partial(w')=mw$. Thus $\frac{1}{m}w$ is a rational q-cycle and $\partial(\frac{1}{m}w')=\frac{1}{m}\partial(w')=\frac{1}{m}\cdot mw=w$. In other words, w bounds the rational $(q+1)$-chain $\frac{1}{m}w'$ and consequently $w\in B(K,\mathbb{Q})$. Thus $H_q(K,\mathbb{Q})$ must be spanned by $\{z_1\},\dots,\{z_{\beta_q}\}$.

Now suppose some rational linear combination of some generating representative cycles z_i's produce a bounding rational cycle $w\in B_q(K,\mathbb{Q})$. Thus it must exist a rational $(q+1)$-chain $y\in C_{q+1}(K,\mathbb{Q})$ such that $\partial y=w$. Just as in (1.5.3), we can compute both w

and y as integer chains times a rational coefficient.

$$\begin{aligned} w &= \frac{a_1}{b_1} z_1 + \cdots + \frac{a_{\beta_q}}{b_{\beta_q}} z_{\beta_q} \\ &= \frac{1}{b_1 \cdots b_{\beta_q}} \sum_{i=1}^{\beta_q} \lambda_i z_i = \frac{1}{b} w', \qquad\qquad b := b_1 \cdots b_{\beta_q},\ \lambda_i \in \mathbb{Z}. \end{aligned}$$

Observe that $w' \in Z_q(K)$. Analogously, we can compute

$$\begin{aligned} y &= \sum_{i=1}^{l} \frac{\hat{a}_i}{\hat{b}_i} \tau_i, \qquad\qquad \tau_i \in V_{q+1} \\ &= \frac{1}{\hat{b}_1 \cdots \hat{b}_l} \sum_{i=1}^{l} \hat{\lambda}_i \tau_i = \frac{1}{\hat{b}} y' \qquad\qquad \hat{b} = \hat{b}_1 \cdots \hat{b}_l,\ \hat{\lambda}_i \in \mathbb{Z}, \end{aligned}$$

where $y' \in C_q(K)$. Since $\partial y = w$, it must be that

$$\frac{b}{\hat{b}} \partial y' = w' \in Z_q(K). \tag{1.5.4}$$

Since both y' and w' have integer coefficients, by comparing both sides of the equation 1.5.4, it must be that $b/\hat{b} \in \mathbb{Z}$. Thus we conclude that $\frac{b}{\hat{b}} y' \in C_{q+1}(K)$. Then w' must be an integer bounding cycle. As the $[z_i]$'s are generators of a free Abelian group, it must be the case that each coefficient in w' is zero. Thus $\frac{a_1}{b_1} = \ldots = \frac{a_{\beta_q}}{b_{\beta_q}} = 0$ and we conclude that $\{\{z_1\}, \ldots, \{z_{\beta_q}\}\}$ is linearly independent. □

From the last lemma we can rewrite β_q as $\dim(Z_q(K, \mathbb{Q})) - \dim(B_q(K, \mathbb{Q}))$. Finally we are ready to prove the topological invariance of the Euler characteristic via a stronger result.

Theorem 1.5.3. *The Euler characteristic of a finite d-dimensional complex K is given by*

$$\chi(K) = \sum_{q=0}^{d} (-1)^i \beta_i. \tag{1.5.5}$$

Proof. Throughout this proof we will abbreviate $C_q(K, \mathbb{Q}), Z_q(K, \mathbb{Q}), B_q(K, \mathbb{Q})$ as C_q, Z_q, B_q respectively. We will then choose bases for C_q as follows.

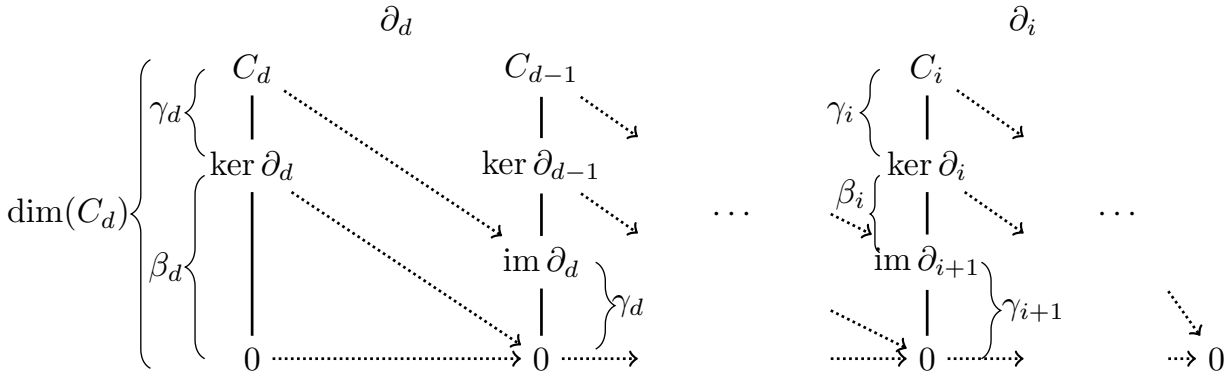


Figure 1.11: Due to the Dimension Theorem $\dim(C_i) = \dim(Z_i) + \dim(B_{i+1})$.

Since K has no $d+1$-simplices, $B_d = \{0\}$ and $\beta_d = \dim(Z_d)$ from the previous lemma. Choose a basis $\{z_1^d, \ldots, z_{\beta_d}^d\}$ for Z_n. Then consider additional d-cycles $c_1^d, \ldots, c_{\gamma_d}^d$ to form a basis for the whole C_n. As $\partial Z_n = \{0\}$, we know from the Dimension Theorem that $\{\partial c_1^n, \ldots, \partial c_{\gamma_d}^d\}$ is a basis for B_{d-1}. Refer to diagram 1.11. Extend that basis by $(d-1)$-cycles $z_1^{d-1}, \ldots, z_{\beta_{d-1}}^{d-1}$ to form a basis for Z_{d-1}. Next extend it further by elements $c_1^{d-1}, \ldots, c_{\gamma_{d-1}}^{d-1}$ so it is a basis for C_{d-1}. Then repeat the same steps for C_{d-2} and proceed algorithmically for every dimension below. The process will terminate with a basis $\{\partial c_1^1, \ldots, \partial c_{\gamma_1}^1\}$ for $Z_0 = C_0$. From the Dimension Theorem we observe that $\dim(C_q) = \gamma_{q+1} + \beta_q + \gamma_q$. But we also know that $\dim(C_q) = |V_q(K)|$. Thus

$$\begin{aligned}\chi(K) = \sum_{q=0}^{d}(-1)^q |V_q(K)| &= \sum_{q=0}^{d}(-1)^q(\gamma_{q+1} + \beta_q + \gamma_q) \\ &= \sum_{q=0}^{d}(-1)^q \beta_q.\end{aligned}$$

□

As the β_q's depend solely on the homology group $H_q(K)$, which in turn depends solely on the topological structure of K, we conclude that $\chi(K)$ is a topological invariant.

Chapter 2

The Euler Characteristic Graph

The main descriptor for the objects to analyze will be the *Euler-Characteristic Graph (ECG)*, first described by Richardson and Werman in their 2014 research paper [RW14]. The reason for this choice lies in its simplicity to compute despite the objects' large number of vertices. In this chapter we will first describe the ECG and follow the same idea of Richardson and Werman to compute the ECG in $O(V + T)$ time, where V is number of vertices of the object and T is the number of thresholds, an integer parameter set by the user. It must be highlighted that Richardson and Werman compute the ECG of a simplicial complex K based on numerical values assigned to the simplices of highest dimension in the comples. The Algorithm 1 that we state follows the same idea but it is based on numerical values assigned to the vertices of K.

2.1 The Euler Characteristic Graph construction

Consider a d-dimensional simplicial complex $K = \bigcup_{q=0}^{d} V_q(K)$, with V_q the finite set of all its q-dimensional simplexes. Remember from 1.5.1 that its Euler Characteristic (EC) is defined as

$$\chi(K) := \sum_{q=0}^{d} (-1)^q |V_q(K)|.$$

The ECG is based on this, using additional objects: a filter function and a positive number of thresholds.

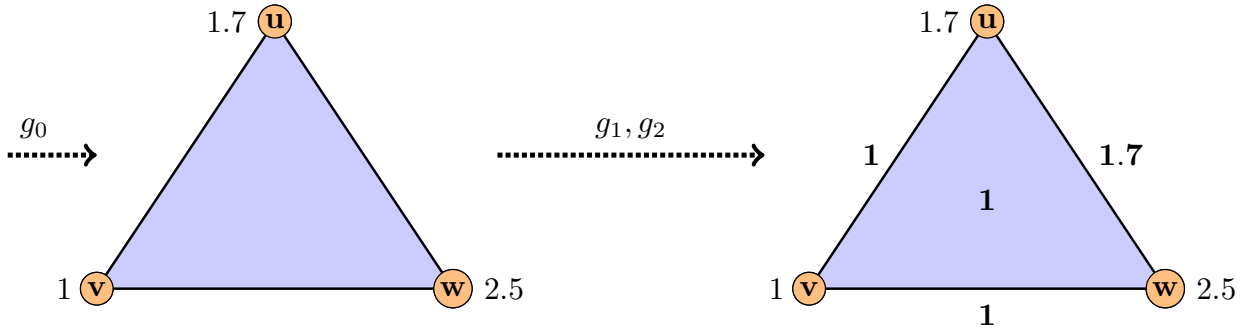


Figure 2.1: Example on how to extend g_0 for higher dimensional simplexes

2.1.1 Filtering functions

The first part of the algorithm consists on assigning numerical values to every q-simplex. For that end, we fix a real-valued function $g_0 : V_0 \to [a,b]$ with $[a,b]$ a closed interval. Later, we extend g_0 to higher dimensions q through auxiliary functions $g_q : V_q(K) \to [a,b]$ which are constructed as follows: for each q-simplex $\sigma_q = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_q)$ with $\mathbf{v}_i \in V_0(K)$, define

$$g_q(\sigma_q) = g_q((\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_q)) := \min_{0 \le i \le q} \{g_0(\mathbf{v}_i)\}. \tag{2.1.1}$$

For a given threshold value $t \in \mathbb{R}$ and a fixed dimension q we define the subsets of q-simplexes $V_q^{(i)}(K) := \{\sigma \in V_q : g_q(\sigma) > t_i\}$. The *Euler Characteristic at threshold* t *of* K is then defined as

$$\chi_i(K) = \sum_{q=0}^{d} (-1)^q \left| V_q^{(i)}(K) \right|. \tag{2.1.2}$$

Suppose that for a given q-simplex σ one of its vertices is $\mathbf{v}$. If $g_0(\mathbf{v}) \le t$. then by definition of g_q it follows that $g_q(\sigma) \le g_0(\mathbf{v}) \le t$. In other words, if a vertex fails to surpass a given threshold, then every q-simplex that contains it will fail to surpass such threshold as well.

For example, an edge $(\mathbf{v}, \mathbf{u})$ has two vertices, each of them with an associated numerical value. The numerical value assigned to such edge will be the smaller of those two vertex assigned numbers. The extension is analogous for a triangle. Refer to Figure 2.1. For instance, g_0 could be the distance from the vertex to the center of mass of the simplicial complex, or it could be the mean curvature at such point.

2.1.2 Thresholding

Once every simplex has been assigned a numerical value, the interval $[a, b]$ is divided into T equally-spaced thresholds $a = t_0 < t_1 < t_2 < \ldots < t_T = b$. Relabel χ_{t_i} as χ_i in this case. Finally, the *Euler Characteristic Graph (ECG)* of the complex K is simply defined as the graph given by $\chi_i\ (K)$ versus t_i. Notice that as the threshold value increases, there must be less q-simplexes that surpass such threshold. Qualitatively speaking, the ECG is a descriptor that summarizes the topological-geometrical changes of the object as it disintegrates through the filtration. It is expected that a larger number of thresholds will yield a sharper ECG, as it ought to pick smaller variations of the Euler Characteristic as the simplicial complex is filtered.

For example, take a pre-Columbian mask 3D mesh as a 2-dimensional simplicial complex embedded in the unitary sphere $S^2 \subset \mathbb{R}^3$ with its barycenter at the origin. Define g_0 as the squared distance from each vertex to the barycenter of such complex. In other words, we can take $g_0(\mathbf{v}) = g_0((x, y, z)) = 2 - (x^2 + y^2 + z^2)$ and extend it to edges and triangles as in (2.1.1). An example of 4 different masks and their filtrations using this filter function can be seen in Figures 2.2-2.5. Later their ECGs are plotted using different numbers of thresholds T as seen in Figure 2.6.

2.2 ECG computing algorithm

Suppose a fixed dimension $q > 0$. Assuming that the numerical values of function g_0 have already been computed, assigning numerical values to every q-simplex is an $O(|V_q|)$ complexity algorithm as g_q by definition only requires the comparison of a small number of quantities.

The computation $|V_q^{(i)}(K)|$ for each $1 \leq i \leq T$ and $1 \leq q \leq d$ can be achieved in $O(|V_0|)$ time and $O(T)$ memory using a bucket-sort-like `for` cycle. That is, consider a histogram $\mathcal{H}_q$ on $[a, b]$ with bins

$$B_q^{(k)} := \left(k\,\frac{b-a}{T},\ (k+1)\,\frac{b-a}{T} \right], \quad 0 \leq k \leq T-1. \tag{2.2.1}$$

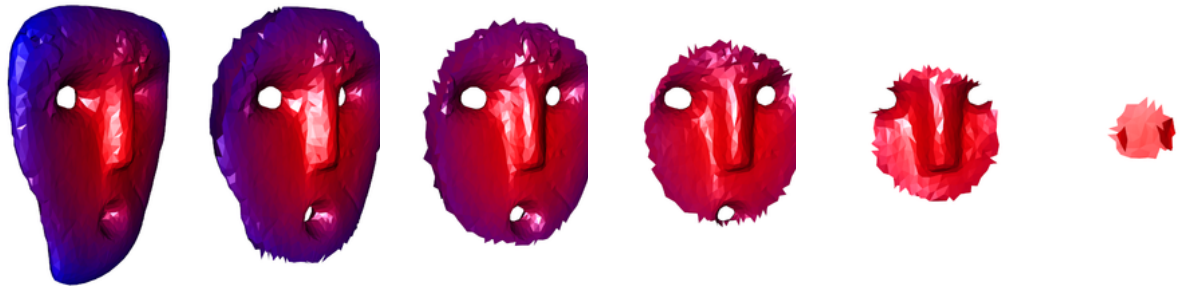


Figure 2.2: Destroying the mask CIII_0026_0001 with g_0 as 1 minus the Eucliedean distance to its center of mass

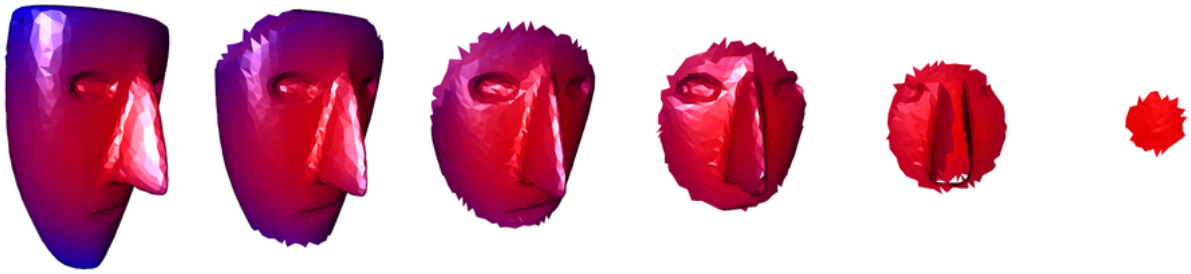


Figure 2.3: Destroying the mask CIII_0196_0001 with g_0 as 1 minus the Eucliedean distance to its center of mass

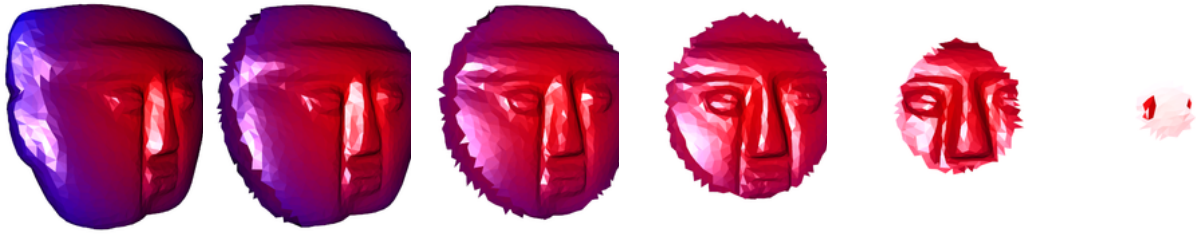


Figure 2.4: Destroying the mask CIII_0161_0001 with g_0 as 1 minus the Eucliedean distance to its center of mass

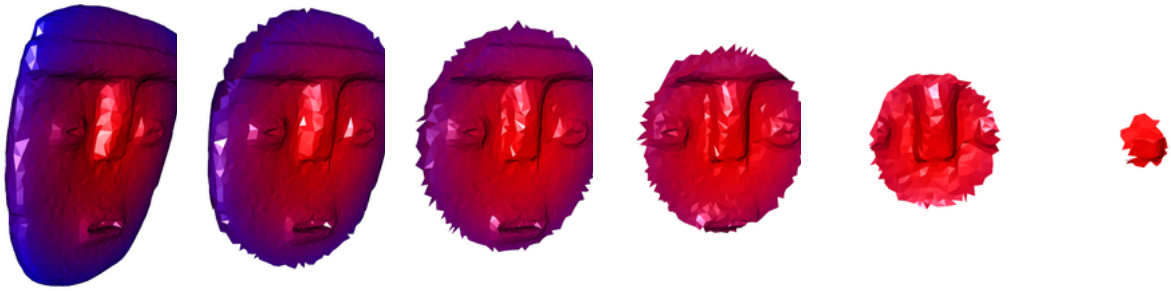
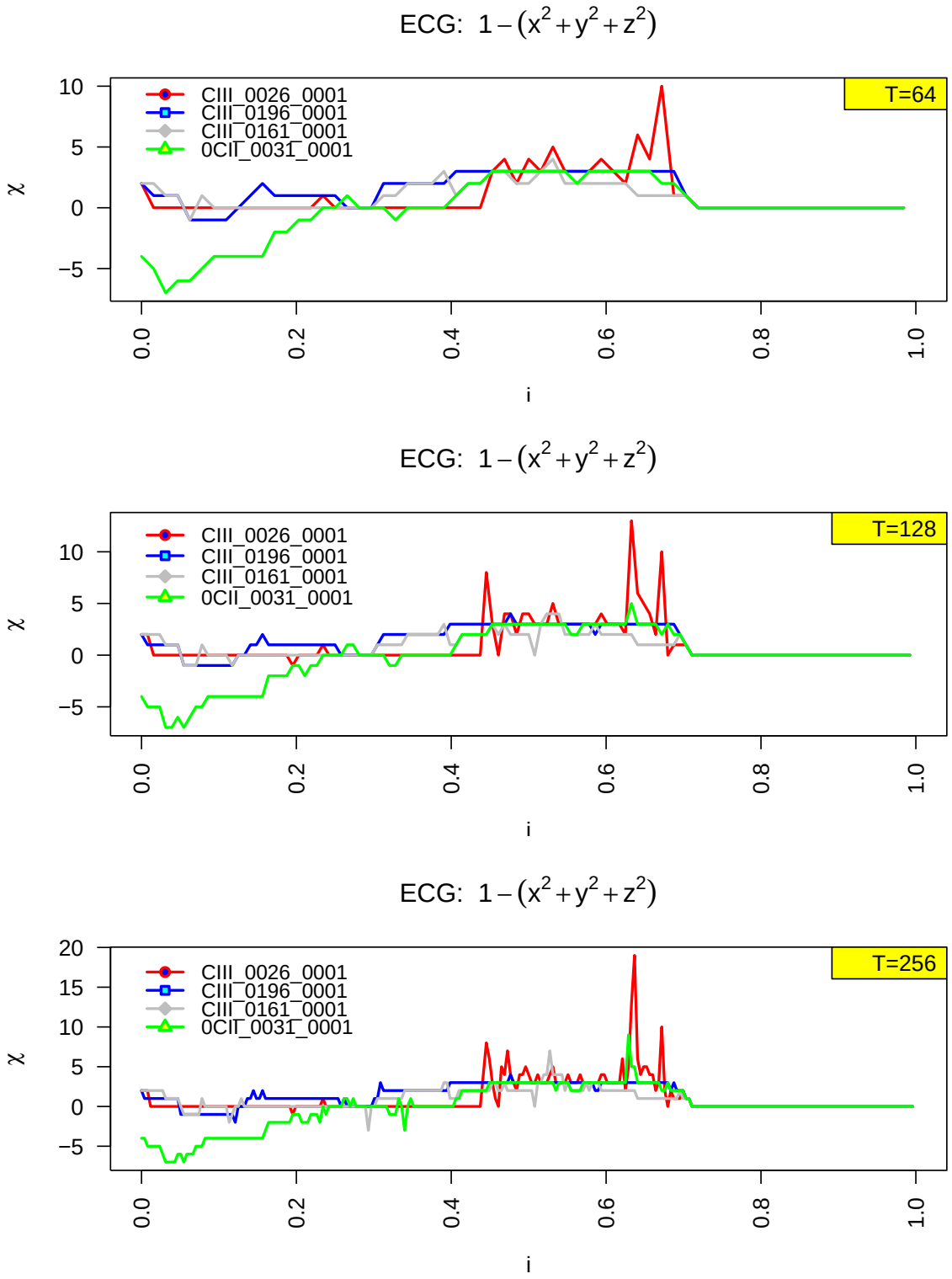


Figure 2.5: Destroying the mask 0CII_0031_0001 with g_0 as 1 minus the Eucliedean distance to its center of mass

Figure 2.6: ECGs with 64, 128 and 256 thresholds for g_0 as 1 minus the Euclidean distance to its center of mass.



Then for every q-simplex σ in constant time we compute

$$b_q(\sigma) := \left\lfloor \frac{1}{b-a}\, T \cdot g_q(\sigma) \right\rfloor \in \{0, 1, \ldots, T-1\}. \tag{2.2.2}$$

Note that $g_q(\sigma) \in B_q^{b_q(\sigma)}$. After running through all the q-simplexes applying (2.2.2), if frequency of the i-th bin is denoted by freq(B_k), then we see that

$$\left|V_q^{(i)}(K)\right| = \sum_{k=i}^{T-1} \text{freq}\left(B_q^{(k)}\right). \tag{2.2.3}$$

Computing $|V_q^{(i)}|$ for every threshold and dimension is thus achieved in $O\left(|V_q|\right)$ time and stored in $O(T)$ memory. Finally, observe that

$$\chi_i\ (K) = \sum_{q=1}^{d} (-1)^q \left|V_q^{(i)}(K)\right| = \sum_{q=1}^{d} (-1)^q \sum_{k=i}^{T-1} \text{freq}\left(B_q^{(k)}\right) \tag{2.2.4}$$

$$= \chi_{i+1}\ (K) + \sum_{q=1}^{d} (-1)^q\, \text{freq}\left(B_q^{(k)}\right). \tag{2.2.5}$$

So we may compute every $\chi_i\ (K)$ in time $O(T)$. Thus, the whole algorithm runs with complexity $O(|V_0|+T)$, demanding $O(T)$ memory. Overall, computation of ECGs is a very efficient algorithm. Algorithm 1 presents a possible pseudocode to implement the ECG computation. It must be noted that the algorithm in [RW14] assumes that the numerical values are first assigned to the simplices of highest dimension and later extends the value to lower dimensions. Our proposed algorithm first assigns values to vertices and later extends the values to higher dimensions.

The ECG of the complex K can actually be thought as the vector

$$\chi(K) := (\chi_0\ (K), \chi_1\ (K), \ldots, \chi_{T-1}\ (K)) \in \mathbb{R}^T.. \tag{2.2.6}$$

Assume now different filter functions $g_0^{(1)}, \ldots, g_0^{(n)}$. Each of them produces different ECG vectors $\chi^{1)}\ (K), \chi^{2)}\ (K), \ldots, \chi^{n)}\ (K)$ as mentioned above. The *composite ECG* obtained by the filter functions above is the concatenation of vectors $\chi^{1)}\ (K), \ldots, \chi^{n)}\ (K)$. We will denote the *composite ECG filter function* as g_0, where

$$g_0 := g_0^{(1)} \oplus \ldots \oplus g_0^{(n)} \tag{2.2.7}$$

$$\text{with composite ECG vector } \chi(K) := (\chi^{1)}\ (K), \ldots, \chi^{n)}\ (K)). \tag{2.2.8}$$

Algorithm 1 ECG Computation

Input: $g_0,\ T$ ▷ g_0 *computed already for every vertex*
$\chi[T] \leftarrow 0$
for all $q = 0 \to d$ **do** ▷ *dimensions*
    $\mathcal{H}_q[T] \leftarrow 0$ ▷ *set up a histogram with* T *empty bins*
    for all $i = 1 \to |V_q(K)|$ **do** ▷ *for every* q*-simplex*
        $g_k(\sigma_i) =\leftarrow \min g_0$ ▷ *Compute as in* (2.1.1)
        $b \leftarrow b_q(\sigma)$ ▷ *Compute the bin as shown in* (2.2.2)
        $\mathcal{H}_q[b] = \mathcal{H}_q[b] + 1$ ▷ *Increase by 1 the frequency of the* b*-th bin*
$c \leftarrow 0$ ▷ *Once all* q*-simplex values have been computed and tallied*
for all $i = T - 1 \to 0$ **do** ▷ *thresholds*
    for all $q = 0 \to d$ **do**
        $c \leftarrow c + (-1)^q\, \mathcal{H}_q[i]$
    $\chi[i] \leftarrow\chi[i] + c$ ▷ χ_i *is computed via* (2.2.5).
return χ ▷ *The Euler Characteristic Graph*

From here onwards the 0 subscript will be omitted as it is clear that the filter functions are determined when defined for the vertex. For some fixed positive scalar M we would rather work with filter functions of the form

$$f(\mathbf{v}) := M - g(\mathbf{v}),\ \text{such that } f(\mathbf{v}) \geq 0\ \forall\, \mathbf{v} \in V(K). \tag{2.2.9}$$

That way, we may assume that the range for the filter function f is $[0, M]$ and the size of each bin (2.2.1) would be M/T. Suppose now that both scalars M_1, M_2 satisfy (2.2.9) by defining filter functions $f_i := M_i - g$ for $i = 1, 2$. Suppose they yield two T-dimensional ECGs χ^1, χ^2 of the same simplicial complex K. If $M_1 < M_2$, then the bins 2.2.1 to compute χ^1 will be smaller. This in turn is expected to yield a finer ECG as smaller bins are expected to catch sharper details.

Our data results suggest that similar ECGs of two given objects indicate topological and geometrical similarity between these two as the Euler characteristic is a topological invariant as stated by Theorem 1.5.3. We must also note nonetheless that there is

currently no known stability result as with persistent homology in [CEH07]. Thus similar ECGs strongly suggest that those objects should belong into the same category. To determine similarity between graphs, supervised and unsupervised machine learning algorithms can be used. This particular work exploited supervised and unsupervised Support Vector Machines (SVM). The theory behind SVMs is explored throughout the next two chapters.

Chapter 3

Supported Vector Machines

The second key aspect of the work is the automation of the classification task of a data set into several subsets. A computer must be able to tell differences apart. Several classification techniques in machine learning are deemed as *supervised algorithms*. This means that the user provides the machine with explicitly classified data, referred as *training data*, from which the machine *learns* by identifying certain patterns within each subset. The *Support Vector Machine (SVM)* technique has been a main focus of attention within the supervised machine learning algorithms.

In chapter 2 we already described a procedure to describe a simplicial complex K as data in $\mathbb{R}^T$ for any given $T \in \mathbb{N}$. Whenever there are only two possible labels, a SVM aims to split $\mathbb{R}^T$ in two semi-spaces via an "optimal" hyperplane, each semi-space corresponding to a different label. That way, for new data the computer just need to check in which side of the hyperplane does the data lies. In this chapter we will state more carefully the SVM problem and define what do we mean by optimal hyperplane.

Theoretically, the SVM problem reduces to a constrained optimization problem that can be solved with the Karush-Kuhn-Tucker conditions. This constrained optimization definitions and results were not explicitly stated in most of the computational literature we reviewed. We thus include a section on optimization theory and later make explicit computations to see how do this theory relates to SVMs. This section mainly merges the discussion on constrained optimization [Fle00, Ch. 7] and the SVM tutotial [Bur98].

3.1 Linear Support Vector Machines

Consider a set of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ in $\mathbb{R}^d$ with labels $y_1, y_2, \ldots, y_n$ in $\mathbb{Z}$ respectively. Suppose that if $y_i = y_j$ then $\mathbf{x}_i$ and $\mathbf{x}_j$ must belong to the same family. The goal is to define a classification function $\Phi : \mathbb{R}^d \to \mathbb{Z}$ such that $\Phi(\mathbf{x}_i) = y_i$ for every $i = 1, 2, \ldots, n$. The basic setting for a *linear SVM* is as follows. Assume that each y_i is either 1 or -1, that is, there are only two different families. The goal is to find a hyperplane $\mathbf{H}$ such that it splits $\mathbb{R}^d$ so that each connected component of $\mathbb{R}^d \setminus \mathbf{H}$ corresponds to one of the labels. This in turn is equivalent to find a normal vector $\mathbf{w}$ and a scalar b to define $\mathbf{H}$ as $\{\mathbf{z} \in \mathbb{R}^d : \langle \mathbf{z}, \mathbf{w} \rangle + b = 0\}$. Hence, the classification function Φ would be the side of the hyperplane where each vector lies, i.e., $\Phi(\mathbf{x}) = \text{sgn}(\langle \mathbf{x}, \mathbf{w} \rangle + b)$ where "sgn" is the sign function. For example say the two labels $-1, 1$ correspond to circular and triangular data as in Figure 3.1.

3.1.1 The separable case

Say the hyperplane $\mathbf{H}$ defined by $\mathbf{w}$ and b separates both families. As vector $\mathbf{w}$ is nonzero, there must be an scalar λ such that the vector $\mathbf{r} := \lambda \mathbf{w}$ is in the hyperplane $\mathbf{H}$. Then it must be that

$$0 = \langle \mathbf{r}, \mathbf{w} \rangle + b = \lambda \langle \mathbf{w}, \mathbf{w} \rangle + b = \lambda \|\mathbf{w}\|^2 + b.$$

It follows that $|\lambda| = |b|/\|\mathbf{w}\|^2$, and $\|\mathbf{r}\| = |\lambda| \|\mathbf{w}\| = |b|/\|\mathbf{w}\|$, which in turn is the shortest distance from $\mathbf{H}$ to the origin.

Let d_+ (respectively d_-) represent the shortest distance from $\mathbf{H}$ to the closest positive (respectively negative) labeled vector. Then the *margin* of such hyperplane is defined as $m_{\mathbf{H}} = d_+ + d_-$. An SVM will look for the hyperplane $\mathbf{H}$ which maximizes its margin $m_{\mathbf{H}}$. The condition that $\mathbf{H}$ splits the data accordingly can be expressed as the following constraints.

$$\langle \mathbf{x_i}, \mathbf{w} \rangle + b \geq +1, \text{ if } y_i = +1; \tag{3.1.1a}$$

$$\langle \mathbf{x_i}, \mathbf{w} \rangle + b \leq -1, \text{ if } y_i = -1. \tag{3.1.1b}$$

Both of them are summarized in just one constraint,

$$y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1 \geq 0 \text{ for all } i = 1, \ldots, n. \tag{3.1.2}$$

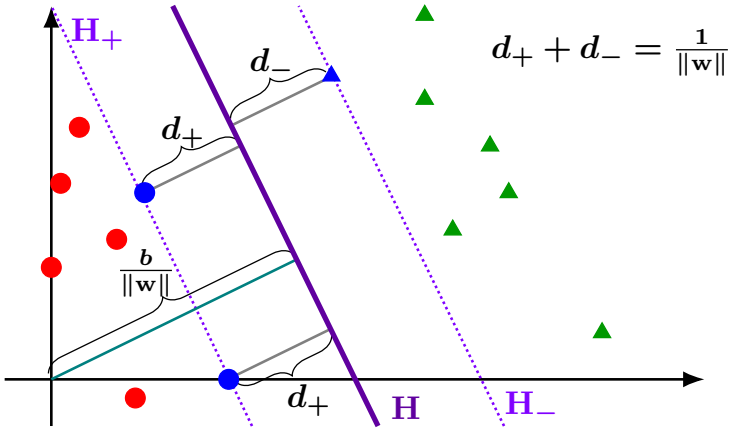


Figure 3.1: The separable case in $\mathbb{R}^2$. The support vectors are drawn in blue.

Now suppose the equality in constraint (3.1.1a) holds for some $\mathbf{x_i}$. Then $\mathbf{x_i}$ lies on the hyperplane $\mathbf{H}_+ := \{\mathbf{z} : \langle \mathbf{w}, \mathbf{z} \rangle + b = 1\}$. Notice that the distance between $\mathbf{H}_+$ and the origin is $|b-1|/\|\mathbf{w}\|$. Analogously, if equality holds for constraint (3.1.1b), there must be a vector $\mathbf{x_j}$ which lies on a hyperplane $\mathbf{H}_-$ whose distance from the origin is $|b+1|/\|\mathbf{w}\|$. That way is seen that $d_+ = d_- = 1/\|\mathbf{w}\|$ and $m_H = 2/\|\mathbf{w}\|$. Hence the algorithm's focus lies on minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ subject to constraint (3.1.2).

It is worth mentioning that the removal of any training vector that doesn't lie on either $\mathbf{H}_+$ or $\mathbf{H}_-$ will not affect the final solution. Hence the important vectors are only the ones lying on those hyperplanes. Such vectors are called *support vectors* from which the algorithm's name is derived. The previous reasoning can be rewritten as computing the following constrained optimization problem with objective function $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ and constraints $c_i : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$.

$$\min_{(\mathbf{w},b)\in\mathbb{R}^d\times\mathbb{R}} f(\mathbf{w}, b) := \frac{1}{2}\|\mathbf{w}\|^2, \tag{3.1.3a}$$

$$\text{such that } c_i(\mathbf{w}, b) := y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 1 \text{ for all } i = 1, \ldots, n. \tag{3.1.3b}$$

In order to solve the problem above it is necessary to develop some basic understanding of constrained optimization. Lagrange multipliers and their generalization, known as the Karush-Kuhn-Tucker (KKT) conditions, will be particularly useful. The following section is mostly based on Chapter 9 of [Fle00].

3.2 Constrained Optimization

In general, an optimization problem takes an *objective function* $f : \mathbb{R}^d \to \mathbb{R}$ and looks for its minimum

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}). \tag{3.2.1}$$

This problem can be complicated when the possible $\mathbf{x}$ must also satisfy a set of *constraints*, which we express as equalities or inequalities in the functions $c_i : \mathbb{R}^d \to \mathbb{R}$. More specifically, the structure of most constraint optimization problems is usually expressed as

$$\min f(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^d, \tag{3.2.2a}$$

$$\text{where } c_i(\mathbf{x}) = 0, \qquad i \in E, \tag{3.2.2b}$$

$$c_i(\mathbf{x}) \geq 0, \qquad i \in I. \tag{3.2.2c}$$

Through the next subsections, we will explore necessary and sufficient conditions for a point $\mathbf{x}$ to be a solution of system (3.2.2). In order to establish our results we will assume that f and c_i are C^1 functions. Notice that if $I = \varnothing$, then (3.2.2) consists of only equality constraints and it can be solved via Lagrange multipliers. However, having inequality constraints makes the problem more difficult to deal with. H. Kuhn and A. Tucker were the first to announce a rigorous way to deal with such problems in [KT51]. We focus mostly on inequality constraints here.

We first define the subset of *active constraint indexes* by $\mathcal{A}(\mathbf{x}) := \{i \in E \cup I \,:\, c_i(\mathbf{x}) = 0\}$. We may omit $\mathbf{x}$ whenever there is no confusion in this regard. Notice that if $\mathbf{x}$ is a local minimum of the unconstrained problem (3.2.1) and it happens that $c_i(\mathbf{x}) > 0$ for all $i \in I$, then it is also a local minimum of the constrained problem (3.2.2). Thus we can safely disregard such cases and just focus on the *active inequalities*, that is, only consider the indexes in E and $I' := I \cap \mathcal{A}$. Assume for a moment that $E = \varnothing$, that is, (3.2.2) consists only of inequality constraints.

3.2.1 First Order Conditions

In order to develop the necessary theory, we must first establish some definitions. We will say $\mathbf{x} \in \mathbb{R}^d$ is a *feasible point* if $c_i(\mathbf{x}) = 0$ and $c_i(\mathbf{x}) \geq 0$ for all $i \in E$ and $i \in I$

respectively as in (3.2.2b), (3.2.2c). Assume we can take a feasible sequence of points $\{\mathbf{x_k}\}$ such that $\mathbf{x_k} \to \mathbf{x}$ and $\mathbf{x_k} \neq \mathbf{x}$ for all $k \in \mathbb{N}$. Then we write

$$\mathbf{x_k} - \mathbf{x} = \delta_k \mathbf{s_k} \quad \forall k \in \mathbb{N}, \tag{3.2.3}$$

where $\delta_k > 0$ scalar and $\mathbf{s_k}$ is a unitary vector. Then it must be that $\delta_k \to 0$. If we also assume that the sequence $\{\mathbf{s_k}\}$ converges, we will say it is a *directional sequence* which converges to a *feasible direction* $\mathbf{s}$. Finally, define $\mathcal{F}(\mathbf{x})$ as the *set of all feasible directions* of the vector $\mathbf{x}$. We may omit $\mathbf{x}$ whenever there is no confusion about the basis point.

It is convenient to consider the 1st order Taylor series approximation about $\mathbf{x}$ for each constraint:

$$c_i(\mathbf{x} + \boldsymbol{\delta}) = c_i(\mathbf{x}) + \langle \nabla c_i(\mathbf{x}), \boldsymbol{\delta} \rangle + o(\delta_k). \tag{3.2.4}$$

This yields a related set of feasible directions which are obtained by linearized constraints, which we denote by $F(\mathbf{x})$, i.e.,

$$F(\mathbf{x}) = \{\mathbf{s} \ : \ \mathbf{s} \neq 0, \langle \mathbf{s}, \nabla c_i(\mathbf{x}) \rangle = 0, i \in E, \langle \mathbf{s}, \nabla c_i(\mathbf{x}) \rangle \geq 0, i \in I'\}. \tag{3.2.5}$$

As $\mathbf{s}$ is of norm 1, the inner product of $\nabla c_i(\mathbf{x})$ with $\mathbf{s}$ produces the directional derivative of c_i along the direction $\mathbf{s}$. Assume we are standing on a feasible point $\mathbf{x}$ as on the green point in Figure 3.2. In the case of c_i with $i \in E$, we know $c_i(\mathbf{x}) = 0$. This equality must hold for any other feasible point. Thus the directional derivative along all the feasible directions $\mathbf{s}$ from $\mathbf{x}$ must vanish. In other words, $\mathbf{s}$ and $\nabla c_i(\mathbf{x})$ must be perpendicular. In the case when $i \in I'$, $c_i(\mathbf{x}) = 0$ again. However, in this case we are allowed to increase the value of $c_i(\mathbf{x})$ along any feasible direction $\mathbf{s}$. That is, the directional derivative of c_i along $\mathbf{s}$ must be nonnegative.

We will now see that $\mathcal{F}$ and F are closely related.

Lemma 3.2.1. *For any feasible* $\mathbf{x}$, $\mathcal{F} \subset F$.

Proof. Let $\mathbf{s}$ be feasible direction of $\mathbf{x}$. That is, there exists a feasible sequence $\{\mathbf{x}_k\}$ such that $\mathbf{x}_k \to \mathbf{x}$ such that $\mathbf{s}_k \to \mathbf{s}$ as in (3.2.3). We expand the Taylor series of c_i about $\mathbf{x}$ to obtain:

$$c_i(\mathbf{x_k}) = c_i(\mathbf{x}) + \langle \delta_k \mathbf{s}_k, \nabla c_i(\mathbf{x}) \rangle + o(\delta_k). \tag{3.2.6}$$

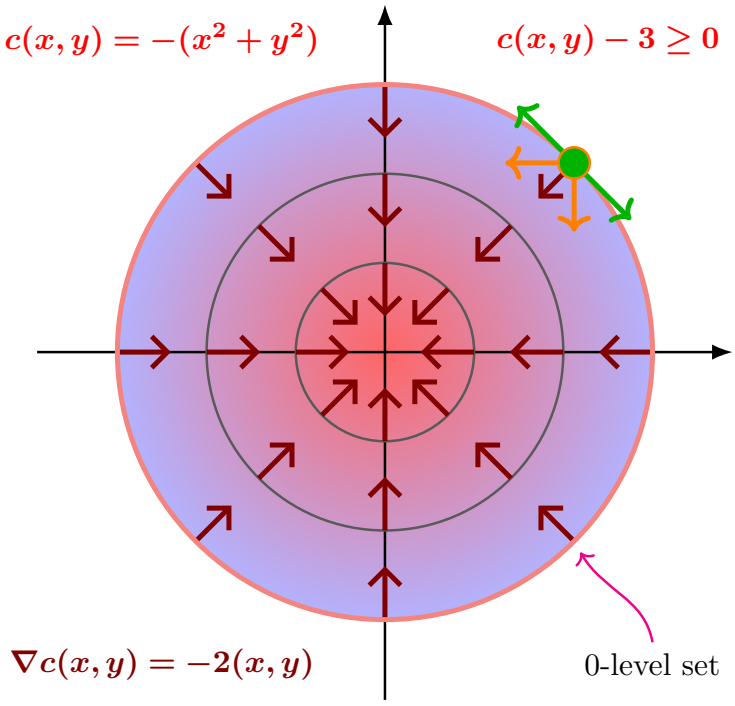


Figure 3.2: Level sets of $c_i(x,y) := -(x^2+y^2)$

Due to feasibility of each $\mathbf{x}_k$ and the definition of I', we see that

$$\begin{aligned} c_i(\mathbf{x}_k) &= c_i(\mathbf{x}) = 0, && \text{for } i \in E, \\ c_i(\mathbf{x}_k) &\geq c_i(\mathbf{x}) = 0, && \text{for } i \in I'. \end{aligned}$$

Dividing (3.2.6) by $\delta_k > 0$ it follows that

$$\begin{aligned} \langle \mathbf{s}_k, \nabla c_i(\mathbf{x}) \rangle + o(1) &= 0, \text{ for } i \in E; \\ \langle \mathbf{s}_k, \nabla c_i(\mathbf{x}) \rangle + o(1) &\geq 0, \text{ for } i \in I'. \end{aligned}$$

Taking $k \to \infty$, we see that $\mathbf{s}_k \to \mathbf{s}$ and $o(1) \to 0$. Thus, as $k \to \infty$,

$$\begin{aligned} \langle \mathbf{s}, \nabla c_i(\mathbf{x}) \rangle &= 0, \text{ for } i \in E; \\ \langle \mathbf{s}, \nabla c_i(\mathbf{x}) \rangle &\geq 0, \text{ for } i \in I'. \end{aligned}$$

By definition, we see that $s \in F(\mathbf{x})$. □

Unfortunately, the other inclusion does not hold in general. Take for example the feasible region R defined as shown in Figure 3.3. Notice that $\mathbf{0} \in R$, so clearly $\mathbf{s} = (-1, 0)$

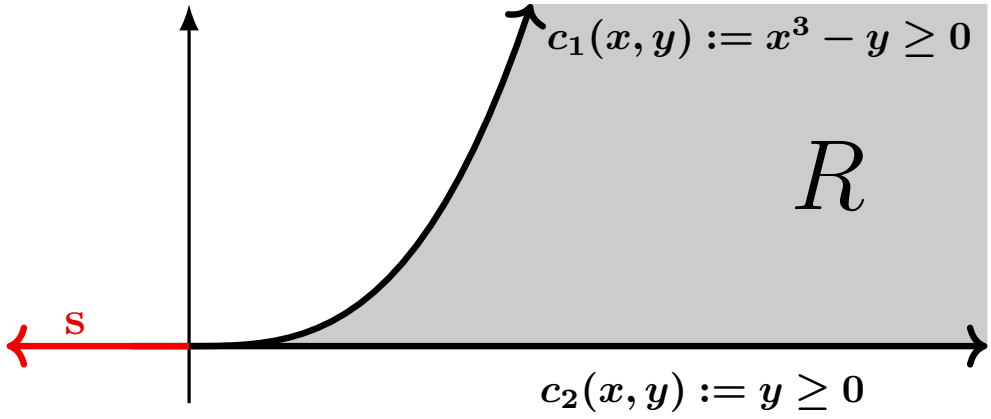


Figure 3.3: Feasible region $R \subset \mathbb{R}^2$ determined by constraints c_1 and c_2.

is not a feasible direction from the origin. However, $\nabla c_1(\mathbf{0}) = (0,-1)$ and $\nabla c_2(\mathbf{0}) = (0,1)$. Thus $\langle \mathbf{s}, \nabla c_i(\mathbf{0})\rangle \geq 0$ for both $i = 1,2$. That is, $\mathbf{s} \in F(\mathbf{0})$. Hence $\mathcal{F}(\mathbf{0}) \neq F(\mathbf{0})$.

Kuhn and Tucker simply make the assumption $\mathcal{F} = F$ which is referred as the *constraint qualification*. However, this qualification follows when we consider stronger hypothesis on constraints. For instance, the constraint qualification follows immediately if all active constraints $i \in \mathcal{A}$ are linear. A stronger result is as follows.

Lemma 3.2.2. *Sufficient conditions for $F = \mathcal{F}$ at a feasible point $\mathbf{x}$ are either*

1. *The constraints $i \in \mathcal{A}$ are all linear.*

2. *The vectors $\nabla c_i(\mathbf{x})$ for $i \in \mathcal{A}$ are all linearly independent.*

Proof. By definition of F, the equality is immediate if all active constraints are linear. On the other hand, if all vectors $\nabla c_i(\mathbf{x})$ are linearly independent, assume the active constraints are labeled as $\mathcal{A} = \{1,2,\ldots,m\}$. Observe that $m \leq d$ due to linear independence. Consider then a fixed linear feasible direction $\mathbf{s} \in F$ and a non-linear system $r : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$

$$r(\mathbf{z},\theta) = \mathbf{0}. \tag{3.2.7}$$

defined by

$$\begin{aligned} r_i(\mathbf{z},\theta) &= c_i(\mathbf{z}) - \theta\langle \mathbf{s}, \nabla c_i(\mathbf{x})\rangle, && i = 1,\dots m \\ r_i(\mathbf{z},\theta) &= \langle \mathbf{z}-\mathbf{x}, \mathbf{b}_i\rangle - \theta\langle \mathbf{s}, \mathbf{b}_i\rangle, && i = m+1,\dots,n. \end{aligned}$$

The vectors $b_i \in \mathbb{R}^d$ will be defined below. Observe that $(\mathbf{x}, 0)$ is a solution to the system 3.2.7. Observe that any solution $(\mathbf{z}, \theta)$ to the system above is also a feasible point in the general optimization problem 3.2.2 for sufficiently small $\theta \geq 0$. Keep in mind that actually 3.2.2 only considers active constraints based on §, $\mathcal{A}(\mathbf{x})$, as we can ignore locally the inactive ones.

Consider now the matrices defined by row vectors $A = [\nabla c_1(\mathbf{z}), \dots, \nabla c_m(\mathbf{z})]$ and $B = [\mathbf{b}_{m+1}, \dots, \mathbf{b}_n]$. We observe that the transposed Jacobian is

$$J(\mathbf{z},\theta)^\top = \begin{pmatrix} \frac{\partial c_1}{\partial z_1}(\mathbf{z}) & \cdots & \frac{\partial c_m}{\partial z_m}(\mathbf{z}) & b_{m+1,1} & \cdots & b_{d,1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_1}{\partial z_d}(\mathbf{z}) & \cdots & \frac{\partial c_m}{\partial z_d}(\mathbf{z}) & b_{m+1,d} & \cdots & b_{d,d} \end{pmatrix}^\top = [A : B]. \tag{3.2.8}$$

By hypothesis, A has full rank so it is possible to choose B such that $J^\top$ is nonsingular. Due to the implicit function theorem, there exists open neighborhoods $\Omega_{\mathbf{x}}$, Ω_0 around $\mathbf{x}$ and $\theta = 0$ respectively such that for every $\theta \in \Omega_0$, there exists a unique $\mathbf{x}(\theta) \in \Omega_{\mathbf{x}}$ such that $(\mathbf{x}(\theta), \theta)$ is a solution to system 3.2.7. Additionally, $\mathbf{x}(\cdot)$ can be thought as a C^1 function of θ.

From above, 3.2.7 and the chain rule, we know that

$$0 = \frac{dr_i}{d\theta} = \sum_{j=1}^{d}\left(\frac{\partial r_i}{\partial x_j}\frac{dx_j}{d\theta}\right) + \frac{\partial r_i}{\partial \theta} = \begin{pmatrix} \frac{\partial r_1}{\partial z_1}(\mathbf{z}) & \cdots & \frac{\partial r_1}{\partial z_n}(\mathbf{z}) \\ \vdots & \ddots & \vdots \\ \frac{\partial r_n}{\partial z_1}(\mathbf{z}) & \cdots & \frac{\partial r_n}{\partial z_n}(\mathbf{z}) \end{pmatrix} \begin{pmatrix} \frac{dz_1}{d\theta} \\ \vdots \\ \frac{dz_n}{d\theta} \end{pmatrix} + \begin{pmatrix} \frac{\partial r_1}{\partial \theta} \\ \vdots \\ \frac{\partial r_n}{\partial \theta} \end{pmatrix}. \tag{3.2.9}$$

Observe that $\frac{\partial r_i}{\partial \theta} = -\langle \mathbf{s}, \nabla c_i(\mathbf{x})\rangle$ for $i = 1, \dots, m$ and $\frac{\partial r_i}{\partial \theta} = -\langle \mathbf{s}, \mathbf{b}_i\rangle$ otherwise. Just as above, we can see that

$$\begin{pmatrix} \frac{\partial r_1}{\partial z_1}(\mathbf{x}) & \cdots & \frac{\partial r_1}{\partial z_n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial r_n}{\partial z_1}(\mathbf{x}) & \cdots & \frac{\partial r_n}{\partial z_n}(\mathbf{x}) \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_d \end{pmatrix} = - \begin{pmatrix} \frac{\partial r_1}{\partial \theta} \\ \vdots \\ \frac{\partial r_n}{\partial \theta} \end{pmatrix}. \tag{3.2.10}$$

Thus we conclude that

$$J(\mathbf{z},\theta)^\top \frac{d\mathbf{x}}{d\theta} = J(\mathbf{x},\theta)^\top \mathbf{s}.$$

Now, as $\theta = 0$ we know $(\mathbf{x}(\theta),\theta)$ is a solution. Due to uniqueness, $\mathbf{x}(0) = \mathbf{x}$. Then $\frac{d\mathbf{x}}{d\theta} = \mathbf{s}$ whenever $\theta = 0$. Finally, consider a nonnegative descending sequence $\theta_k \downarrow 0$. As any solution to 3.2.7 is in the feasible region of 3.2.2, every $\mathbf{x}_k := \mathbf{x}(\theta_k)$ is feasible. Thus $\{\mathbf{x}_k\}_k$ produces the feasible direction $\mathbf{s}$, proving that $\mathbf{s} \in \mathcal{F}$ as desired.

□

Define now the set of *descending directions*

$$\mathcal{D}(\mathbf{x}) = \{\mathbf{s} \,:\, \langle \mathbf{s}, \nabla f(\mathbf{x})\rangle < 0\}. \tag{3.2.11}$$

As with $\mathcal{F}$, we will omit $\mathbf{x}$ whenever it is clearly implied. Geometrically speaking, $\mathcal{D}$ is the set of all the possible directions based from $\mathbf{x}$ that immediately decrease the value of f. The next lemma simply states that once we are at a local minimum none of the feasible directions from there will immediately lead to a lower valued level set.

Lemma 3.2.3. *If $\mathbf{x}$ is a local minimum, then $\mathcal{F} \cap \mathcal{D} = \varnothing$.*

Proof. Let $\mathbf{s} \in \mathcal{F}$. That is, there exists a feasible sequence $\{\mathbf{x}_k\}$ such that $\mathbf{x}_k \to \mathbf{x}$ such that $\mathbf{s}_k \to \mathbf{s}$ as in (3.2.3). We expand with Taylor about $\mathbf{x}$ to obtain:

$$f(\mathbf{x}_k) = f(\mathbf{x}) + \langle \delta_k \mathbf{s}_k, \nabla f(\mathbf{x})\rangle + o(\delta_k). \tag{3.2.12}$$

Due to minimality, for a sufficiently large K,

$$f(\mathbf{x}_k) \geq f(\mathbf{x}) \quad \forall k > K.$$

Therefore, as $\delta_k > 0$,

$$\langle \mathbf{s}_k, \nabla f(\mathbf{x})\rangle + o(1) \geq 0.$$

As $k \to \infty$, $\mathbf{s}_k \to \mathbf{s}$ and $o(1) \to 0$. Thus $\langle \mathbf{s}, \nabla f(\mathbf{x})\rangle \geq 0$. That is, $\mathbf{s} \notin \mathcal{D}$. □

Unfortunately, we can't say the same when considering the larger set F. We must assume further the *regularity condition*

$$F \cap \mathcal{D} = \mathcal{F} \cap \mathcal{D}. \tag{3.2.13}$$

This condition certainly holds if K-T constraint qualification, $\mathcal{F} = F$, holds as well. However (3.2.13) may also hold even when such constraint qualification does not. For example, suppose we want to minimize the objective function $f(x, y) := y$ in the feasible region R defined by Figure 3.3. Notice that for any linear feasible direction $\mathbf{s} = (s_1, s_2)$ of the origin we have that $\langle \mathbf{s}, \nabla c_1(\mathbf{0}) \rangle = -s_2$ and $\langle \mathbf{s}, \nabla c_2(\mathbf{0}) \rangle = s_2$. It follows then that $s_2 = 0$. Thus, $\langle \mathbf{s}, \nabla f(\mathbf{0}) \rangle = s_2 \geq 0$ and $F(\mathbf{0}) \cap \mathcal{D}(\mathbf{0}) = \mathcal{F}(\mathbf{0}) \cap \mathcal{D}(\mathbf{0})$.

Figure 3.3 also shows the necessity of the regularity condition (3.2.13). Suppose we instead want to minimize the objective function $g(x, y) := x$. Take $\mathbf{s} = (-1, 0) \in F(\mathbf{0})$. Then $\langle s, \nabla g(\mathbf{0}) \rangle = -1 < 0$. Thus $F(\mathbf{0}) \cap \mathcal{D}(\mathbf{0}) \neq \varnothing$. From the previous lemma, assumption (3.2.13) tells us that no linear feasible direction is also a descending direction.

Theorem 3.2.4 (Farkas)**.** *Given any vectors* $\mathbf{a}_1, \ldots, \mathbf{a}_m$ *and* $\mathbf{g}$*, the set*

$$S = \{\mathbf{s} \,:\, \langle \mathbf{s}, \mathbf{g} \rangle < 0, \langle \mathbf{s}, \mathbf{a}_i \rangle \geq 0, i = 1 \ldots, m\} \tag{3.2.14}$$

is empty if and only if there exists nonnegative α_i *such that*

$$\mathbf{g} = \sum_{i=1}^{m} \alpha_i \mathbf{a}_i. \tag{3.2.15}$$

Proof. First suppose there exists $\mathbf{s} \in S$. If (3.2.15) holds, then we can take the interior product on both sides with $\mathbf{s}$. This yields

$$\langle \mathbf{s}, \mathbf{g} \rangle = \sum_i \alpha_i \langle \mathbf{s}, \mathbf{a}_i \rangle \geq 0.$$

This, of course, is a contradiction. Therefore, S must be empty.

Now assume (3.2.15) does not hold. Consider the polyhedral cone C generated by the $\mathbf{a}_i$'s, that is

$$C := \{\mathbf{v} \,:\, \mathbf{v} = \sum_{i=1}^{m} \alpha_i \mathbf{a}_i, \alpha_i \geq 0\}.$$

By assumption, $\mathbf{g} \notin C$. As the cone is a closed and convex subset of $\mathbb{R}^d$, there exists a hyperplane with normal vector $\mathbf{s}$ which separates C and $\mathbf{g}$. That is, $\langle \mathbf{s}, \mathbf{a}_i \rangle \geq 0$ and $\langle \mathbf{s}, \mathbf{g} \rangle < 0$. Therefore, $\mathbf{s} \in S$. That is, S is not empty. □

We can translate Farkas' lemma to the language of constraints.

Corollary 3.2.5.

$$S := \{\mathbf{s} : \langle \mathbf{s}, \nabla f(\mathbf{x})\rangle < 0, \langle \mathbf{s}, \nabla c_i(\mathbf{x})\rangle = 0, i \in E, \langle \mathbf{s}, \nabla c_i(\mathbf{x})\rangle \geq 0, i \in I' = I \cup \mathcal{A}\} \quad (3.2.16)$$

is empty if and only if there exist nonnegative multipliers α_i *such that*

$$\nabla f(\mathbf{x}) = \sum_{i \in \mathcal{A}} \alpha_i \nabla c_i(\mathbf{x}). \quad (3.2.17)$$

Proof. Remember $\mathcal{A}$ is the set of active equalities. The key part is to realize that the equalities $\langle \mathbf{s}, \mathbf{a}_i\rangle = 0$ can be rewritten as inequalities $\langle \mathbf{s}, \mathbf{a}_i\rangle \geq 0$ and $\langle \mathbf{s}, \mathbf{a}_i\rangle \leq 0$. That being said, due to Farkas' lemma there are nonnegative α_i, $i \in I'$ and α_i^+, α_i^-, $i \in E$ such that

$$\nabla f(\mathbf{x}) = \sum_{i \in I'} \alpha_i \nabla c_i(\mathbf{x}) + \sum_{i \in E} \alpha_i^+ \nabla c_i(\mathbf{x}) + \sum_{i \in E} \alpha_i^- \nabla c_i(\mathbf{x}).$$

if and only if the set S is empty. We redefine $\alpha_i := \alpha_i^+ + \alpha_i^-$ for $i \in E$ and we are done. □

Finally we establish necessary conditions for $\mathbf{x}$ to be a local minimum for the constrained problem (3.2.2). If $\mathbf{x}$ is a local minimum, there are no feasible descent directions by Lemma 3.2.3. With the regularity assumption (3.2.13), there are no linear feasible descent directions. By the Corollary 3.2.5 of Farkas' lemma, we get the existence of nonnegative multipliers and the fact that $\nabla f(\mathbf{x})$ is spanned by the gradients of the constraints. In other words, we have the following.

Theorem 3.2.6 (First Order Necessary Conditions, or Karush-Kuhn-Tucker Conditions)**.** *If* $\mathbf{x}$ *is a local minimum of* (3.2.2) *and if* (3.2.13) *holds at* $\mathbf{x}$*, then there are multipliers* $\{\alpha_i\}_i$ *such that the following system is satisfied:*

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) = 0, \text{ where } \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) - \sum_{i \in E \cup I} \alpha_i c_i(\mathbf{x}); \quad (3.2.18a)$$

$$c_i(\mathbf{x}) = 0, i \in E; \quad (3.2.18b)$$

$$c_i(\mathbf{x}) \geq 0, i \in I; \quad (3.2.18c)$$

$$\alpha_i \geq 0, i \in I; \quad (3.2.18d)$$

$$\alpha_i c_i(\mathbf{x}) = 0, \forall i. \quad (3.2.18e)$$

We will call the $\mathcal{L}$ *function in* (3.2.18a) *the* Lagrangian function *of the problem.*

The last condition arises since it is consistent to assume that inactive constraints will have 0 as their multiplier, as they play no actual role when looking for the solution. Note that if $I = \varnothing$, meaning we have no inequality constraints, then the Karush-Kuhn-Tucker conditions would reduce to the first two equations, which are the usual conditions for the method of Lagrange multipliers.

3.2.2 Convexity

If we assume further that we are working in convex spaces with convex functions, then it is possible to establish results about the sufficiency of Karush-Kuhn-Tucker conditions (3.2.18). We will need to establish some general definitions and results first.

A set $K \subset \mathbb{R}^d$ is said to be *convex* if for all $\mathbf{x}_0, \mathbf{x}_1$ each $\mathbf{x}_\theta := (1-\theta)\mathbf{x}_0 + \theta\mathbf{x}_1$ belongs to K for every $\theta \in (0,1)$. In other words, for every two points in K, the line segment connecting them lies completely within K. Unless stated otherwise, θ and $\mathbf{x}_\theta$ will be understood as above. From the definition the following lemma is clear.

Lemma 3.2.7. *If K_i are convex sets, then $K := \bigcap_i K_i$ is convex as well.*

For K a fixed convex set, a *convex function* $f : K \subset \mathbb{R}^d \to \mathbb{R}$ is one for which every pair of points $\mathbf{x}_0, \mathbf{x}_1$ in K the following inequality holds

$$f(\mathbf{x}_\theta) \leq (1-\theta)f(\mathbf{x}_0) + \theta f(\mathbf{x}_1). \tag{3.2.19}$$

If f is C^1 and K is open, an alternative definition of convex function is that for every pair of points $\mathbf{x}_0, \mathbf{x}_1 \in K$ we have

$$f(\mathbf{x}_1) \geq f(\mathbf{x}_0) + \langle \mathbf{x}_1 - \mathbf{x}_0, \nabla f(\mathbf{x}_0) \rangle. \tag{3.2.20}$$

Keeping in mind the Taylor approximation around $\mathbf{x}_1$, we see that, geometrically, the condition (3.2.20) states that the graph of f must lie above or along the tangent hyperplane of f at $\mathbf{x}_0$. We now check when both definitions are equivalent.

Lemma 3.2.8. *If $K \subset \mathbb{R}^d$ is a fixed convex, open set and $f : K \to \mathbb{R}$ is C^1, then properties* (3.2.19) *and* (3.2.20) *are equivalent.*

Proof. If (3.2.19) holds, then we have

$$\frac{f(\mathbf{x}_\theta) - f(\mathbf{x}_0)}{\theta} = \frac{f(\mathbf{x}_0 + \theta(\mathbf{x}_1 - \mathbf{x}_0)) - f(\mathbf{x}_0)}{\theta} \leq f(\mathbf{x}_1) - f(\mathbf{x}_0).$$

Taking $\theta \to 0$ and using the definition of directional derivative we obtain (3.2.20).

Now if (3.2.20) holds, then we have

$$\begin{aligned} f(\mathbf{x}_1) \geq & f(\mathbf{x}_\theta) + \langle \mathbf{x}_1 - \mathbf{x}_\theta, \nabla f(\mathbf{x}_\theta) \rangle; \\ f(\mathbf{x}_0) \geq & f(\mathbf{x}_\theta) + \langle \mathbf{x}_0 - \mathbf{x}_\theta, \nabla f(\mathbf{x}_\theta) \rangle. \end{aligned}$$

We compute

$$\begin{aligned} (1-\theta) f(\mathbf{x}_\theta) + \theta f(\mathbf{x}_1) &\geq f(\mathbf{x}_\theta) + \langle (1-\theta)\mathbf{x}_1 + \theta \mathbf{x}_0, \nabla f(\mathbf{x}_\theta) \rangle \\ &= f(\mathbf{x}_\theta) + \langle \mathbf{x}_\theta, \nabla f(\mathbf{x}_\theta \rangle \\ &= f(\mathbf{x}_\theta), \end{aligned}$$

which is exactly (3.2.19). □

From (3.2.20) it also follows that

$$\langle \mathbf{x}_1 - \mathbf{x}_0, \nabla f(\mathbf{x}_1) \rangle \geq f(\mathbf{x}_1) - f(\mathbf{x}_0) \geq \langle \mathbf{x}_1 - \mathbf{x}_0, \nabla f(\mathbf{x}_0) \rangle. \tag{3.2.21}$$

That is, the slope of a convex function is non-decreasing along any line.

With the definition (3.2.19), it is easy to see the following result.

Lemma 3.2.9. *If $f_1, \ldots, f_m$ are convex functions on the same convex set K, and if $\alpha_1, \ldots, \alpha_m$ are nonnegative, then $\sum_i \alpha_i f_i$ is a convex function on K.*

We say that f is a *concave function* if $-f$ is a convex function. A *convex programming problem* is similar to the one presented in (3.2.2) as follows:

$$\min \ f(\mathbf{x}) \text{ subject to } \ \mathbf{x} \in K := \{\mathbf{x} \in \mathbb{R}^d \ : \ c_i(\mathbf{x}) \geq 0, i = 1, \ldots, m\}, \tag{3.2.22}$$

where f is a convex function on K and $c_i : \mathbb{R}^d \to \mathbb{R}$ are concave constraints. Notice that K is convex automatically, due to the following.

Lemma 3.2.10. *If c is concave, then $S_k := \{\mathbf{x} \ : \ c(\mathbf{x}) \geq k\}$ is convex.*

Proof. For any $\mathbf{x}_0, \mathbf{x}_1 \in S_k$ we have due to concavity

$$c(\mathbf{x}_\theta) \geq (1-\theta)c(\mathbf{x}_0) + \theta c(\mathbf{x}_1) \geq (1-\theta)k + \theta k = k$$

Thus $\mathbf{x}_\theta \in S_k$ and we are done. □

As each constraint is a concave function, each one has its own convex feasible region. Thus K is the intersection of convex sets, and by Lemma 3.2.7 we conclude K is convex. One important property of convex programming problems lies in the following theorem.

Theorem 3.2.11. *Every local solution* $\mathbf{x}$ *to a convex programming problem* (3.2.22) *is a global solution. Additionally, the set of global solutions* S *is convex.*

Proof. Let $\mathbf{x}$ be a local but not a global solution. Then there exists a $\mathbf{x}_1 \in K$ such that $f(\mathbf{x}_1) < f(\mathbf{x})$. Due to convexity of f we see that

$$f(\mathbf{x}_\theta) \leq (1-\theta)f(\mathbf{x}) + \theta f(\mathbf{x}_1) = f(\mathbf{x}) + \theta(f(\mathbf{x}_1) - f(\mathbf{x})) < f(\mathbf{x}).$$

As $\theta \to 0$ we see that $\mathbf{x}$ is not even a local solution, which is a contradiction.

Now let $\mathbf{x}_0, \mathbf{x}_1 \in S$. Since both of them are global solutions we know that $f(\mathbf{x}_\theta) \geq f(\mathbf{x}_0) = f(\mathbf{x}_1)$. By convexity on the other hand, $f(\mathbf{x}_\theta) \leq (1-\theta)f(\mathbf{x}_0) + \theta f(\mathbf{x}_1) = f(\mathbf{x}_0)$, and it follows that $f(\mathbf{x}_\theta) = f(\mathbf{x}_0)$. So $\mathbf{x}_\theta \in S$. □

Finally, we are able to show that the Karush-Kuhn-Tucker conditions are sufficient for convex problems.

Theorem 3.2.12. *In the convex programming problem, if* f *and* c_i *are* C^1 *functions on an open and convex* K *and if conditions* (3.2.18) *hold at* $\mathbf{x} \in K$*, then* $\mathbf{x}$ *is a global solution to* (3.2.22).

Proof. Let $\mathbf{x}' \in K \setminus \{\mathbf{x}\}$. As $\alpha_i \geq 0$ and $c_i(\mathbf{x}') \geq 0$, we have

$$\begin{aligned} f(\mathbf{x}') &\geq f(\mathbf{x}') - \sum_{i=1}^{m} \alpha_i c_i(\mathbf{x}') \\ &\geq f(\mathbf{x}) + \langle \mathbf{x}' - \mathbf{x}, \nabla f(\mathbf{x}) \rangle - \sum_{i=1}^{m} \alpha_i (c_i(\mathbf{x}) + \langle \mathbf{x}' - \mathbf{x}, \nabla c_i(\mathbf{x}) \rangle). \end{aligned}$$

The last inequality is due to the fact that f and $-c_i$ are convex and satisfy equation (3.2.20). From (3.2.18), we know that $\alpha_i c_i(\mathbf{x}) = 0$ and $\nabla f(\mathbf{x}) = \sum_i \alpha_i \nabla c_i(\mathbf{x})$. Substituting this in the expression above yields

$$\begin{aligned} f(\mathbf{x}') &\geq f(\mathbf{x}) + \langle \mathbf{x}' - \mathbf{x}, \sum \alpha_i \nabla c_i(\mathbf{x}) \rangle - \sum \alpha_i c_i(\mathbf{x}) - \sum \langle \mathbf{x}' - \mathbf{x}, \alpha_i \nabla c_i(\mathbf{x}) \rangle \\ &= f(\mathbf{x}). \end{aligned}$$

Therefore, $\mathbf{x}$ is a global solution. □

Just as in Corollary 3.2.5, we can add equality constraints $c_i(\mathbf{x}) = 0$ to the convex programming problem (3.2.22) by considering the inequality constraints $c_i(\mathbf{x}) \geq 0$ and $-c_i(\mathbf{x}) \geq 0$. Thus, if (3.2.22) satisfies the regularity condition (3.2.13), by combining Theorems 3.2.6 and 3.2.12 we see that Karush-Kuhn-Tucker conditions (3.2.18) are sufficient and necessary for $\mathbf{x}$ to be a minimum. Moreover, due to Theorem 3.2.11, these minimums will be global.

3.2.3 Duality

The idea of duality in general is to provide an alternative formulation of a mathematical programming problem which is more convenient computationally or has some theoretical significance. The original problem is referred to as the *primal* and its reformulation will be known as the *dual*. We will focus on the *Wolfe dual* as proposed in [Wol61]. It replaces the constraint conditions by an optimality requirement on the Lagrangian function.

Theorem 3.2.13. *If $\mathbf{x}$ solves the convex programming primal problem* (3.2.22)*, if f and c_i are C^1 functions and if the regularity assumption* (3.2.13) *holds, then $\mathbf{x}, \boldsymbol{\alpha}$ solve the dual problem*

$$\max_{\mathbf{x},\boldsymbol{\alpha}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}), \text{ with } \mathcal{L} \text{ as in (3.2.18a)} \tag{3.2.23a}$$

$$\text{where } \nabla \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) = \mathbf{0}, \quad \boldsymbol{\alpha} \geq \mathbf{0}. \tag{3.2.23b}$$

Furthermore, the minimum primal and maximal dual function values are equal, that is,

$$f(\mathbf{x}) = \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}).$$

Proof. By Theorem 3.2.6 we know that there exist nonnegative multipliers $\boldsymbol{\alpha}$ such that $\nabla\mathcal{L}(\mathbf{x},\boldsymbol{\alpha}) = \mathbf{0}$ and $\alpha_i c_i(\mathbf{x}) = 0$ for every index i. It follows then

$$\mathcal{L}(\mathbf{x},\boldsymbol{\alpha}) = f(\mathbf{x}) - \sum_i \alpha_i c_i(\mathbf{x}) = f(\mathbf{x}).$$

We just need to see that these $\mathbf{x}$ and $\boldsymbol{\alpha}$ are effectively the solution of problem (3.2.23). Let $\mathbf{x}'$ and $\boldsymbol{\alpha}'$ be dual feasible. As each α_i' is nonnegative, $\mathcal{L}$ is convex (see Lemma 3.2.9) and $\nabla_{\mathbf{x}}\mathcal{L} = \mathbf{0}$, we have that

$$\begin{aligned}\mathcal{L}(\mathbf{x},\boldsymbol{\alpha}) = f(\mathbf{x}) \geq f(\mathbf{x}) - \sum \alpha_i' c(\mathbf{x}) &= \mathcal{L}(\mathbf{x},\boldsymbol{\alpha}') \\ &\geq \mathcal{L}(\mathbf{x}',\boldsymbol{\alpha}') + \langle \mathbf{x}-\mathbf{x}', \nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}',\boldsymbol{\alpha}')\rangle \\ &= \mathcal{L}(\mathbf{x}',\boldsymbol{\alpha}').\end{aligned}$$

Thus, $(\mathbf{x},\boldsymbol{\alpha})$ effectively is a global maximum. □

3.3 Linear SVM recast as a constrained optimization problem

Recall that the SVM problem considers vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Each vector has its label $y_1, y_2, \ldots, y_n$ from a set of possible labels $\{-1, +1\}$. The SVM problems asks for a partition of $\mathbb{R}^d$ via a hyperplane such that the vectors on one side of the hyperplane all have positive labels while the vectors on the other side of such hyperplane are all labeled negatively.

3.3.1 The separable case

As mentioned in Section 3.1, first consider the case where there are only two possible labels ($m = 2$) $\{l_1, l_2\} = \{-1, 1\}$. Assume further that both families of vectors can be split by a hyperplane $\mathbf{H}$ defined by a normal vector $\mathbf{w}$ and a scalar b. Then the linear SVM problem is recast as the constrained optimization problem (3.1.3).

$$\min_{(\mathbf{w},b)\in\mathbb{R}^d\times\mathbb{R}} f(\mathbf{w},b) := \frac{1}{2}\|\mathbf{w}\|^2, \tag{3.3.1a}$$

$$\text{where } c_i(\mathbf{w},b) := y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) \geq 1 \text{ for all } i = 1,\ldots,n. \tag{3.3.1b}$$

Notice that all the constraints (3.1.3b) are linear. Thus all the feasible directions are actually linear feasible directions and the regularity condition (3.2.13) holds. On the other hand, the domain of both f and c_i is the convex set $\mathbb{R}^{d+1}$. We state a general analysis lemma to see that f is a convex function.

Lemma 3.3.1. *If $g : \mathbb{R}^d \to \mathbb{R}$ is convex and $h : \mathbb{R} \to \mathbb{R}$ is convex and nondecreasing, then $h \circ g$ is convex.*

Proof. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\theta \in (0,1)$ we have first due to convexity of g that:

$$g(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta g(\mathbf{x}) + (1-\theta)g(\mathbf{y}).$$

As h is nondecreasing and convex, we conclude that

$$\begin{aligned} h(g(\theta\mathbf{x} + (1-\theta)\mathbf{y})) &\leq h(\theta g(\mathbf{x}) + (1-\theta)g(\mathbf{y})) \\ &\leq \theta h(g(\mathbf{x})) + (1-\theta)h(g(\mathbf{y})). \end{aligned}$$

□

Notice that $\|(1-\theta)\mathbf{w}_0 + \theta\mathbf{w}_1\| \leq (1-\theta)\|\mathbf{w}_0\| + \theta\|\mathbf{w}_1\|$ for any $\mathbf{w}_0\mathbf{w}_1 \in \mathbb{R}^d$ and $\theta \in (0,1)$. Hence $g(\mathbf{w}, b) := \|\mathbf{w}\|$ is convex. $h(x) := \frac{1}{2}x^2$ is a convex nondecreasing function. Thus due to the previous lemma, our objective function $f = h \circ g$ is convex as well. Due to linearity, we see that each c_i is concave. Hence the problem (3.1.3) is a convex programming problem.

Theorems 3.2.6 and 3.2.12 tell us then that Karush-Kuhn-Tucker conditions are necessary and sufficient for any minimum $(\mathbf{w}, b)$. The choice of the KKT technique is based on the fact that it will enable us to code most of our information in terms of inner products. That way we will be able to generalize easily to non-separable or nonlinear cases later. Theorem 3.2.13 also tell us that it is possible to solve the problem through its Wolfe dual (3.2.23).

Karush-Kuhn-Tucker conditions

The primal Lagrangian (as in (3.2.18a)) of the problem (3.1.3) is

$$\mathcal{L}_P(\mathbf{w}, b) := \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) + \sum_{i=1}^{n} \alpha_i. \tag{3.3.2}$$

The KKT conditions of the primal problem are then

$$\frac{\partial \mathcal{L}_P}{\partial w_j} = w_j - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_{i_j} = 0, \quad j = 1, \dots, d \tag{3.3.3a}$$

$$\frac{\partial \mathcal{L}_P}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.3.3b}$$

$$y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1 \geq 0, \quad i = 1, \dots, n \tag{3.3.3c}$$

$$\alpha_i \geq 0 \quad \forall\, i \tag{3.3.3d}$$

$$\alpha_i(y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1) = 0 \quad \forall\, i. \tag{3.3.3e}$$

An immediate application of this recast is that, while we might be able to determine the minimizing $\mathbf{w}$ through the training procedure, we will be able to determine b easily through the complementarity condition (3.3.3e). We will just need to take an index i for which $\alpha_i > 0$. If we now take equalities (3.3.3a) and (3.3.3b) then:

$$\begin{aligned}\sum_{i=1}^{n} \alpha_i y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) &= \sum_{i=1}^{n} \alpha_i y_i \left[\sum_{j=1}^{n} \alpha_j y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right] + b \sum_{i=1}^{n} \alpha_i y_i \\ &= \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle.\end{aligned}$$

Similarly,

$$\begin{aligned}\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle &= \left\langle \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \ , \ \sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j \right\rangle \\ &= \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle.\end{aligned}$$

Finally we substitute the previous equalities in $\mathcal{L}_P$. The formulation of the Wolfe dual problem then tells us that we must maximize $\mathcal{L}_P(\mathbf{w}, b)$ subject to constraints (3.3.3a), (3.3.3b) and (3.3.3d). That way the Wolfe dual may be stated as maximizing $\mathcal{L}_D$ subject to the previous constraints, where

$$\mathcal{L}_D(\mathbf{w}, b) := \sum_{1 \leq i \leq n} \alpha_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \tag{3.3.4}$$

Test phase

Once the problem (3.1.3) is solved, the hyperplane $\mathbf{H} = \{\mathbf{z} \in \mathbb{R}^d : \langle \mathbf{z}, \mathbf{w} \rangle + b = 0\})$ splits $\mathbb{R}^d$ in two sides. One side corresponds to those vectors labeled $y_i = -1$ and the other for those labeled $y_i = 1$. For any other vector x different from the training ones, its label will simply be according to which side of the hyperplane it lies in. In other words,

$$\text{label}(\mathbf{x}) = \text{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b\right) \tag{3.3.5}$$

where *sgn* is the sign function. The second equality is due to (3.3.3a). The function $t : \mathbb{R}^d \to \mathbb{R}$ defined as

$$t(\mathbf{x}) := \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{3.3.6}$$

will be referred as the *decision* or *test function* yielded by the SVM.

3.3.2 The non-separable case

Now assume $\mathbb{R}^d$ cannot be split by a hyperplane in such a way that it also splits the training vectors according to their label. The procedure described above will then find no solution. A way to handle the non-separable problem would be to relax the constraints (3.1.1a) and (3.1.1b) but only when it is strictly necessary. Nonnegative cost variables ξ_i will be associated to each training vector $\mathbf{x}_i$ to indicate when the above constraints need to be relaxed. The new constraints thus become

$$\langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 - \xi_i, \quad \text{if } y_i = 1 \tag{3.3.7a}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle \geq -1 + \xi_i, \quad \text{if } y_i = -1 \tag{3.3.7b}$$

$$\xi_i \geq 0 \quad \forall\, i. \tag{3.3.7c}$$

A classification error occurs whenever $\xi_i \geq 1$, that is, whenever the constraint is relaxed to allow a training vector on the other side of the hyperplane. Observe that the number of training errors is upper bounded by $\sum_{i=1}^{n} \xi_i$. The problem now asks not just to minimize $\|\mathbf{w}\|^2$ but to minimize the number of classification errors as well. We state

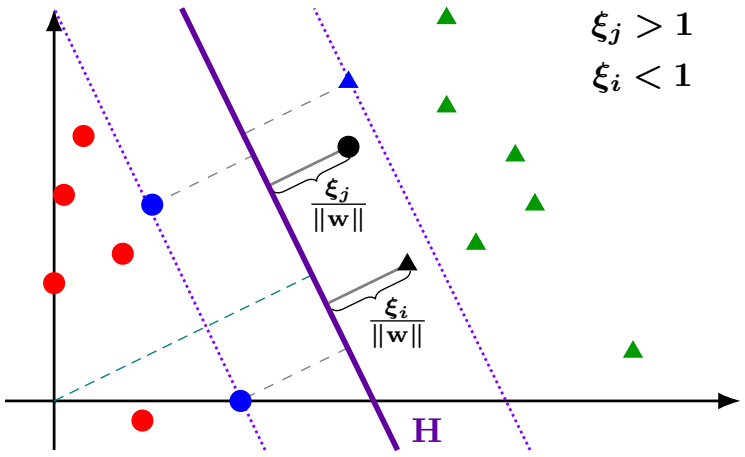


Figure 3.4: Non-separable case. Notice that the black circle presents a classification error

thus a new constrained optimization problem as

$$\min_{(\mathbf{w},b,\boldsymbol{\xi})\in\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d} f(\mathbf{w},b,\boldsymbol{\xi}) := \frac{\|\mathbf{w}\|^2}{2} + C\left(\sum_{i=0}^{n}\xi_i\right)^k, \qquad C\geq 0, k\geq 1 \tag{3.3.8a}$$

$$\text{where } c_i(\mathbf{w},b,\boldsymbol{\xi}) := y_i(\langle\mathbf{w},\mathbf{x}_i\rangle + b) + \xi_i \geq 1, \qquad \forall\, i=1,\dots,n \tag{3.3.8b}$$

$$\text{and } \xi_i \geq 0, \qquad \forall\, i=1,\dots,n. \tag{3.3.8c}$$

Here C, k are fixed values. Larger C's and k's mean a harsher penalty for every classification error. Notice that $g_i(\boldsymbol{\xi}) := \xi_i$ is a convex function. Thus by Lemma 3.2.9, $g := \sum_i g_i$ is a convex function as well. On the other hand, $h(x) := Cx^k$ is a convex and increasing real-valued function. Thus $h \circ g$ is a convex function by Lemma 3.3.1. Hence the objective function f is convex as well. Additionally, notice that each c_i is linear and hence each of them is a concave function. Finally, observe that the domain for all of them is $\mathbb{R}^{2d+1}$ which is a convex set. Therefore (3.3.8) is a convex optimization problem. Moreover, as each c_i is linear, the regularity condition 3.2.13 holds and thus KKT conditions (3.2.6) are sufficient and necessary for any minimums, which happen to be global.

To make computations easier take $k = 1$. First observe that the Lagrangian of the

primal problem is then

$$\mathcal{L}_P(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right)^1 - \sum_{i=0}^{n}\alpha_i y_i\langle\mathbf{w}, \mathbf{x}_i\rangle - b\sum_{i=0}^{n}\alpha_i - \sum_{i=0}^{n}\alpha_i\xi_i - \sum_{i=0}^{n}\mu_i\xi_i. \tag{3.3.9}$$

The Lagrange multipliers μ_i are introduced to reinforce the nonnegativity of the costs ξ_i. The KKT conditions are then stated as

$$\frac{\partial\mathcal{L}_P}{\partial w_j} = w_j - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_{i_j} = 0, \quad j = 1, \dots, d \tag{3.3.10a}$$

$$\frac{\partial\mathcal{L}_P}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{3.3.10b}$$

$$\frac{\partial\mathcal{L}_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad i = 1, \dots, n \tag{3.3.10c}$$

$$y_i(\langle\mathbf{w}, \mathbf{x}_i\rangle + b) + \xi_i - 1 \geq 0, \quad i = 1, \dots, n \tag{3.3.10d}$$

$$\xi_i \geq 0, \quad i = 1, \dots, n \tag{3.3.10e}$$

$$\alpha_i \geq 0, \quad i = 1, \dots, n \tag{3.3.10f}$$

$$\mu_i \geq 0, \quad i = 1, \dots, n \tag{3.3.10g}$$

$$\alpha_i[y_i(\langle\mathbf{w}, \mathbf{x}_i\rangle + b) + \xi_i - 1] = 0, \quad i = 1, \dots, n \tag{3.3.10h}$$

$$\mu_i\xi_i = 0, \quad i = 1, \dots, n. \tag{3.3.10i}$$

Observe that (3.3.10c), (3.3.10f) and (3.3.10g) are summarized as the constraints

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \tag{3.3.11}$$

Note that by combining both (3.3.11) and (3.3.10i), if it happens that $\alpha_i < C$, then $\xi_i = 0$. Then if a vector is taken such that $0 < \alpha_i < C$, b can be deduced from (3.3.10h). As (3.3.8) is a convex optimization problem, due to Theorem 3.2.13 it is equivalent to solve its Wolfe dual problem. Substituting (3.3.10c) in (3.3.9), we obtain

$$\mathcal{L}_P(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=0}^{n}\alpha_i y_i\langle\mathbf{w}, \mathbf{x}_i\rangle - b\sum_{i=0}^{n}\alpha_i.$$

We then substitute 3.3.10a and 3.3.10b and work exactly as in the separable case. Hence we obtain the dual Lagrangian $\mathcal{L}_D(\mathbf{w}, b, \boldsymbol{\xi}) := \sum_{1\leq i\leq n}\alpha_i - \frac{1}{2}\sum_{1\leq i,j\leq n}\alpha_i\alpha_j y_i y_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle$. We

thus state the Wolfe dual problem as

$$\text{maximize } \mathcal{L}_D(\mathbf{w}, b, \boldsymbol{\xi}), \tag{3.3.12a}$$

$$\text{subject to } 0 \leq \alpha_i \leq C \tag{3.3.12b}$$

$$\text{and } \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{3.3.12c}$$

From (3.3.10a), the solution is given by $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$. Observe that this optimization problem is almost identical to the separable case problem. The only difference is the fact that now the α_i's also have an upper bound.

3.4 Nonlinear SVM

So far it has been assumed that the training vectors can be split by a hyperplane with certain relaxations. However, this might not always be the case. Sometimes the training vectors might be better split by a nonlinear surface. The key idea behind the generalization to nonlinear surfaces lies on the fact that virtually all the previous computations hinge on inner products.

Suppose all the training vectors are mapped to a new Hilbert space $(\mathcal{H}, \langle , \rangle_{\mathcal{H}})$ via a map $\Phi : \mathbb{R}^d \to \mathcal{H}$. Observe that the KKT conditions encode the information of the optimization problems (3.1.3) and (3.3.8) mostly as inner products. Thus our main interest lies on the behavior of inner products $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}}$ rather that the behavior of Φ itself.

For the remainder of the section, assume that all our data points $\{\mathbf{x}_i\}_{i=1}^n$ are contained in a compact set $C \subset \mathbb{R}^d$. Define then a *kernel function* $K : C \times C \to \mathbb{R}$ with $(\mathbf{x}, \mathbf{y}) \mapsto \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}}$. Observe that with an explicit kernel function it is not necessary to know Φ explicitly.

If we then replace $\langle \mathbf{x}, \mathbf{y} \rangle$ by $K(\mathbf{x}, \mathbf{y})$ throughout all our computations and results above for the linear case, then we will produce a support vector machine which lives in a more abstract space $\mathcal{H}$. Observe that all the results on constrained optimization hold even on infinite dimensional spaces. Thus all the considerations follow as we are still doing a linear separation but in a different higher-dimensional space.

As the support vector machine lives now in $\mathcal{H}$, the optimal hyperplane defined by some normal vector $\mathbf{w}$ will live in $\mathcal{H}$ as well. However, just as in the d dimensional case,

the solution will come from the equivalent of (3.3.5) and (3.3.10a). That is,

$$\text{class}(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\right). \tag{3.4.1}$$

3.4.1 Examples and conditions of kernel functions

For $d = 2$, $C = [-1, 1]^2$ and $\mathcal{H} = \mathbb{R}^3$ with the usual inner product some explicit examples can be worked out. For instance, consider

$$\Phi_1(x_1, x_2) := \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}. \tag{3.4.2}$$

Thus $K_1(\mathbf{x}, \mathbf{y}) := \langle \Phi_1(x_1, x_2), \Phi_2(y_1, y_2)\rangle = x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 = \langle \mathbf{x}, \mathbf{y}\rangle^2$. The square C is deformed in $\mathbb{R}^3$ via Φ_1 as seen in Figure 3.5a. Observe that in reality, it is not necessary to know Φ_1 but only K_1. Note that different Φ's and $\mathcal{H}$'s can yield the same kernel function. For example consider a new deformation of $\mathbb{R}^2$ in $\mathbb{R}^3$ or $\mathbb{R}^4$ via

$$\Phi_2(x_1, x_2) := \frac{1}{\sqrt{2}} \begin{pmatrix} x_1^2 - x_2^2 \\ 2x_1x_2 \\ x_1^2 - x_2^2 \end{pmatrix}, \quad \Phi_3(x_1, x_2) := \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{pmatrix}. \tag{3.4.3}$$

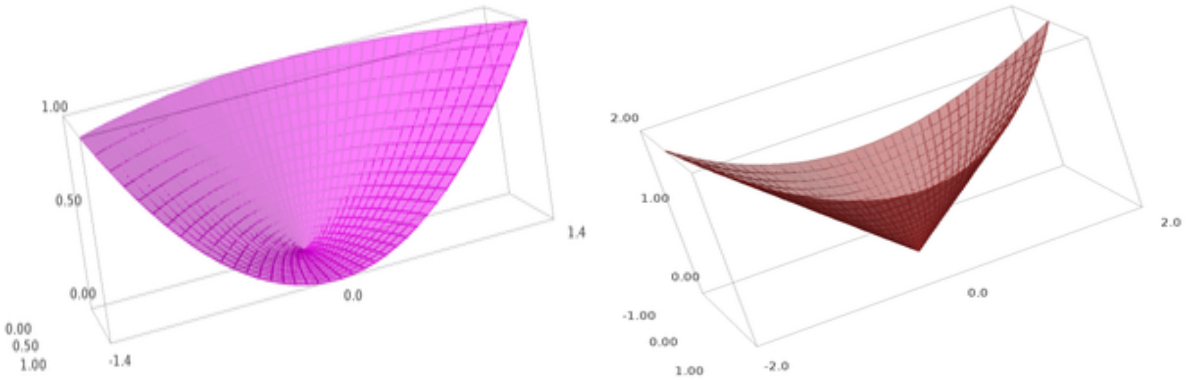
As stated before, the idea is to suggest a continuous real-valued function K and thus ignore computations involving Φ and $\mathcal{H}$. For a given continuous, real-valued function K on $C \times C$, an interesting question is to determine when does a Hilbert space $\mathcal{H}$ and a map Φ exist as described above. We might turn then to Mercer's condition. A proof of the following result can be seen [PR16, ch. 11].

Theorem 3.4.1 (Mercer's condition)**.** *For a compact subset $C \subset \mathbb{R}^d$ and given continuous function $K : C \times C \to \mathbb{R}$ there exists a mapping Φ, and a Hilbert space $\mathcal{H}$ such that*

$$K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle_{\mathcal{H}} \quad \forall\, \mathbf{x}, \mathbf{y} \in C \tag{3.4.4}$$

if and only if for any $L_2(C)$ function $g : C \to \mathbb{R}$ (that is, g^2 is Lebesgue-integrable on C) the following inequality holds

$$\int_C \int_C K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y})\, \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \geq 0. \tag{3.4.5}$$



(a) $C^2 \overset{\Phi_1}{\hookrightarrow} \mathbb{R}^3$ **(b)** $C^2 \overset{\Phi_2}{\hookrightarrow} \mathbb{R}^3$

Figure 3.5: Deformation of the $C = [-1,1]^2$ square in $\mathbb{R}^3$

As an quick application of Mercer's condition, we can establish that positive integral powers of the usual dot product are valid kernel functions. As an example, consider the following $K : C \times C \to \mathbb{R}$ defined as $K(\mathbf{x},\mathbf{y}) := \langle \mathbf{x},\mathbf{y}\rangle^m$ for some $m \in \mathbb{N}$. First consider a fixed $g \in L_2(\mathbb{R})$ and expand the expression $(\sum_i^d x_i y_i)^m$. Observe that each summand contributes to a term in the integral (3.4.5) of the form

$$\binom{m}{r_1, r_2, \cdots, (p - r_1 - r_2 - \cdots)} \int_C \int_C x_1^{r_1} x_2^{r_2} \cdots y_1^{r_1} y_2^{r_2} \cdots g(\mathbf{x}) g(\mathbf{y}) \, \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{y}.$$

Due to Fubini, the integral above is split in half which yields

$$\binom{m}{r_1, r_2, \cdots, (p - r_1 - r_2 - \cdots)} \left(\int_C x_1^{r_1} x_2^{r_2} \cdots g(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right) \left(\int_C y_1^{r_1} y_2^{r_2} \cdots g(\mathbf{y}) \, \mathrm{d}\mathbf{y} \right) =$$
$$\binom{m}{r_1, r_2, \cdots, (p - r_1 - r_2 - \cdots)} \left(\int_C x_1^{r_1} x_2^{r_2} \cdots g(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right)^2 \geq 0.$$

Thus K satisfies Mercer's condition and hence it is a valid kernel. Due to linearity of the integral, the same reasoning can be used to show that $K(\mathbf{x},\mathbf{y}) := \sum_{i=1}^p c_i \langle \mathbf{x},\mathbf{y}\rangle_i^m$ is also a valid kernel for nonnegative coefficients c_i and natural exponents m_i.

3.5 Multiclass SVMs

So far we have just discussed the case when only two possible labels can be assigned to the data. Suppose now that each of the n training data vectors $\mathbf{x}_i$ is labeled with exactly

one of the possible m labels $l_i \in \{1, \dots, m\}$. Similarly, test points $\mathbf{x}$ must be assigned exactly one of those m labels. The strategies to determine proper training and testing on the multiclass scenario are part of ongoing research on SVMs as discussed in [HL02]. We will discuss here only the two basic strategies to this problem: One-versus-All (OvA) and All-versus-All (AvA). According to [HL02], the AvA approach tends to provide better results than the OvA approach. For both strategies, we will reduce the multiclass SVM problem to the already solved binary SVM problem (3.1.3). Observe from the previous sections that we can also reduce the multiclass problem into a nonlinear, non-separable SVM problem instead.

3.5.1 One-versus-All (OvA)

The first strategy consists on solving m different SVMs with n points each, one per possible label. The j-th SVM will relabel the training data as $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $y_i = +1$ if $l_i = j$ and $y_i = -1$ otherwise. Then we solve the classic binary SVM problem (3.1.3). Thus we would compute an optimal hyperplane defined by $\mathbf{w}_j$ and b_j. This procedure would then produce m different test functions as in 3.3.6:

$$t_j(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w}_j \rangle + b_j.$$

Finally, we say for a new test point $\mathbf{x}$ that its label would be the one that maximizes its test function. That is

$$\text{label}(\mathbf{x}) = \arg \max_{1 \le j \le m} t_j(\mathbf{x}). \tag{3.5.1}$$

3.5.2 All-versus-All (AvA)

Also known as one-versus-one, it consists on solving $\binom{m}{2}$ different SVMs, one per possible pair of labels. The (j,k)-th SVM will relabel the training data as $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $y_i = +1$ if $l_i = j$, $y_i = -1$ if $l_i = k$ and $y_i = 0$ otherwise. Then we solve the SVM problem (3.1.3) for all the vectors with nonzero label. Although more SVMs ought to be computed, observe that on average, each of the SVMs will work with just $2/m$ points. This procedure will yield $\binom{m}{2}$ different separating hyperplanes defined by $\mathbf{w}_{j,k}$ and $b_{j,k}$ which

in turn define $\binom{m}{2}$ test functions as in (3.3.6),

$$t_{j,k}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w}_{j,k} \rangle + b_{j,k}.$$

There are several proposed test strategies involving all these test functions, a popular approach being the *voting max wins* strategy. That is, for each test item $\mathbf{x}$ compute $\operatorname{sgn}(t_{j,k}(\mathbf{x}))$. If the sign is positive, then the voting tally for label j increases one unit. Otherwise, the tally for label k is increased. After $\binom{m}{2}$ votes, label$(\mathbf{x})$ is defined as the label with the most votes. Observe that this approach might fail when two or more labels have the exact number of votes.

3.6 Discussion

By construction, the SVM is highly dependent on a good and large training set. Larger training sets allow a better placement of the splitting hyperplane. It is still a ongoing research question on how to determine if a given training set provides a meaningful test function. There is ongoing research as well regarding on how to determine the appropriate kernel function for a fixed dataset.

It is important to remember that SVMs work with a fixed number of different possible groups to assign. It may well be the case that a given test vector $\mathbf{z}$ is very distant from every training vector $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Thus SVM would rather classify $\mathbf{z}$ with label y_i, stating that the i-th training set S_i is the least distant to $\mathbf{z}$ rather that $\mathbf{z}$ being close to S_i. Hence the choice of y_i for $\mathbf{z}$ may be erroneous and a brand new label z would be a better choice. That said, the assortments obtained could be due to two reasons.

1. The data suggests that the given test point $\mathbf{x}$ is similar to a certain group and ought to assigned in such group.

2. The test point $\mathbf{x}_i$ is very distant from the rest of training data and the SVM picks the least distant group.

Part of the difficulties of the SVM is to determine which case occurred for each assignment. There is ongoing discussion in the literature on the best way to distinguish between those two cases above.

Chapter 4

Unsupervised SVMs

The Support Vector Machine problem described in the previous chapter is a supervised machine learning algorithm. This means that our data is divided in a training and a testing set. The training set is conformed by vectors $\mathbf{x}$ *and* their classifying labels y. From these labeled vectors, the algorithm finds the optimal splitting hyperplane which defines the classification for the vectors in the test set. This supervised approach, however, is unusable when there is no data for which the labels are known. In a simple unsupervised setting, we only have data and the assumption that it is divided between two or more possible families. Thus, the problem lies in finding the optimal labeling of the data, and from there on deduce a way to classify additional data.

Due to the lack of almost all *a priori* information, unsupervised methods tend to be more computationally demanding. This chapter reviews the idea of a unsupervised support vector machine. It is based on the ideas and algorithm presented in the 2012 paper by Karnin *et al.* [Kar+12]. We also discuss some slight improvements of our own to reduce the computational running time of the algorithm. We also present basic theory regarding singular values and singular value decomposition of matrices, which is strongly necessary to understand the algorithm and it was not presented in [Kar+12]. This algorithm is later used to produce some explicit results with archaeological data in Chapter 5 while [Kar+12] is limited to the theoretical implications of the algorithm.

We end the chapter with a brief discussion on several problems that may arise when the data lies in high dimensional spaces. We have a strong suspicion that effects of con-

centration of measure are especially present in the results obtained in Chapter 5 via this unsupervised learning algorithm.

4.1 The Furthest Hyperplane Problem

In the previous chapter, the basic SVM problem poses that we have a set of data vectors $\{\mathbf{x}_i\}_{i=1}^n$ in $\mathbb{R}^d$, each of them with a label $y_i \in \{-1, 1\}$. Our main interest is to find a splitting hyperplane $\mathbf{H}$ defined by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and a scalar $b \in \mathbb{R}$. If we assume that $\mathbf{w}$ is a unitary vector, then the constraint (3.1.2) is rewritten as

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \theta, \quad \text{for some fixed } \theta \text{ and } i = 1, \ldots, n. \tag{4.1.1}$$

Thus the optimization problem consists of maximizing the separation margin θ such that the constraints above all hold. Compare it to (3.1.3) in the previous section.

For the unsupervised setting, all the y_i's are unknown. One initial approach to a unsupervised SVM is the *Maximal Marginal Clustering (MMC)*. The method consists of considering all the possible labelings $\{y_i\}_{i=1}^n$, that is, all the vectors in $\mathbb{R}^d$ with entries in $\{-1, 1\}$, and compute then the standard supervised SVM for each of these possible labelings. Assuming we ignore the trivial labelings (where all the labels are identical), each labeling will yield a hyperplane $\mathbf{H}$ and a separation margin θ. The optimal labeling is thus the one that yields the largest possible margin θ. Although it is theoretically simple, it is computationally unfeasible. The MMC problem asks to solve the SVM optimization problem $(2^n - 2)/2 = 2^{n-1} - 1$ times which is computationally possible only for very small values of n. The MMC problem raises several questions, such as whether it is a convex programming problem or if there are ways to find the optimal labeling without going through all the possibilities.

One possible approach to simplify the computations is to make the assumption that the optimal hyperplane $\mathbf{H}$ goes through the origin. This is referred as the *Furthest Hyperplane Problem (FHP)*. In this case $b = 0$ and the optimization problem as in (4.1.1) is rewritten as

$$\text{maximize } \theta \tag{4.1.2a}$$

$$\text{subject to } \|\mathbf{w}\| = 1, \quad |\langle \mathbf{w}, \mathbf{x}_i \rangle| \geq \theta. \tag{4.1.2b}$$

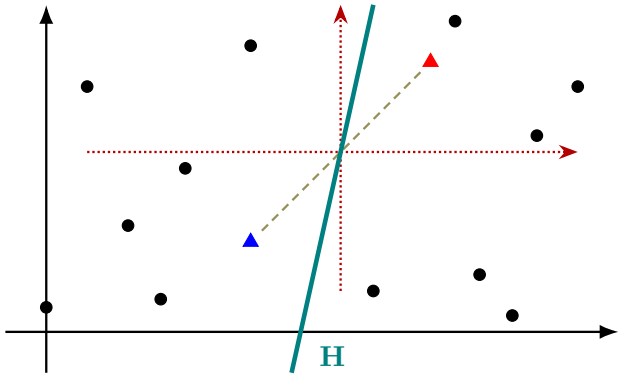


Figure 4.1: Translate the origin to the midpoint between two fixed points

The optimal labeling then will be the one given by $y_i := \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i \rangle)$. In order to make sense of the going-through-the-origin assumption, pick a pair of two different vectors $\mathbf{x}_i, \mathbf{x}_j$. We can assume that they belong to opposite families as we are discarding trivial labelings. Let $\mathbf{m}$ be the midpoint of the line segment between $\mathbf{x}_i$ and $\mathbf{x}_j$. Then the optimal hyperplane $\mathbf{H}$ must go through $\mathbf{m}$. If we translate every vector $\mathbf{x}_k$ by $-\mathbf{m}$, then $\mathbf{H}$ will effectively go through the new origin. Refer to Figure 4.1.

Thus in order to find the overall maximum margin θ, we must solve (4.1.2) for every possible pair of vectors. That implies that solving the MMP is equivalent to solving the FHP $\binom{n}{2}$ times. Hence the focus for the rest of the chapter will be on solving the FHP as fast as possible.

To make computations easier, we can assume for all the data vectors $\mathbf{x}_i$ that $\|\mathbf{x}_i\| \leq 1$ holds after they have been displaced by $-\mathbf{m}$. This assumption is valid as the optimal hyperplane won't be affected by scaling. Thus, due to Cauchy-Schwartz inequality and (4.1.2b), we have that $\theta \leq 1$, the maximum possible margin.

We will say that a labeling $\{y_i\}_{i=1}^n$ is *feasible* if there exists a unitary vector $\mathbf{w} \in \mathbb{R}^d$ such that $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0$ for every i. That is, the labeling yields a linear separable case as in the Figure 3.1 which has a solution as we know. Having said that, the *labeling margin* θ of a given feasible labeling is the solution of problem 4.1.2. Also, given a unitary vector $\mathbf{w}$ we will say that its *corresponding labeling* is the one produced by $y_i := \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i \rangle)$. We finally say a feasible labeling is *optimal* if it is the one that maximizes its labeling margin.

Computing the actual optimal labeling in the FHP setting is still a hard problem as the number of possible labels grows exponentially as the number of data vectors grow. An

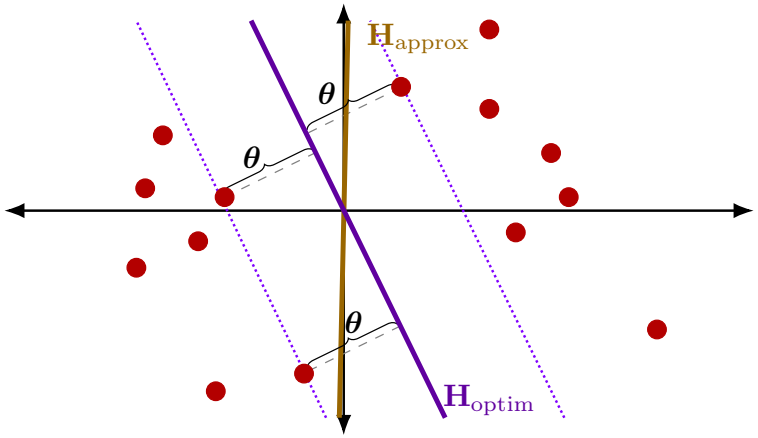


Figure 4.2: Approximate hyperplane vs. optimal hyperplane

approximation algorithm provides a faster computation with certain guarantees about being good enough. We might not compute the actual optimal splitting hyperplane $\mathbf{H}$, but we might compute an approximate hyperplane $\mathbf{H}'$ whose corresponding labelings coincide. For example, in Figure 4.2 both hyperplanes produce the same labeling of points.

However, in order to establish a good approximation algorithm we need to review some results on singular value decompositions. The following section follows Chapter 3 from [BHK17].

4.2 Singular Value Decompositions

We can think our initial data vectors $\mathbf{x}_i \in \mathbb{R}^d$ as the rows of a $n \times d$ matrix A. That is, the matrix entries are $(A)_{i,j} = x_{i,j}$. We would like to find the closest k-dimensional linear subspace V_k with respect to the set of data vectors. That is, we want to solve the optimization problem

$$\text{minimize } \sum_{i=1}^{n} d(\mathbf{x}_i, V_k)^2, \quad V_k \text{ is a } k\text{-dimensional subspace of } \mathbb{R}^d. \tag{4.2.1}$$

The above problem is known as the *best least square fit* problem. For the 1-dimensional case, we would like to find a line through the origin determined by a unit vector $\mathbf{v}$ such

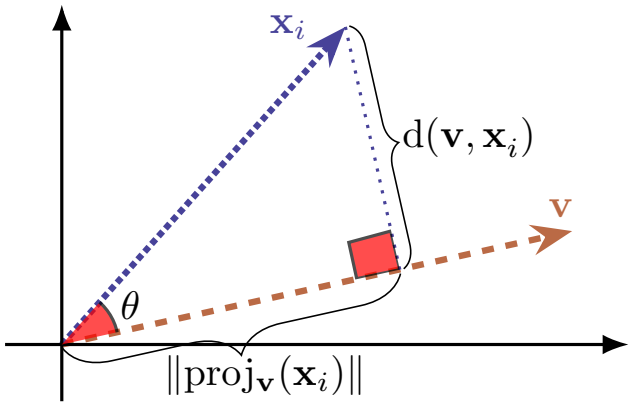


Figure 4.3: As $\|\mathbf{v}\| = 1$, observe that $\|\mathrm{proj}_v(x_i)\| = \|\mathbf{x}_i\| \cos\theta = |\langle \mathbf{v}, \mathbf{x}_i \rangle|$

that $\sum_{i=1}^n d(\mathbf{v}, \mathbf{x}_i)$ is minimized. As shown in Figure 4.3,

$$\|\mathbf{x}_i\|^2 = \|\mathrm{proj}_{\mathbf{v}}(\mathbf{x}_i)\|^2 + d(\mathbf{v}, \mathbf{x})^2,$$

from which

$$\|\mathbf{x}_i\|^2 - \|\mathrm{proj}_{\mathbf{v}}(\mathbf{x}_i)\|^2 = d(\mathbf{v}, \mathbf{x})^2.$$

Thus (4.2.1) for $k = 1$ is equivalent to maximize $\sum_{i=1}^n \|\mathrm{proj}_{\mathbf{v}}(\mathbf{x}_i)\|^2$.

4.2.1 Singular vectors

Consider a unit vector $\mathbf{v}$ and the line it defines through the origin. Observe that

$$\begin{aligned}\sum_{i=1}^n \|\mathrm{proj}_{\mathbf{v}}(\mathbf{x}_i)\|^2 &= \sum_{i=1}^n |\langle \mathbf{v}, \mathbf{x}_i \rangle|^2 \\ &= \sum_{i=1}^n \left[\sum_{j=1}^d x_{i,j} v_j \right]^2 = \sum_{i=1}^n (A\mathbf{v})_i^2 = \|A\mathbf{v}\|^2.\end{aligned}$$

Thus the best fit line is the one that maximizes $\|A\mathbf{v}\|^2$. We define the solution as the *first right singular vector* $\mathbf{v}_1$ of A as the solution to the optimization problem

$$\text{maximize } \|A\mathbf{v}\|, \quad \text{such that } \|\mathbf{v}\| = 1. \tag{4.2.2}$$

Observe that $\mathbf{v}_1$ always exists due to the continuity of $\|\cdot\|$ and the compacity of S^{d-1}. We define the *first singular value* of A as

$$\hat{\lambda}_1(A) := \max_{\|\mathbf{v}\|=1} \|A\mathbf{v}\|. \tag{4.2.3}$$

Notice that $\hat{\lambda}_1^2(A) = \sum_{i=1}^n \|\text{proj}_{\mathbf{v}_1}(\mathbf{x}_i)\|^2$. Now for the 2-dimensional case, consider a plane π that goes through the origin such that it contains $\mathbf{v}_1$. Consider a unit vector $\mathbf{v}_2$ perpendicular to $\mathbf{v}_1$. Due to Pythagoras,

$$\sum_{i=1}^n \|\text{proj}_{\pi}(\mathbf{x}_i)\|^2 = \sum_{i=1}^n \|\text{proj}_{\mathbf{v}_1}(\mathbf{x}_i)\|^2 + \sum_{i=1}^n \|\text{proj}_{\mathbf{v}_2}(\mathbf{x}_i)\|^2.$$

In order to maximize the left hand side above, we just need to look for a unit vector $\mathbf{v}_2$ perpendicular to $\mathbf{v}_1$ such that $\|A\mathbf{v}_2\|^2$ is maximized. As in the 1-dimensional case, such $\mathbf{v}_2$ exists due to continuity of $\|\cdot\|$ and compacity of S^{d-2}. We refer to $\mathbf{v}_2$ as the *second right singular vector* and to $\hat{\lambda}_2(A) := \|A\mathbf{v}_2\|$ as the *second singular value*. We construct analogously $\mathbf{v}_3, \mathbf{v}_4, \ldots, \mathbf{v}_r$ for $r \leq d$. The construction ends whenever

$$\max_{\mathbf{v} \perp \text{Span}\{\mathbf{v}_1,\ldots,\mathbf{v}_r\},\ \|\mathbf{v}\|=1} \|A\mathbf{v}\| = 0.$$

Observe that the algorithm ends with $\mathbf{v}_r$ if and only if $\text{rank}(A) = r$. We have to check that the linear subspaces $V_k := \text{Span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ for $1 \leq k \leq r$ effectively solve the best least square problem as in (4.2.1).

Theorem 4.2.1. *Let $A \in M_{n\times d}(\mathbb{R})$ with $\mathbf{v}_1, \ldots, \mathbf{v}_r$ and V_k as above. Then for each k, V_k is the best-fit k-dimensional subspace for A.*

Proof. The proof goes by induction on k. As discussed above, the theorem is valid for $k = 1$. For $k = 2$ let W be a best-fit 2-dimensional subspace for A. For any basis $\mathbf{w}_1, \mathbf{w}_2$ of unit vectors of W it is true that

$$\|A\mathbf{w}_1\|^2 + \|A\mathbf{w}_2\|^2 = \sum_{i=1}^n \|\text{proj}_{\mathbf{w}_1}(\mathbf{x}_i)\|^2 + \sum_{i=1}^n \|\text{proj}_{\mathbf{w}_2}(\mathbf{x}_i)\|^2 = \sum_{i=1}^n \|\text{proj}_{W}(\mathbf{x}_i)\|^2$$

Without loss of generality, we can assume that $\mathbf{w}_2 \perp \text{proj}_W(\mathbf{v}_1)$. Due to the definition of $\mathbf{v}_i$, it must be that $\|A\mathbf{w}_i\| \leq \|A\mathbf{v}_i\|$ for $i = 1, 2$. Combining both inequalities it yields

$$\sum_{i=1}^n \|\text{proj}_{W}(\mathbf{x}_i)\|^2 = \|A\mathbf{w}_1\|^2 + \|A\mathbf{w}_2\| \leq \|A\mathbf{v}_1\|^2 + \|A\mathbf{v}_2\|^2 = \sum_{i=1}^n \|\text{proj}_{V_2}(\mathbf{x}_i)\|^2.$$

In other words, V_2 is at least as good as W, so it must be a best-fit 2-dimensional subspace. The inductive step is analogous for k whenever the theorem holds for V_{k-1}. □

In general, we define the *Frobenius norm* $\|\cdot\|_F$ of a matrix as the square root of the sum of its squared entries. That is,

$$|A|_F^2 := \sum_{i=1}^{n}\sum_{j=1}^{r}(A)_{i,j}^2, \quad A \in M_{n\times d}(\mathbb{R}). \tag{4.2.4}$$

Lemma 4.2.2. *For any matrix A with exactly r singular values we have that $|A|_F^2 = \sum_{i=1}^{r}\hat{\lambda}_i^2(A)$.*

Proof. With A as always, it is a simple computation

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{r}x_{i,j}^2 = \sum_{i=1}^{n}\|\mathbf{x}_i\|^2 &= \sum_{i=1}^{n}\sum_{j=1}^{r}\|\text{proj}_{\mathbf{v}_j}(\mathbf{x}_i)\|^2 \\ &= \sum_{j=1}^{r}\sum_{i=1}^{n}\|\text{proj}_{\mathbf{v}_j}(\mathbf{x}_i)\|^2 = \sum_{j=1}^{r}\|A\mathbf{v}_j\|^2 = \sum_{j=1}^{r}\hat{\lambda}_j^2(A).\end{aligned}$$

□

That way, we can think of the singular values as the "components" of A. Also, for every d-dimensional vector $\mathbf{v}$, the n-dimensional vector $A\mathbf{v}$ is a linear combination of $A\mathbf{v}_1, \ldots, A\mathbf{v}_r$. We normalize these vectors and set

$$\mathbf{u}_i := \frac{1}{\hat{\lambda}_i(A)}A\mathbf{v}_i, \quad 1 \le i \le r. \tag{4.2.5}$$

These $\mathbf{u}_i$'s will be referred as *left singular vectors.* Just as with the right singular vectors, these left ones are orthogonal as well.

Theorem 4.2.3. *Let A be a rank r matrix. The left singular vectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$ are all orthogonal.*

Proof. The proof goes by induction on r. For $r = 1$ it is trivial. Now consider $\hat{\lambda}_1 := \hat{\lambda}_1(A)$ and the $n \times d$ matrix $B := A - \hat{\lambda}_1\mathbf{u}_1\mathbf{v}_1^\top$. Observe that $B\mathbf{v}_1 = 0$ and that for any $\mathbf{v}\perp\mathbf{v}_1$, we have $B\mathbf{v} = A\mathbf{v}$. Say $\mathbf{z}_1$ is the first right singular vector of B. If $\mathbf{z}_1$ had a nonzero component $\mathbf{z}_1'$ along $\mathbf{v}_1$, then $\|\mathbf{z}_1 - \mathbf{z}_1'\| < 1$ and

$$\left\|B\frac{\mathbf{z}_1 - \mathbf{z}_1'}{\|\mathbf{z}_1 - \mathbf{z}_1'\|}\right\| = \frac{\|B\mathbf{z}_1\|}{\|\mathbf{z}_1 - \mathbf{z}_1'\|} - \cancelto{0}{\frac{\|B\mathbf{z}_1'\|}{\|\mathbf{z}_1 - \mathbf{z}_1'\|}} > \|B\mathbf{z}_1\|.$$

This contradicts the maximality of $\mathbf{z}_1$ and it must be that $\mathbf{z}_1$ and $\mathbf{v}_1$ are orthogonal. Due to maximality, it must be that $\mathbf{z}_1 = \mathbf{v}_2$. With a similar reasoning we conclude that B has right singular vectors $\mathbf{v}_2, \ldots, \mathbf{v}_r$ and corresponding left singular vectors $\mathbf{u}_2, \ldots, \mathbf{u}_r$. As B has rank $r-1$, by induction hypothesis all $\mathbf{u}_2, \ldots, \mathbf{u}_r$ are orthogonal. All is left to do is to check orthogonality of $\mathbf{u}_1$.

Suppose this is false and without loss of generality, there is a $i \geq 2$ such that $\langle \mathbf{u}_1, \mathbf{u}_i \rangle > 0$. For a small $\epsilon > 0$,

$$\|\mathbf{v}_1 + \epsilon \mathbf{v}_i\|^2 = \langle \mathbf{v}_1 + \epsilon \mathbf{v}_i, \mathbf{v}_1 + \epsilon \mathbf{v}_i \rangle = \cancelto{1}{\|\mathbf{v}\|^2} + 2\epsilon \cancelto{0}{\langle \mathbf{v}_i, \mathbf{v}_1 \rangle} + \epsilon \cancelto{1}{\|\mathbf{v}_i\|^2}.$$

We compute then the d-dimensional vector

$$\mathbf{w} := A \frac{\mathbf{v}_1 + \epsilon \mathbf{v}_i}{\|\mathbf{v}_1 + \epsilon \mathbf{v}_i\|} = \frac{\hat{\lambda}_1 \mathbf{u}_1 + \epsilon \hat{\lambda}_1 \mathbf{u}_1}{\sqrt{1+\epsilon^2}},$$

whose norm must be at least as large as its component along $\mathbf{u}_1$. We approximate the denominator with its Taylor series and compute

$$\begin{aligned} \text{proj}_{\mathbf{u}_1}(\mathbf{w}) = \left\langle \mathbf{u}_1, \frac{\hat{\lambda}_1 \mathbf{u}_1 + \epsilon \hat{\lambda}_1 \mathbf{u}_1}{\sqrt{1+\epsilon^2}} \right\rangle &= (\hat{\lambda}_1 + \epsilon \hat{\lambda}_i \langle \mathbf{u}_i, \mathbf{u}_1 \rangle)(1 - \epsilon^2/2 + O(\epsilon^4)) \\ &= \hat{\lambda}_1 + \epsilon \hat{\lambda}_i \langle \mathbf{u}_i, \mathbf{u}_1 \rangle - O(\epsilon^2) > \hat{\lambda}_1. \end{aligned}$$

This contradicts the maximality of the first singular value $\hat{\lambda}_1$. Thus $\mathbf{u}_1, \ldots, \mathbf{u}_r$ are all orthogonal. □

4.2.2 SVD

Remember that in general two $n \times d$ matrices A, B are identical if and only if $A\mathbf{v} = B\mathbf{v}$ for every $v \in \mathbb{R}^n$.

Theorem 4.2.4. *Let A be an $n \times d$ matrix with right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$, left singular vectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$ and corresponding singular values $\hat{\lambda}_1, \ldots, \hat{\lambda}_r$. Then*

$$A = \sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top. \tag{4.2.6}$$

Proof. Due to orthonormality, for each $1 \leq j \leq r$ we have that $A\mathbf{v}_j = \sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top \mathbf{v}_j$. By construction, $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ forms a basis of V_r, a r-dimensional subspace of $\mathbb{R}^d$ which is also the image of A when seen as a linear transformation. Complete a basis for the whole $\mathbb{R}^d$ by considering additional vectors $\mathbf{w}_{r+1}, \ldots, \mathbf{w}_d$. We know that every $\mathbf{w}_j$ is perpendicular to every vector in V_r. Also by the rank-nullity theorem we know that $A\mathbf{w}_j = 0$. Thus for every vector $\mathbf{v} \in \mathbb{R}^d$ we have that

$$\begin{aligned}
\sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top \mathbf{v} &= \sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top \left[\sum_{j=1}^{r} \alpha_j \mathbf{v}_j + \sum_{j=r+1}^{d} \alpha_j \mathbf{w}_j \right] \\
&= \sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \sum_{j=1}^{r} \alpha_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle + \sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \sum_{j=r+1}^{r} \alpha_j \langle \mathbf{v}_i, \mathbf{w}_j \rangle^{0} \\
&= \sum_{j=1}^{r} \alpha_j \sum_{i=i}^{r} \hat{\lambda}_i \mathbf{u}_i \\
&= \sum_{j=1}^{r} \alpha_j A\mathbf{v}_j = A\mathbf{v}.
\end{aligned}$$

□

The decomposition shown in (4.2.6) is referred as the *singular value decomposition (SVD)* of A. In matrix notation it can also be expressed as $A = UDV^\top$ where the columns of U (resp. V) are the left (resp. right) singular vectors and D is diagonal with the singular values of A as entries. For any matrix A, by definition, its singular values area unique. If all the singular values are different, then the singular vectors are unique as well.

To compute the SVD it is quicker to compute the eigenvectors and eigenvalues of the $d \times d$ matrix $A^\top A$. This computation is cheaper than solving multiple optimization problems as posed in (4.2.2). First observe that $A^\top A$ is certainly of rank r. A quick computation reveals that due to orthonormality,

$$\begin{aligned}
A^\top A\mathbf{v}_k &= \left(\sum_{i=1}^{r} \hat{\lambda}_i \mathbf{v}_i \mathbf{u}_i^\top \right) \left(\sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top \right) \mathbf{v}_k \\
&= \left(\sum_{i=1}^{r} \hat{\lambda}_i \mathbf{v}_i \mathbf{u}_i^\top \right) \hat{\lambda}_k \mathbf{u}_k \\
&= \hat{\lambda}_k \left(\sum_{i=1}^{r} \hat{\lambda}_i \mathbf{v}_i \langle \mathbf{u}_i, \mathbf{u}_k \rangle \right) = \hat{\lambda}_k (\hat{\lambda}_k \mathbf{v}_k) = \hat{\lambda}_k^2 \mathbf{v}_k.
\end{aligned}$$

In other words, each right singular vector $\mathbf{v}_k$ is an eigenvector of $A^\top A$ with eigenvalue $\hat{\lambda}_k^2(A)$. From here, V and D can be immediately computed. Finally, U is simply computed as AVD^{-1}.

4.2.3 Best k-rank approximations

Say A is a $n \times d$ matrix of rank r with SVD $\sum_{i=1}^r \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top$. Then its (i,j)-th entry is of the form $(A)_{i,j} = \sum_{l=1}^r \hat{\lambda}_l u_{i,l} v_{j,l}$. For every $k \in \{1, \ldots, r\}$ let

$$A_k := \sum_{i=1}^k \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top. \tag{4.2.7}$$

The truncated sum of matrices would have entries of the form $(A_k)_{i,j} = \sum_{l=1}^k \hat{\lambda}_l u_{i,l} v_{j,l}$. These A_k are the best k-rank matrix approximations of A in the sense that for any other k-rank matrix, $|A - A_k| \leq |A - B|$ where $|\cdot|$ is either the 2-norm or the Frobenius norm of the matrix. Remember that the 2-norm of any matrix A is defined as

$$|A|_2^2 := \max_{\|\mathbf{v}\|=1} \|A\mathbf{v}\|^2. \tag{4.2.8}$$

We will first see that A_k is the best k-ranked approximation with the Frobenius norm via a quick lemma.

Lemma 4.2.5. *The rows of A_k are the projections of the rows of A onto the subspace $V_k =$ Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$.*

Proof. Let $\mathbf{a}$ be an arbitary row vector of A. Since the $\mathbf{v}_i$ form an orthonormal basis, we can rewrite the projections as

$$\text{proj}_{V_k}(\mathbf{a}) = \sum_{i=1}^k \|\text{proj}_{\mathbf{v}_i}(\mathbf{a})\| \mathbf{v}_i^\top = \sum_{i=1}^k \langle \mathbf{a}, \mathbf{v}_i \rangle \mathbf{v}_i^\top.$$

Then the matrix whose rows are the projections of the rows of A onto V_k is given by

$$\sum_{i=1}^k A\mathbf{v}_i \mathbf{v}_i^\top = \sum_{i=1}^k \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top = A_k$$

□

Theorem 4.2.6. *For any matrix B of rank at most k we have $|A - A_k|_F \leq |A - B|_F$.*

Proof. Let B be a matrix with rank at most k which minimizes the expression $|A-B|_F$ for every matrix whose rank is at most k. Suppose its row vectors are $\mathbf{b}_1, \ldots, \mathbf{b}_n$ and define $V := \text{Span}\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. Thus $\dim(V) \leq k$. Say B' is the matrix such that each of its rows is the projection of the corresponding row in A onto V. Observe that the rows of B' also span V. Thus the rank of B' is at most k. We compute then

$$\begin{aligned} |A - B'|_F^2 = \sum_{i=1}^{n} \|\mathbf{x}_i - \text{proj}_V(\mathbf{x}_i)\|^2 &= \sum_{i=1}^{n} d(\mathbf{x}_i, V)^2 \\ &\leq \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{b}_i\|^2 = |A - B|_F^2. \end{aligned}$$

Due to minimality of B, it must be $B = B'$. From Theorem 4.2.1 we know that V_k, the subspace spanned by the first k right singular vectors, is the best-fit k-dimensional subspace for A. As A_k minimizes the sum of squared distances of rows of A to any k-dim subspace,

$$|A - A_k|_F \leq |A - B|_F.$$

□

Finally, to see that A_k is the best k-rank approximation when working with the 2-norm we first need a simple computation lemma.

Lemma 4.2.7. $|A - A_k|_2^2 = \hat{\lambda}_{k+1}^2$

Proof. Let $A = \sum_{i=1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top$ be its SVD. Then $A - A_k = \sum_{i=k+1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top$. Let $\mathbf{v}$ be the first right singular vector of $A - A_k$, which may be rewritten as a linear combination of $\mathbf{v}_1, \ldots, \mathbf{v}_r$. Next

$$\begin{aligned} \|(A - A_k)\mathbf{v}\|^2 = \left\| \sum_{i=k+1}^{r} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top \sum_{j=1}^{r} \alpha_j \mathbf{v}_j \right\|^2 &= \left\| \sum_{i=k+1}^{r} \alpha_i \hat{\lambda}_i \mathbf{u}_i \right\|^2 \\ &= \left\langle \sum_{i=k+1}^{r} \alpha_i \hat{\lambda}_i \mathbf{u}_i \, , \sum_{i=k+1}^{r} \alpha_i \hat{\lambda}_i \mathbf{u}_i \right\rangle \\ &= \sum_{i=k+1}^{r} \alpha_i^2 \hat{\lambda}_i^2. \end{aligned}$$

By definition, $\mathbf{v}$ maximizes the left hand side above subject to $\|\mathbf{v}\|^2 = \sum_{i=1}^{r} \alpha_i^2 = 1$. As $\hat{\lambda}_1, \ldots, \hat{\lambda}_r$ is a decreasing sequence, it must be that $\alpha_{k+1} = 1$ while the rest of α_i's vanish. Therefore $|A - A_k|_2^2 = \hat{\lambda}_{k+1}^2$. □

Theorem 4.2.8. *For any matrix B of rank at most k we have $|A - A_k|_2 \leq |A - B|_2$.*

Proof. We may safely assume that $\text{rank}(A) < k$, otherwise there is nothing to prove. Suppose the theorem is false, so there exists a better 2-norm approximation B. From the lemma above, that is equivalent to $|A - B|_2 < \hat{\lambda}_{k+1}$. Due to the rank-nullity theorem, the nullity of B is at least $d - k$. Thus there must exist a unit vector $\mathbf{z}$ such that

$$\mathbf{z} \in \ker(B) \cap \text{Span}\,\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\} \setminus \{0\}.$$

Then we compute

$$\begin{aligned}
\|(A-B)\mathbf{z}\|^2 = \|A\mathbf{z}\|^2 = \left\| \sum_{i=1}^{n} \hat{\lambda}_i \mathbf{u}_i \mathbf{v}_i^\top \mathbf{z} \right\|^2 &= \left\| \sum_{i=1}^{k+1} \hat{\lambda}_i \mathbf{u}_i \langle \mathbf{v}_i, \mathbf{z} \rangle \right\|^2 \\
&= \left\langle \sum_{i=1}^{k+1} \hat{\lambda}_i \mathbf{u}_i \langle \mathbf{v}_i, \mathbf{z} \rangle, \sum_{i=1}^{k+1} \hat{\lambda}_i \mathbf{u}_i \langle \mathbf{v}_i, \mathbf{z} \rangle \right\rangle \\
&= \sum_{i \leq i,j \leq k+1} \hat{\lambda}_i \hat{\lambda}_j \langle \mathbf{v}_i, \mathbf{z} \rangle \langle \mathbf{v}_j, \mathbf{z} \rangle \, \langle \mathbf{u}_i, \mathbf{u}_j \rangle \\
&= \sum_{i=1}^{k+1} \hat{\lambda}_i^2 \langle \mathbf{v}_i, \mathbf{z} \rangle^2 \\
&\geq \hat{\lambda}_{k+1}^2 \sum_{i=1}^{k+1} \langle \mathbf{v}_i, \mathbf{z} \rangle^2 = \hat{\lambda}_{k+1}^2 \|\mathbf{z}\|^2 = \hat{\lambda}_{k+1}^2.
\end{aligned}$$

Due to definition of 2-norm, we also have that $|A-B|_2^2 \geq \|(A-B)\mathbf{z}\|^2$, which contradicts our initial assumption. □

4.3 FHP Approximation Algorithm

Instead of computing the optimal hyperplane, which is expensive, it is preferable to compute a good enough approximation of such hyperplane, which is computationally less demanding. To make the problem more precise, by good enough we refer to find a

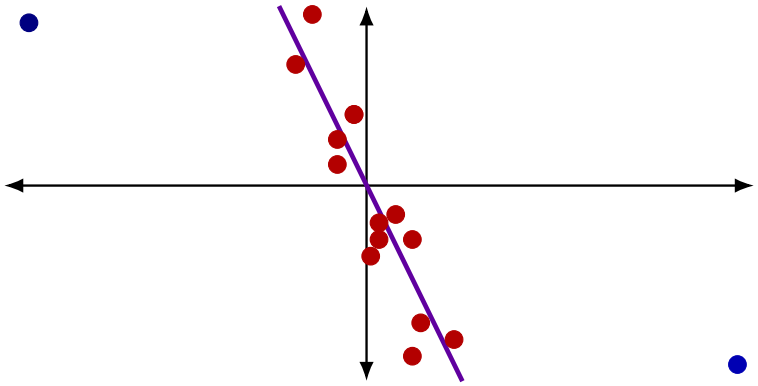


Figure 4.4: On average, every point is distant to the hyperplane

hyperplane which produces de same labeling as the optimal one as seen in Figure 4.2. More formally, for $0 < \alpha < 1$, the new problem will be to find a hyperplane $\mathbf{H}_\alpha$ such that for at least a $(1-\alpha)$-fraction of the vectors their distance to the actual optimal hyperplane $\mathbf{H}$ is at least $\alpha\theta$, where θ is the actual optimal margin. Consider first the easier problem of finding the hyperplane whose average margin is the largest, that is, to compute

$$\max_{\|\mathbf{w}\|=1} \text{mean}\{|\langle \mathbf{w}, \mathbf{x}_i \rangle|\}_{i=1}^{n}. \tag{4.3.1}$$

Say A is the matrix $n \times d$ that has the data vectors $\mathbf{x}_i$ as its rows. From the previous section, we know that the first right singular vector $\mathbf{w}$ of A maximizes $\sum_{i=1}^{n} \text{proj}_{\cdot}(\mathbf{x}_i)$ for all possible unit vectors. Thus the hyperplane with normal $\mathbf{w}$ maximizes the previous mean.

However, a hyperplane that *on average* is distant from all the points may not necessarily be the optimal choice. Some vectors may lie very close to the hyperplane whereas others may lie far apart from it, thus, on average every point is at a considerable distance from such hyperplane. Refer to Figure 4.4. Instead of computing the distance from each point to the hyperplane, we will compute weighted distances. Given an initial hyperplane $\mathbf{H}_i$, we may compute the distance from each point to $\mathbf{H}_i$ and the weight associated to the largest distances will be reduced and viceversa. With these new weights in place, we will compute a new hyperplane $\mathbf{H}_{i+1}$ which satisfies the new (4.3.1) problem. This procedure is iterated until weight penalties are small enough so that the average distance is actually a good representative for all the points. The approximation technique thus seeks to produce a series of hyperplanes with normal vectors $\mathbf{w}_1, \ldots, \mathbf{w}_t$ and later

combines them all using independent random Gaussian weights.

Assume first then that the optimal hyperplane goes through the midpoint $\mathbf{m}_{i,j}$ between two fixed vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. After translating the origin to $\mathbf{m}_{i,j}$ and rescaling the data vectors so all of them have norm less than 1, we can run the approximate FHP algorithm. A pseudocode is presented in Algorithm 2 for a fixed positive scalar c. The value of c will be discussed later. As it will be seen below, the while cycle in Algorithm 2 (lines 4-9) always terminates and the produced hyperplane normal $\mathbf{w}$ is distant from most of the input data vectors. Thus running the FHP approximate algorithm $\binom{n}{2}$ times will yield a solution $\mathbf{w}$ to the MMP.

Algorithm 2 Approximate FHP(i, j)

1: **Input:** $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d,\ \|\mathbf{x}_i\| \leq 1$
2: $\tau_i(i) \leftarrow 1,\ \forall\, i = 1, \ldots, n$ ▷ *Set initial weights for each vector*
3: $j \leftarrow 1$ ▷ *Number of solutions produced*
4: **while** $\sum_{i=1}^n \tau_j(i) \geq \frac{1}{n}$ **do**
5: $A_j \leftarrow n \times d$ matrix whose ith row is $\sqrt{\tau_j(i)}\mathbf{x}_i$ ▷ *Weighted distances*
6: $\mathbf{w}_j \leftarrow$ top right singular vector of A_j
7: $\sigma_i(j) \leftarrow |\langle \mathbf{x}_i, \mathbf{w}_j \rangle|$ ▷ *Compute distances*
8: $\tau_{j+1}(i) \leftarrow \tau_j(i)\, c^{-\sigma_j^2(i)}$ ▷ *Reweighting for a fixed scalar c*
9: $j \leftarrow j + 1$
10: $\mathbf{w}' \leftarrow \sum_{j=1}^t g_j \mathbf{w}_j$ for $g_j \sim \mathcal{N}(0, 1)$ ▷ *Independent Gaussian combination of solutions*
11: **Return:** $\mathbf{w} \leftarrow \mathbf{w}'/\|\mathbf{w}'\|$ ▷ *Output the normalization of $\mathbf{w}'$*

Lemma 4.3.1. *Algorithm 2 terminates after at most $t \leq 2\log(n)/(\theta^2(1 - \frac{1}{c}))$ iterations.*

Proof. For a fixed j, define $\tau_j := \sum_{i=1}^n \tau_j(i)$, the sum of weights. Say $\mathbf{w}^*$ is the optimal solution for the present FHP. Then $|\langle \mathbf{x}_i, \mathbf{w}^* \rangle| \geq \theta$ for every $i = 1, \ldots, n$. Observe that

$$\|A_j \mathbf{w}^*\|^2 = \sum_{k=1}^n \left\| \sum_{i=1}^d \sqrt{\tau_j(k)} x_{k,i} w_i^* \right\|^2 = \sum_{k=1}^n \tau_j(k) |\langle \mathbf{x}_k, \mathbf{w}^* \rangle|^2.$$

As $\mathbf{w}_j$ is the first right singular vector of A_j, we have that

$$\sum_{i=1}^n \sigma_j^2(i)\tau_j(i) = \|A_j \mathbf{w}_j\|^2 \geq \|A_j \mathbf{w}^*\|^2 = \sum_{i=1}^n \tau_j(i) \langle \mathbf{x}_i, \mathbf{w}^* \rangle^2 \geq \tau_j \theta^2.$$

Recall that for $c > 1$ and $0 \leq x \leq 1$ the inequality $c^{-x} \leq 1 - (1 - \frac{1}{c})x$ holds. With that in mind, we compute

$$\begin{aligned}\tau_{j+1} = \sum_{i=1}^{n} \tau_j(i) c^{-\sigma_j^2(i)} &\leq \sum_{i=1}^{n} \tau_j(i) \left[1 - \left(1 - \frac{1}{c}\right) \sigma_j^n(i)\right] \\ &= \sum_{i=1}^{n} \tau_j(i) - \left(1 - \frac{1}{c}\right) \sum_{i=1}^{n} \tau_j(i)\sigma_j^2(i) \\ &\leq \tau_j - \left(1 - \frac{1}{c}\right) \tau_j \theta^2 = \tau_j - \left(1 - \frac{1}{c}\right) \tau_j \theta^2 = \tau_j \left[1 - \left(1 - \frac{1}{c}\right) \theta^2\right].\end{aligned}$$

Since $\tau_1 = n$, we have that $\tau_{j+1} \leq n \left[1 - \left(1 - \frac{1}{c}\right) \theta^2\right]^j$. As $0 < \left[1 - \left(1 - \frac{1}{c}\right) \theta^2\right] < 1$, the while cycle in the FHP Algorithm is bound to terminate. Now, $j \geq 1$ is such that $n \left[1 - \left(1 - \frac{1}{c}\right) \theta^2\right]^j \geq \frac{1}{n}$ if and only if,

$$\begin{aligned} j \log \left[1 - \left(1 - \frac{1}{c}\right) \theta^2\right] &\geq -2 \log(n) \\ j \log \left[\frac{1}{1 - \left(1 - \frac{1}{c}\right) \theta^2}\right] &\leq 2 \log(n) \\ j \leq \frac{2 \log(n)}{\log \frac{1}{1 - \left(1 - \frac{1}{c}\right)\theta^2}} &\leq \frac{2 \log(n)}{(1 - \frac{1}{c})\theta^2}. \end{aligned}$$

For the last inequality recall that $\log(\frac{1}{1-x}) \geq x$ for every $0 \leq x \leq 1$. As t is the infimum of j's such that $\tau_j < \frac{1}{n}$, the conclusion follows. □

Theorem 4.3.2. *Let* $\sigma_i := \sqrt{\sum_{j=1}^{t} \sigma_j^2(i)}$. *When Algorithm 2 terminates, for each* i *it holds*

$$\sigma_i^2 \geq \frac{\log(n)}{\log(c)}. \tag{4.3.2}$$

Proof. For a fixed $i \in \{1, \ldots, n\}$ when the while ends we have that $\tau_t(i) \leq \tau_t < \frac{1}{n}$. As $\tau_1(i) = 1$ we see that

$$\frac{1}{n} \geq \tau_t(i) = \tau_{t-1} c^{-\sigma_{t-1}^2(i)} = \tau_1(i) \prod_{j=1}^{t} c^{-\sigma_j^2(i)} = 1 \cdot c^{\sum_{j=1}^{t} \sigma_j^2(i)}.$$

Taking logarithms from both sides we obtain

$$-\log(n) \geq -\sum_{j=1}^{t} \sigma_j^2(i)\log(c)$$

$$\text{from which } \sum_{j=1}^{t} \sigma_j^2(i) \geq \frac{\log(n)}{\log(c)}.$$

□

Finally, we guarantee that the output $\mathbf{w}$ of the FHP Algorithm is an approximation of the optimal $\mathbf{w}^*$.

Theorem 4.3.3. *Let* $0 < \alpha < 1$. *Algorithm 2 outputs a random* $\mathbf{w}$ *unit vector such that with probability at least* $\frac{1}{147}$ *at most a* 2α *fraction of the points are such that* $|\langle \mathbf{w}, \mathbf{x}_i \rangle| \leq \alpha\theta$.

Proof. Recall that $\mathbf{w}' = \sum_{j=1}^{t} g_j \mathbf{w}_j$ for $g_j \sim \mathcal{N}(0,1)$. Observe that for each of the components of $\mathbf{w}'$ is distributed as a normal random variable with mean 0 and variance $\sum_{j=1}^{t} w_{j,i}$. Then

$$\mathbb{E}(\|\mathbf{w}'\|^2) = \sum_{i=1}^{d} \mathbb{E}(w_i'^2) = \sum_{i=1}^{d}\sum_{j=1}^{t} w_{j,i}^2 = \sum_{j=1}^{t} \|\mathbf{w}_j\|^2 = t.$$

Then by Markov's inequality $\mathbb{P}(\|\mathbf{w}'\|^2 \geq \frac{7}{4}\sqrt{t}) \leq \mathbb{E}(\|\mathbf{w}\|^2)/[(\frac{7}{4})^2 t] = \frac{16}{49}$. Thus with probability at least $\frac{33}{49}$ we have that $\|\mathbf{w}'\|^2 \leq \frac{7}{4}\sqrt{t}$. We will assume this to be the case hereafter. Notice that we do not condition on this event happening but we rather accept the $\frac{16}{49}$ failure probability that we will later add in a union bound. Now we bound the probability that the final output fails at point i, that is, the hyperplane is too close to the i-th point. We then compute

$$\mathbb{P}(|\langle \mathbf{w}, \mathbf{x}_i \rangle| \leq \alpha\theta) = \mathbb{P}\left(\frac{1}{\|\mathbf{w}'\|}|\langle \mathbf{w}', \mathbf{x}_i \rangle| \leq \alpha\theta\right) = \mathbb{P}(|\langle \mathbf{w}', \mathbf{x}_i \rangle| \leq \|\mathbf{w}\|\alpha\theta)$$
$$\leq \mathbb{P}\left(|\langle \mathbf{w}', \mathbf{x}_i \rangle| \leq \frac{7}{4}\sqrt{t}\alpha\theta\right).$$

On the other hand, notice that

$$\langle \mathbf{w}', \mathbf{x}_i \rangle = \left\langle \mathbf{x}_i, \sum_{j=1}^{t} g_j \mathbf{w}_j \right\rangle = \sum_{j=1}^{t} g_j \langle \mathbf{x}_i, \mathbf{w}_j \rangle \sim \mathcal{N}\left(0, \sum_{j=1}^{t} \langle \mathbf{x}_i, \mathbf{w}_j \rangle^2\right).$$

In other words, $\langle \mathbf{x}_i, \mathbf{w}' \rangle$ distributes as a normal random variable with mean 0 and variance σ_i^2. Combining Lemma 4.3.2 we have that

$$\begin{aligned}\mathbb{P}(|\langle \mathbf{w}, \mathbf{x}_i \rangle| \leq \alpha\theta) &\leq \mathbb{P}\left(|Z| \leq \frac{7}{4}\sqrt{t}\alpha\theta\right), && Z \sim \mathcal{N}\left(0, \frac{\log(n)}{\log(c)}\right) \\ &= \mathbb{P}\left(|Z| \leq \frac{7}{4}\frac{\sqrt{\log(c)}\sqrt{t}\alpha\theta}{\sqrt{\log(n)}}\right), && Z \sim \mathcal{N}(0,1).\end{aligned}$$

The last equality is due to a simple change of variables. Recall the error function defined by

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-u^2}\,du = \frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n x^{2n+1}}{n!(2n+1)}. \tag{4.3.3}$$

In general for any normal random variable $X \sim \mathcal{N}(0, \sigma^2)$ with a change of variables and a small x we see that

$$\text{erf}\left(\frac{x}{\sigma\sqrt{2}}\right) = \frac{2}{\sqrt{2\pi}\sigma}\int_o^x e^{-\frac{u^2}{2\sigma^2}}\,du = \mathbb{P}(|X| \leq x) \leq \frac{2}{\sqrt{2\pi\sigma}}\,x.$$

With the last equality and Lemma 4.3.1, we conclude

$$\mathbb{P}(|\langle \mathbf{w}, \mathbf{x}_i \rangle| \leq \alpha\theta) \leq \frac{7}{2\sqrt{2\pi}}\frac{\sqrt{\log(c)}\sqrt{t}\alpha\theta}{\sqrt{\log(n)}} \leq \frac{7\sqrt{\log(c)}\alpha}{2\sqrt{\pi}(1-\frac{1}{c})}.$$

Finally, observe that due to l'Hôpital,

$$f(c) := \frac{7\sqrt{\log(c)}}{2\sqrt{\pi}(1-\frac{1}{c})} \xrightarrow[c\to 1]{} \frac{7}{2\sqrt{\pi}} \approx 1.97$$

and for any $1 < c \leq 1.05$, we will have that $f(c)\alpha < 2\alpha$ fraction of the points are close enough to our approximate hyperplane. Refer to Figure 4.5. □

4.4 MMP Aproximation

As discussed before, to solve the Maximum Margin Problem it suffices to solve the Furthest Hyperplane Problem at most $\binom{n}{2}$ times. A basic pseudocode is presented in Algorithm 3. When the Approximate FHP Algorithm 2 is used, a couple of small improvements can be implemented to run an approximate FHP less than $\binom{n}{2}$ times and to cap the number the iterations required by Algorithm 2.

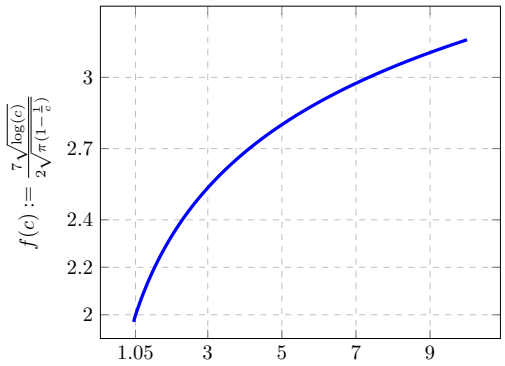


Figure 4.5: Plot of $f(c) := 7\sqrt{\log(c)}\alpha/2\sqrt{\pi}(1-\frac{1}{c})$

Algorithm 3 Approximate MMP

Input: $\{\mathbf{x}_k\}_{k=1}^n \subset \mathbb{R}^d,\ \|\mathbf{x}_i\| \leq 1$
for $1 \leq i < j \leq n$ **do** ▷ *For every possible pair of points*
  $\{\mathbf{z}_k\}_{k=1}^n \leftarrow \{\mathbf{x}_k\}_{k=1}^n$ ▷ *copy the data*
  $\mathbf{m}_{i,j} \leftarrow (\mathbf{z}_i + \mathbf{z}_j)/2$ ▷ *midpoint*
  $\{\mathbf{z}_k\}_{k=1}^n \leftarrow \{\mathbf{z}_k - \mathbf{m}_{i,j}\}_{k=1}^n$ ▷ *centers the data*
  $\{\mathbf{z}_k\}_{k=1}^n \leftarrow \{\mathbf{z}_k / \max_\ell \|\mathbf{z}_\ell\|\}_{k=1}^n$ ▷ *scales the centered data*
  $\mathbf{w}_{i,j} \leftarrow$ Approx FHP(i,j)
  $\theta_{i,j} \leftarrow \min_k \langle \mathbf{w}_{i,j}, \mathbf{z}_k \rangle$ ▷ *scaled margin*
  $\bar{\theta}_{i,j} \leftarrow (\max_k \|\mathbf{z}_k\|) \cdot \theta_{i,j}$ ▷ *unscaled margin*
$\bar{\Theta} = \bar{\theta}_{i',j'} \leftarrow \max_{1 \leq i < j \leq n} \bar{\theta}_{i,j}$ ▷ *optimal unscaled margin*
Return: $\mathbf{w} \leftarrow \mathbf{w}_{i',j'}$ ▷ *optimal hyperplane*

By FHP$_{i,j}$ we will refer to running Algorithm 2 where we assumed that the furthest hyperplane goes through the midpoint $\mathbf{m}_{i,j}$ between fixed data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. That is, during FHP$_{i,j}$ we translate all the data points by $-\mathbf{m}_{i,j}$ and rescale if necessary so every translated data vector has norm less than 1. We will assume that the approximate MMP is solved by running every FHP$_{i,j}$ in lexicographical order. That is, $(i,j) < (i',j')$ whenever $i < i'$ or $j < j'$ if $i = i'$. Assume that the FHP$_{i,j}$ run produces a normal vector $\mathbf{w}_{i,j}$ and a margin $\theta_{i,j}$. This value $\theta_{i,j}$ is most likely scaled down as it refers to data vectors of norm less than 1. We will refer to $\bar{\theta}_{i,j}$ as the margin unscaled for direct comparisons with other margins.

The improvements to MMP stem from keeping record of the *partially optimal margin* $\bar{\theta}_{i',j'}$, defined as the largest unscaled margin up to the (i',j')-th FHP. That is,

$$\bar{\Theta}_{i',j'} := \max_{(i,j)\leq(i',j')} \bar{\theta}_{i,j}. \tag{4.4.1}$$

Assume now that $(i,j) > (i',j')$ and that $\bar{\Theta}_{i',j'}$ is known. First observe that by definition in the (i,j)-th FHP, the data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ could be support vectors. Thus

$$\bar{\theta}_{i,j} \leq \|\mathbf{x}_i - \mathbf{m}_{i,j}\|. \tag{4.4.2}$$

If $\bar{\Theta}_{i',j'} > \|\mathbf{x}_i - \mathbf{m}_{i,j}\|$ then it will not be necessary to compute the approximate margin for the (i,j)-th case.

If the above inequality does not hold and $\text{FHP}_{i,j}$ proceeds, then the number of iterations in the while cycle can be capped due to Lemma 4.3.1 by

$$t \leq \tau_{i',j'} := \frac{2\log(n)}{{\Theta_{i',j'}}^2(1-\frac{1}{c})}. \tag{4.4.3}$$

Observe that the partial optimal margin has been scaled down accordingly to the (i,j) case. Also observe that larger partial optimal margins will produce smaller caps $\tau_{i',j'}$. Following the notation in Lemma 4.3.1, if $\tau_{\tau_{i',j'}} \geq \frac{1}{n}$ it is very unlikely that the produced margin $\bar{\theta}_{i,j}$ will surpass $\bar{\Theta}_{i',j'}$.

4.5 Discussion

The approximate FHP algorithm as presented above is limited in theoretical and computational aspects. It is unclear how to generalize the algorithm when 3 or more families are being considered. Contrary to the supervised case, our data here is not coded exclusively in inner products and it is unclear how to implement a kernel trick to consider approximate nonlinear splitting hyperplanes. Although distances are actively weighted when trying to find the best approximation, testing also suggests that Algorithm 2 gives best results when the data are clearly clustered in two distinct subsets. If both families are mingled, the optimal margin will be small, and as seen by Lemma 4.3.1, this would increase the number of iterations needed

Computationally the algorithm can be slow. The biggest toll comes from the repeated computation of the first right singular vector. Numerical methods such as the Power Method rather than the full computation of the singular value decomposition can improve the algorithm's performance. Capping the number of iterations or not even solving a particular FHP case using $\bar{\Theta}_{i,j}$ can improve the performance as well.

4.6 A word on the curse of dimensionality

Recall the Euler Characteristic Graph computation Algorithm 1 from Chapter 2. Given a filter function f and a fixed number T of thresholds, for a given simplicial complex K it would return a T-dimensional vector $\chi(K)$. As stated before, a larger number of thresholds would be more sensitive to changes in the Euler Characteristic and might offer a better description of the simplicial complex. However, this in turn would produce a descriptor vector in a higher-dimensional space. In high dimensions, most data points inside a finite volume tend to be far away from each other. The more dimensions we work with, the more data points are needed to fill the space for statistical and computational significance. This trade-off between detailed descriptions and statistical significance is commonly known as the *curse of dimensionality*. Depending on the number of data points available, high dimension could refer from 10 dimensions onwards. The following subsection is based on [Roj15] and [Spr14].

4.6.1 Overfitting and concentration

The first difficulty which arises when working in high-dimensional spaces is that the geometry of these spaces might be counterintuitive at first when compared to the familiar geometry in 2 and 3 dimensions. Take for example the d-dimensional hypercube $I^d := [0,1]^d$ and assume that our n data points $\{\mathbf{x}_i\}_{i=1}^d$ are contained in I^d. We know that the volume of this cube is 1. If we want to allocate that volume into n smaller cubes d-dimensional hypercubes (assume each hypercube contains exactly one data point) distributed homogeneously, then each of them must have volume $1/n$. That implies that

each of these smaller hypercubes must have side length

$$r = \left(\frac{1}{n}\right)^{\frac{1}{d}}. \tag{4.6.1}$$

For a number of n fixed then $r \to 1$ when $d \to \infty$. In other words, each of the new smaller hypercubes has roughly the same volume as the original I^d. In an infinite dimensional space, we can allocate n different hypercubes of volume 1 inside a hypercube of volume 1!

Similarly, consider the unit hypersphere S^{d-1} in $\mathbb{R}^d$. In general, the volume of a hypersphere of radius R is

$$\text{Vol}(S^{d-1}) = \frac{2\pi^{d/2}}{\Gamma(d/2)}R^d, \quad \text{where } \Gamma(t) = \int_0^\infty x^{t-1}e^{-x}\,dx, \quad t > 0. \tag{4.6.2}$$

We will usually write the volume as simply $k_d R^d$. If we want to cover the volume of S^{d-1} with n different, smaller hyperspheres of volume $k_d r^d$, then their radii must be

$$r = \left(\frac{k_d}{nk_d}\right)^{\frac{1}{d}} = \left(\frac{1}{n}\right)^{\frac{1}{d}}. \tag{4.6.3}$$

Just as in (4.6.1), $r \to 1$ whenever $d \to \infty$. Thus, the smaller hyperspheres are roughly the same size as S^{d-1}, although they just have $\frac{1}{n}$-th of S^{d-1}'s volume! These and other counterintuitive facts make difficult the visualization of data in higher dimensions, as our intuition from 2 and 3 dimensions starts falling apart.

Back to the hypercube, suppose we now keep r fixed so that each data point is contained in exactly one hypercube of side length r. Then from (4.6.1) we have that

$$n = r^{-d}, \quad 0 < r < 1. \tag{4.6.4}$$

This in turn means that we need an exponentially growing number of data points to keep the same data density as we increase the dimension. For instance, if $r = 0.1$, we would need $n = 10^d$ data points. In other words, when the number n of data points is fixed, the data tends to be more sparse whenever the dimension increases. If data is sparse, it is easier to find hyperplanes that maximize the margins which might result in overfitting. That is, the computed separating hyperplane is custom-built for our particular training dataset and might fail to hold when a wider dataset is considered.

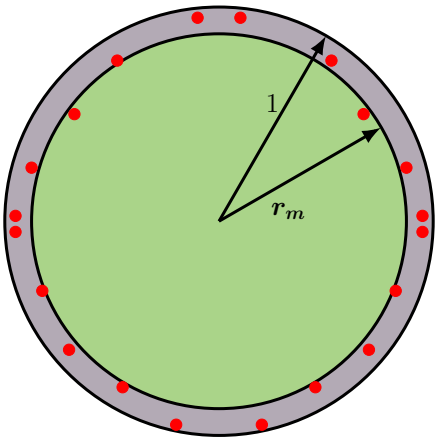


Figure 4.6: When uniformly distributed within S^{d-1}, the data clusters near the border

Finally, suppose that we spread n data points uniformly in $\mathbb{R}^d$. As there is a finite number of data points, it can be assumed after scaling that they lie within S^{d-1}. That is, $\|\mathbf{x}_i\| \leq 1$ for all $i = 1, \ldots, n$. Assume r_m is the median distance from the origin to its nearest neighbor in Euclidean distance. By definition of median, with probability $\frac{1}{2}$ all the data points lie between the hyperspheres of radii r_m and 1. If we equate it with the probability of uniformly picking n points between the hyperspheres of radii r_m and 1, we obtain

$$\frac{1}{2} = \left(\frac{k_d - k_d r_m^d}{k_d} \right)^n . \tag{4.6.5}$$

From the equation above, we obtain that the median distance is then

$$r_m = \left[1 - \left(\frac{1}{2} \right)^{\frac{1}{n}} \right]^{\frac{1}{d}} . \tag{4.6.6}$$

Observe that $r \to 1$ as $d \to \infty$. In other words, as the dimension increases, the data, if spread uniformly within S^{d-1} will tend to be concentrated around S^{d-1}.

For example, consider 128 simplicial complexes and compute the ECG vector $\chi(K)$ with $T = 32$ thresholds for each complex. Thus we are working in a 32-dimensional Euclidean space with only 128 data points. If the ECGs are scaled so their norm is less than 1 and if they spread uniformly, then $r_m \approx 0.85$. Thus, most of the ECGs would be clustered around the unit sphere. Moreover, if $T = 64$, then $r_m \approx 0.92$ and if $T = 128$, then $r_m \approx 0.96$.

4.6.2 A word on concentration of measure

The clustering trend as showed in (4.6.6) is part of an important sub-area of research withing probability known as *concentration of measure*. The clustering above can be informally interpreted as a concentration of measure around a thin shell surrounding S^{d-1}. Thus, on high dimensions, traditional methods such as the SVM might fail to differentiate two different families apart and might see all the data as one big cluster. The following material is based on the first two lectures in [Bar05].

It is a well known fact that

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-x^2/2} \, \mathrm{d}x = 1. \tag{4.6.7}$$

The equation above can be generalized to higher dimensions where the multidimensional integral can be split into several one-dimensional integrals as (4.6.7).

$$\begin{aligned}(2\pi)^{-n/2} \int_{\mathbb{R}^d} e^{-\|\mathbf{x}\|^2/2} \mathrm{d}\mathbf{x} &= (2\pi)^{-n/2} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} \prod_{i=1}^{n} e^{-x_i^2/2} \, \mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \prod_{i=1}^{d} \left[\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-x^2/2} \, \mathrm{d}x \right] \\ &= 1.\end{aligned}$$

Thus we define the *standard Gaussian measure* γ_d *in* $\mathbb{R}^d$ as the probability measure defined by the density function $(2\pi)^{-d/2} \exp(-\|\mathbf{x}\|^2/2)$. That way, for every Borel-measurable set $A \subset \mathbb{R}^d$ we have

$$\gamma_d(A) = \frac{1}{(2\pi)^{d/2}} \int_A e^{-\|\mathbf{x}\|^2/2} \, \mathrm{d}\mathbf{x}. \tag{4.6.8}$$

We will see that for most of the vectors $\mathbf{x} \in \mathbb{R}^d$ we have with respect to γ_d that $\|\mathbf{x}\|$ is approximately $\sqrt{d}$. Thus, from the standard Gaussian measure point of view, the whole $\mathbb{R}^d$ looks like a sphere of radius $\sqrt{d}$. To formalize this assertion we must first develop some basic notions on the *Laplace Transform* to solve inequalities.

Laplace Transform to solve inequalities

In general, let $(X, \mathcal{F}, \mu)$ be a probability space and let $f : X \to \mathbb{R}$ be a measurable function. If for some $x \in X$ and $a \in \mathbb{R}$ we have that $f(x) \geq a$, then for any $\lambda > 0$ we have that

$$e^{\lambda f(x)} \geq e^{\lambda a}. \tag{4.6.9}$$

As $e^{\lambda f}$ is a positive measurable function on X and $e^{\lambda a} > 0$, we have from (4.6.9) that

$$e^{-\lambda a} \int_X e^{\lambda f} \, \mathrm{d}\mu \geq e^{-\lambda a} \int_{\{x \,:\, f(x) \geq a\}} e^{\lambda f} \, \mathrm{d}\mu \geq \mu(\{x \,:\, f(x) \geq a\}). \tag{4.6.10}$$

Analogously we can see that

$$\mu(\{x \,:\, f(x) \leq a\}) \leq e^{\lambda a} \int_X e^{-\lambda f} \, \mathrm{d}\mu, \quad \lambda > 0. \tag{4.6.11}$$

This technique is particularly useful when $\int_X e^{-\lambda f} \, \mathrm{d}\mu$ can be computed explicitly. The parameter λ can be tuned accordingly.

Condensing the sphere from the Gaussian measure

Lemma 4.6.1. *For a fixed $d \in \mathbb{N}$ and γ_d the standard Gaussian measure in $\mathbb{R}^d$ we have that:*

1. *For any $\delta > 0$, the following inequality holds*

$$\gamma_d\{\mathbf{x} \in \mathbb{R} \,:\, \|\mathbf{x}\|^2 \geq d + \delta\} \geq \left(\frac{n}{n+\delta}\right)^{-d/2} e^{-\delta/2}. \tag{4.6.12}$$

2. *For any $0 \leq \delta < d$, the following inequality holds*

$$\gamma_d\{\mathbf{x} \in \mathbb{R} \,:\, \|\mathbf{x}\|^2 \geq d - \delta\} \geq \left(\frac{d}{d-\delta}\right)^{-d/2} e^{\delta/2}. \tag{4.6.13}$$

Proof. For the first part, choose a scalar $\lambda \in (0,1)$. Observe that if $\|\mathbf{x}\|^2 \geq d+\delta$, then from inequality (4.6.11) we see that

$$\begin{aligned}
\gamma_d\{\mathbf{x} \in \mathbb{R} \,:\, \|\mathbf{x}\|^2 \geq d + \delta\} &\geq e^{-\lambda(d+\delta)/2} \int_{\mathbb{R}^d} e^{\lambda\|\mathbf{x}\|^2} \, \mathrm{d}\gamma_n \\
&= e^{-\lambda(d+\delta)/2} (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{(\lambda-1)\|\mathbf{x}\|^2/2} \, \mathrm{d}\mathbf{x} \\
&= e^{-\lambda(d+\delta)/2} \left[\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{(\lambda-1)x^2/2} \, \mathrm{d}x\right]^d \\
&= e^{-\lambda(d+\delta)/2} (1-\lambda)^{-d/2}.
\end{aligned}$$

The last inequality is due to the Substitution Theorem. Finally, choose $\lambda = \delta/(d+\delta) > 0$ in the last equality and (4.6.12) follows. The second part of the lemma follows analogously. □

If for some $0 < \epsilon < 1$ we set $\delta = d\epsilon/(1-\epsilon)$ in (4.6.12), then $d + \delta = d/(1-\epsilon)$ and

$$\begin{aligned}\gamma_d\left(\left\{\mathbf{x} \in \mathbb{R} \,:\, \|\mathbf{x}\|^2 \geq \frac{d}{1-\epsilon}\right\}\right) &\leq (1-\epsilon)^{-d/2} \exp\left(-\frac{d\epsilon}{2(1-\epsilon)}\right) \\ &= \exp\left[-\frac{d}{2}\left(\frac{\epsilon}{1-\epsilon} + \log(1-\epsilon)\right)\right].\end{aligned}$$

Remember that both summands above can be expanded in absolutely convergent series,

$$\begin{aligned}\frac{\epsilon}{1-\epsilon} &= \epsilon + \epsilon^2 + \epsilon^3 + \cdots \\ \log(1-\epsilon) &= -\epsilon - \frac{\epsilon^2}{2} - \frac{\epsilon^3}{3} - \cdots\end{aligned}$$

From which we conclude that

$$\gamma_d\left(\left\{\mathbf{x} \in \mathbb{R} \,:\, \|\mathbf{x}\|^2 \geq \frac{d}{1-\epsilon}\right\}\right) \leq \exp\left(-\frac{d}{2}\cdot\frac{\epsilon^2}{2}\right) = \exp\left(-\frac{1}{4}d\epsilon^2\right), \quad 0 < \epsilon < 1. \tag{4.6.14}$$

Analogously, if we now set $\delta = n\epsilon$ in (4.6.13), then we obtain

$$\gamma_d\left(\left\{\mathbf{x} \in \mathbb{R} \,:\, \|\mathbf{x}\|^2 \leq d(1-\epsilon)\right\}\right) \leq \exp\left(-\frac{d}{2}\cdot\frac{\epsilon^2}{2}\right) = \exp\left(-\frac{1}{4}d\epsilon^2\right), \quad 0 < \epsilon < 1. \tag{4.6.15}$$

Finally, suppose $\{\rho_d\}_{d\in\mathbb{N}}$ is a nonnegative sequence such that $\rho_d \to \infty$ slowly (that is, $\rho_d = o(d)$ so the inequalities below hold) when $d \to \infty$. Then for a sufficiently large d and $\epsilon = \rho_d/\sqrt{d}$ we have that

$$\frac{d}{1-\epsilon} = \frac{d\sqrt{d}}{\sqrt{d}-\rho_d} \leq (\sqrt{d}+\rho_d)^2, \tag{4.6.16}$$

since $0 \leq d - (\sqrt{d} + \rho_d^2)$ and therefore $d\sqrt{d} \leq d\sqrt{d} + d\rho_d - \sqrt{d}\rho_d - \rho_d^3$. We may also realize that

$$d(1-\epsilon) = d - \rho_d\sqrt{d} = \sqrt{d}(\sqrt{d}-\rho_d) \geq (\sqrt{d}-\rho_d)^2 \tag{4.6.17}$$

as $\sqrt{d} \geq \sqrt{d} - \rho_d$. From the two inequalities above we have that

$$(\sqrt{d}-\rho_d)^2 \leq d - \rho_d\sqrt{d} \leq d \leq \frac{d\sqrt{d}}{\sqrt{d}-\rho_d} \leq (\sqrt{d}+\rho_d)^2. \tag{4.6.18}$$

At the same time, by substituting $\epsilon_d = \rho_d/\sqrt{d}$ in (4.6.14) and (4.6.15) with, (4.6.21) inequalities in mind, we obtain

$$\gamma_d\left(\left\{\mathbf{x}\in\mathbb{R}^d \,:\, \|\mathbf{x}\|^2 \geq (\sqrt{d}+\rho_d)^2\right\}\right) \leq \exp\left(-\frac{\rho_d^2}{4}\right) \tag{4.6.19}$$

$$\gamma_d\left(\left\{\mathbf{x}\in\mathbb{R}^d \,:\, \|\mathbf{x}\|^2 \leq (\sqrt{d}-\rho_d)^2\right\}\right) \leq \exp\left(-\frac{\rho_d^2}{4}\right) \tag{4.6.20}$$

respectively. If we combine these last two equations, we realize that

$$\gamma_d\left(\left\{\mathbf{x}\in\mathbb{R}^d \,:\, \sqrt{d}-\rho_d \leq \|\mathbf{x}\| \leq \sqrt{d}+\rho_d\right\}\right) \to 1 \quad \text{as} \quad d\to\infty. \tag{4.6.21}$$

In other words, as we stated earlier, the data in $\mathbb{R}^d$ tends to concentrate around the sphere of radius $\sqrt{d}$ for large d values. For example, suppose $\rho_d := \log(d)$. As in the Subsection 4.6.1, assume we are working with 128 ECG vectors, each of them with $T = 32$ dimensions. Then $\sqrt{128} \approx 11.3$ and

$$\gamma_{32}\left(\left\{\mathbf{x}\in\mathbb{R}^{32} \,:\, \left|\, \|\mathbf{x}\| - \sqrt{32} \,\right| \geq 3.5\right\}\right) \leq 0.1.$$

If $T = 64$, then

$$\gamma_{64}\left(\left\{\mathbf{x}\in\mathbb{R}^{64} \,:\, \left|\, \|\mathbf{x}\| - \sqrt{64} \,\right| \geq 4.2\right\}\right) \leq 0.03.$$

And if $T = 128$, then

$$\gamma_{128}\left(\left\{\mathbf{x}\in\mathbb{R}^{128} \,:\, \left|\, \|\mathbf{x}\| - \sqrt{128} \,\right| \geq 4.9\right\}\right) \leq 0.006.$$

Chapter 5

Archaeological results and discussion

We are now finally able to merge the tools presented in all the previous chapters. Remember that for a given simplicial complex K, its topology can tell us valuable information regarding its shape. The Euler characteristic $\chi(K)$ encodes part of that information. From Chapter 1 we know that the Euler characteristic $\chi(K)$ is a topological invariant. If we remove simplices from the complex, we would inevitably alter its topology and thus its Euler characteristic. The Euler Characteristic Graph (ECG) in Chapter 2 records the change of the Euler characteristic of the complex as we remove simplices in T steps according to some fixed filter function. Remember that the ECG then can be thought as a vector in $\mathbb{R}^T$.

Assume now that we have a collection of complexes $K_1 \ldots, K_n$ with corresponding ECGs $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We then classify these complexes according to their morphology into m different families. If we now consider a new complex K with ECG $\mathbf{x}$, we may try to classify it according to its shape with a supervised SVM procedure as discussed in Chapter 3. Moreover, we might not be sure of the labels y_i, disregard them altogether and attempt an unsupervised SVM clustering as discussed in 4

In this chapter we present classification results of archaeological data employing the procedure described above. We first discuss the origins of the data and the important questions behind it. Later, we present the explicit filter functions and parameters used to compute the ECGs. We also describe the parameters used to compute the SVMs, both in the supervised and unsupervised case. The chapter consists mainly of several clas-

sification results, which are accompanied by renderings of the objects and their corresponding ECGs. Conclusions and future work are found in the last part of the chapter.

5.1 Background and motivation

The results presented here are part of a much larger project involving archaeology and computer science as described by Jiménez in [JRM12]. The idea for the project led by Jiménez came from the need to classify similarities in 162 stone masks found in the remains of the Sacred Precinct of Tenochtitlán, the main ceremonial Aztec complex, located in Mexico City. The excavations were carried out between 1978 and 1982. The masks found in the temple present very distinct features that distinguish them from the usual Aztec styles. These masks, along other artifacts, were located in several Aztec offerings dating from the Late Postclassic period (1340 A.D. — 1521 A.D.), especially between 1390 and 1469 A.D. yet their appearance suggests they might come from the Teotihuacán region and from the South Pacific-coast state of Guerrero. Particularly, most of them seem to come from the Mezcala region which is more than 300km away from Tenochtitlán.

The geographical origin of these artifacts is not surprising as the Aztecs had a vast network for importing distant goods either as trade or tribute from other civilizations. However, the possible temporal origin of the offerings is more difficult to explain, as the Mezcala styles present at Tenochtitlán resemble styles found during Classic (200 A.D. — 1000 A.D.) and even Preclassic (2000 B.C. — 200 A.D.) times in the Guerrero region which raises the question on how the Aztecs acquired such ancient styles. There is an ongoing discussion as to whether the Aztecs collected “antique” artifacts, or if the Preclassic and Classic styles survived over time and were still reproduced in Postclassic settings.

The discussion faces a second difficulty, as there is no clear agreement on how many different Mezcala styles exist, which implies difficulties on knowing how many different groups of people were involved in the Aztec offerings. Some scholars believe there are at least five different styles [Cov61; OG86], whereas others recognize only four distinct styles [Gay67] and another group of experts put the number of different styles as low as two [Ser73]. As noted by Jiménez in [JRM12], “the diversity of views is due in part to a

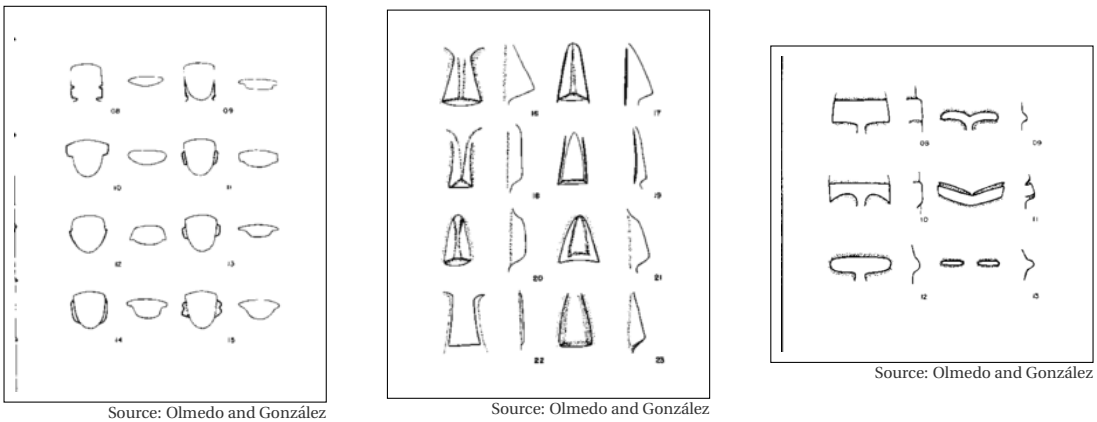


Figure 5.1: Numerical taxonomy on shape of masks' contour, nose and eyebrows.

lack of contextual information available for the majority of artifacts found in Guerrero, but it also reflects the subjective criteria used by researches to classify such artifacts." He also raises awareness of the need of more objective methods to give a better answer to several questions such as the number of actually different Mezcala styles, how many of those where contemporary to the Aztecs, how many of those styles were present at a given time, and how many and which of them are present in the 162 mask collection from Tenochtitlán. In 1986, Olmedo and González in [OG86] proposed a classification based on a numerical taxonomy. That is, the form of the masks' noses, heads, chins, etc. were codified categorically as exemplified by Figure 5.1. This method yielded a vector of 23 different numbers for each of the 162 masks, which were later grouped in 40 different sets.

Their results were nonetheless inconclusive, as 26 of the sets contained less than 3 masks each, and 20 masks couldn't be properly included in any of the 40 proposed groups. Later in 2011, Jiménez *et al.* [JRM12] developed a new classification based on spectral clustering [Lux07], an unsupervised learning technique. As described in [JRM12], spectral clustering seeks to identify groups by analyzing not the exact location of the points —like k-means techniques— but the connectedness between them. For instance, in the clusters shown in Figure 5.2 it is desirable that the points $\mathbf{p}_1$ and $\mathbf{p}_2$ are assigned to the same cluster. Observe that they are distant to each other, however, there are several data points in between them.

The spectral clustering technique grouped the masks into 8 principal sets. These

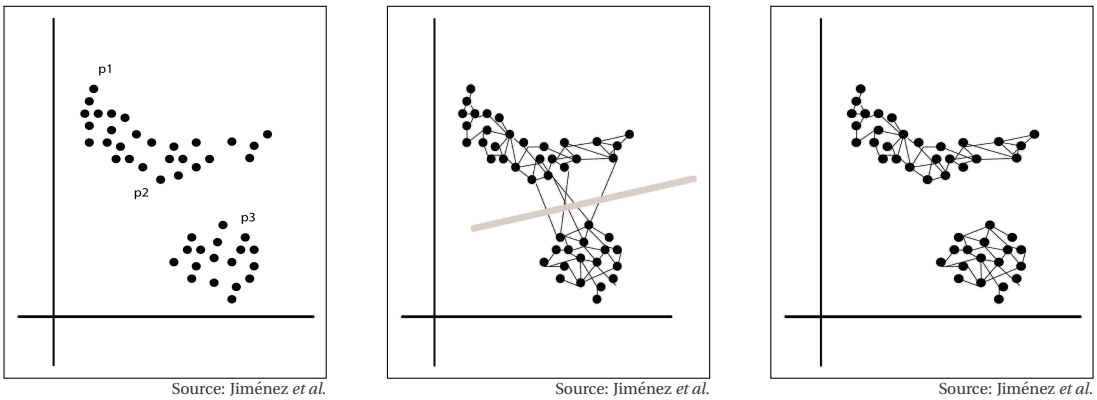


Figure 5.2: Example of spectral clustering

sets were later used as our ground truth to train our supervised SVMs and compare our unsupervised SVMs.

5.2 Archaeological data

The dataset analyzed here consists of 128 digitized 3D triangulated meshes of the masks which are part of the 162 masks found at the Sacred Precinct of Tenochtitlán. The 34 remaining masks were not considered as they are heavily damaged, incomplete, or possibly unfinished. The data was provided by Diego Jiménez and the National Institute of Anthropology and History (INAH). Each mesh was given as a `.stl` file. These files were later processed by Rogelio Hasimoto and Mario Canul from the Computer Science Department at the Mathematics Research Center (CIMAT). After their processing, each mesh was embedded in the $[-1,1]^3$ cube with its center of mass at the origin. Each mask would also face directly the $z = 1$ plane.

The meshes of the masks were originally grouped in 9 different families numbered from `02` to `10`. The first 8 sets were yielded by the spectral clustering algorithm stated above. Each of them consists of masks with clear visual resemblance within each set. The ninth group, Set `10`, is the largest one, and there is no visual resemblance among its items. The first 8 sets are believed to be different according to Jiménez whereas the 9th set is made of essentially those masks where there is no clear agreement within the archaeological community regarding their origin or style. Our main interest lies in finding

where the items from 10 are put, as there is no general consensus within the archaeological community regarding their origins and classification.

It is important to remark the scarcity of data for most of the families. Five out of the eight identified families have less than 6 items each. Specifically, the number of items per family is as follows.

SET	NO. OF ITEMS	SET	NO. OF ITEMS
02	24	07	4
03	6	08	3
04	4	09	7
05	19	10	59
06	2	**TOTAL**	128

The low number of items is problematic when trying to use classification methods such as the supervised and unsupervised support vector machines as discussed in chapters 3 and 4. The physical appearance of individual masks from each family can be seen in the Appendix A.

Another problem observed with the data is that the scans do not reflect the actual Euler Characteristic of the masks. Some of the masks without holes, which are topologically equivalent to a sphere, had an Euler Characteristic different from 2, contrary to what was expected. The reason for this is unknown and most likely took place when the physical masks were scanned. We made no attempt to correct these errors. However, as the results below will show, the ECGs and the following supervised SVMs still managed to produce sensible results.

5.3 Euler Characteristic Graphs

As discussed in Chapter 2, the ECG of a simplicial complex K is determined by a filtering function $f_0 : V_0 \to [a, b]$ and a set of T threshold values $t_0, t_1, \ldots, t_{T-1}$. Remember that V_0 is the set of vertices of the complex K. In this chapter, K will refer to the mesh of a mask as discussed above. As each mesh is embedded in $\mathbb{R}^3$, each vertex $\mathbf{v}$ can be thought as its Cartesian coordinates (x, y, z). Observe that K is a 2-dimensional complex, so its Euler

Characteristic at threshold t is then

$$\chi_i(K) = V_0^{(i)} - V_1^{(i)} + V_2^{(i)}. \tag{5.3.1}$$

As discussed at the end of Subsection 2.2, the ECGs used throughout this work were based on filter functions of the form

$$f(\mathbf{v}) = M - g(\mathbf{v}). \tag{5.3.2}$$

Remember that M and g are chosen such that the range of such f can be taken as $[0, M]$. There were 7 possible g filter functions considered along various possible M scalar values. Each of the filter functions was run with 32, 64 and 128 threshold values. The filter functions considered as g in (5.3.2) were:

planar sweeps	*cylindrical sweeps*	*spherical sweeps*
$\pi^{(1)}(\mathbf{v}) := x$	$g^{(1)}(\mathbf{v}) := x^2 + y^2$	$h^{(1)}(\mathbf{v}) := x^2 + y^2 + z^2$
$\pi^{(2)}(\mathbf{v}) := y$	$g^{(2)}(\mathbf{v}) := y^2 + z^2$	
$\pi^{(3)}(\mathbf{v}) := z$	$g^{(3)}(\mathbf{v}) := z^2 + x^2$	

These filter functions were concatenated in three different composite ECG filter functions as discussed in 2.2.7. An example of them can be seen in figures 5.3 and 5.4. Explicitly, using the notation from Subsection 2.2.7, the ECGs considered were yielded from the following filter functions:

$$\textit{planar sweeps} \quad \pi := (M_\pi - \pi^{(1)}) \oplus (M_\pi - \pi^{(2)}) \oplus (M_\pi - \pi^{(3)}); \tag{5.3.3}$$

$$\textit{cylindrical sweeps} \quad g := (M_g - g^{(1)}) \oplus (M_g - g^{(2)}) \oplus (M_g - g^{(3)}); \tag{5.3.4}$$

$$\textit{spherical sweeps} \quad h := M_h - h^{(1)}. \tag{5.3.5}$$

As the meshes were embedded in a $[-1, 1]^3$ cube, bounds for which 5.3.2 holds are $(M_\pi, M_g, M_h) = (2, 2, 3)$. Close study of the particular 128 meshes suggested tighter bounds, especially for M_g and M_h. The bounds considered for each possible filter function were as follows:

M_π	M_g	M_h
2	2	2
	$\sqrt{2}$	$\sqrt{2}$
	1	1

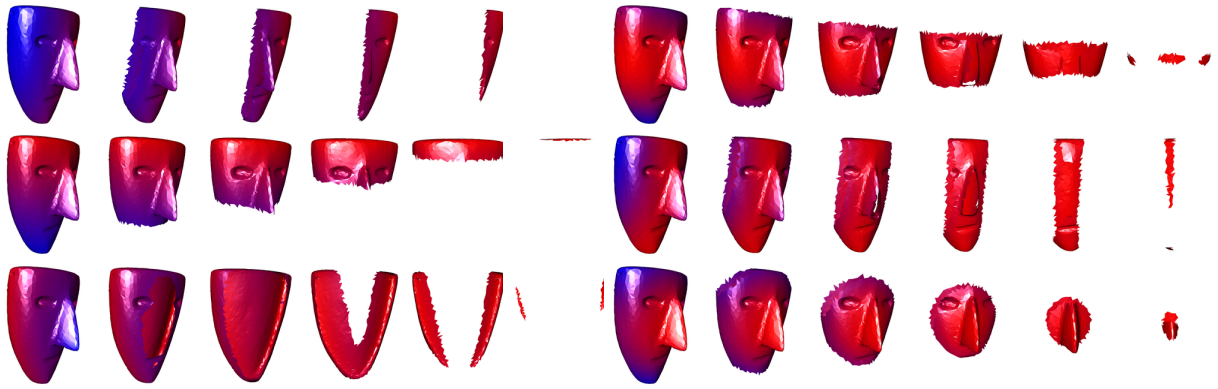


Figure 5.3: Destroying the mask 38 with planar sweeps π on the left and with cylindrical sweeps g on the right.

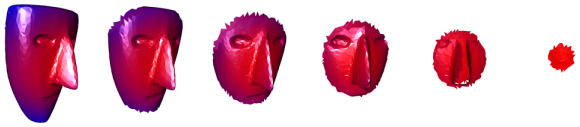


Figure 5.4: Destroying the mask 38 with spherical sweeps h on the right.

Additionally, filter functions based on local curvature around each vertex were considered. However, these gave poor results in both supervised and unsupervised SVM classification.

Auxiliary files were created for each mask. These files gave each vertex Cartesian coordinates, which vertices were contained in each edge and which vertices contained in each triangle. These auxiliary files were made employing the *Point Cloud Library* in C [RC11]. The ECGs were computed using these auxiliary files and later saved in `csv` files. The ECGs were later plotted using R base tools [R C18].

As mentioned in 4.6, it is important to keep in mind possible concentration of measure effects. Remember that at high dimensions, the vectors tend to cluster around a sphere as stated in (4.6.21). For instance, when running either filter function π or g with 128 thresholds, we obtained a 384-dimensional vector for each mask. The effects of concentration of measure were seen as with higher numbers of threholds T the SVMs concluded that the data tended form a unique large cluster.

5.4 Supervised Support Vector Machines

As stated in Chapter 3, the performance of any support vector machine depends on a training subset from the data to analyze. Due to their low number of items, the sets `03,04,06,07,08` and `09` are used as a whole for training. Certain items from sets `02` and `05` are used as well. From set `02` we excluded masks 16 and 23 (refer to Figure A.1) as they do not have holes in their eyes as the rest of masks within this set. We also excluded masks 03, 09, 18, 20 and 21 since, as pointed out by Jiménez, the archaeological community is still discussing their possible origins. From set `05`, masks 50, 51, and 53 (refer to Figure A.4) were not taken as part of the training data, as Jiménez points out that further discussion is needed to assert their inclusion in this set. Mask 48 was not considered either, as Jiménez strongly believes that it was misclassified. None of the masks from set `10` were considered for training, either. Later, the 128 masks are used as test data. The SVM aims to classify the 128 masks into one of the 8 possible categories defined by sets `02`,. . .,`09`.

The supervised SVMs were computed using the library *e1071* available through R [Mey+17]. The SVM follows the code of *LIBSVM* library based in C/C++ [CL11]. *LIBSVM* uses an All-vs-All with a max wins strategy approach to do multiclass classification as discussed in Subsection 3.5.2. Whenever different labels obtained the same amount of votes, the algorithm simply picks the smallest of those labels. *LIBSVM* allows the use of linear, polynomial, radial and sigmoid kernels with several parameters and costs as follows:

$$\text{Linear: } \langle \mathbf{x}, \mathbf{y} \rangle; \tag{5.4.1}$$

$$\text{Polynomial: } (\gamma \langle \mathbf{x}, \mathbf{y} \rangle + k)^{\delta}; \tag{5.4.2}$$

$$\text{Radial: } \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2); \tag{5.4.3}$$

$$\text{Sigmoid: } \tanh(\gamma \langle \mathbf{x}, \mathbf{y} \rangle + k). \tag{5.4.4}$$

Only polynomial kernels were considered for the results as the number of possible input parameters was crucial for the classification methodology. Radial and sigmoid kernels didn't yield sensible results. Most of the SVMs were carried out with a fixed cost $C = 10$.

5.5 Unsupervised Support Vector Machines

The Furthest Hyperplane Problem and Maximum Margin Problem algorithms described in Chapter 4 (Algorithm 2) were implemented in C/C++. The singular values and overall matrix object operations were computed with the *GSL Library* [Gal+17]. As it was pointed out in the Subsection 4.5, it is unclear how to generalize the algorithm to nonlinear, nonseparable cases. It is also unclear how to handle multiple labelling.

The methodology consisted of evaluating ECGs from two of the eight different training sets. It was expected to determine the dividing hyperplane along the support vectors that define each training set. As the computation of singular values and vectors is quite expensive, the dimension of the ECGs was reduced to 25 dimensions using *Principal Component Analysis (PCA)*, cf. [Shl14]. The PCA procedure also minimized the efects of concentration of measure as mentioned in the Subsection 4.6. The PCA was implemented via computation of Singular Value Decomposition with the *GSL Library*.

5.6 Computation of algorithms

All the algorithms and procedures above where computed in a Personal Computer with the following specifications

```
Processor: 4x Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
VGA controller: Intel Corporation Sky Lake Integrated Graphics
memory size: 7892MiB
memory description: SODIMM DDR3 Synchronous 1600 MHz (0.6 ns)
```

The computation of the ECGs was quite efficient. The computation and saving of the ECGs for the 128 masks took less than 5 seconds overall per filtration. Thus the Algorithm 1 proves to be quite efficient even when each mesh has 60,000 vertices on average. The ECGs were saved in a `.csv` file which was later read and analyzed with the supervised and unsupervised procedures as described above. The supervised SVM was quite efficient as well as *LIBSVM* could compute 72 different SVM trainings and testings in less than 5 seconds. The unsupervised SVM was more computationally expensive, even when taking into account dimension reduction via PCA.

5.7 Results with nonlinear SVMs

As stated above, the main interest lies in assigning a family to all the items in Set `10`, as well to the excluded items from Sets `02` and `05`. However, it is not clear which assignment is the correct one and hence, it is unclear which kernel and which parameters (if applicable) are the correct ones. The strategy consisted of setting finite sets of possible values for parameters δ, γ, k and evaluate a different SVM for every possible combination of such parameters. Say N different combination of parameters are possible.

For every possible of combination above, every mask's ECG would be assigned to one of the possible 8 sets. Thus every mask is later associated to a N-dimensional vector $\mathbf{v}_\chi$, one entry per SVM test. Finally, for each of these N-dimensional vectors, the frequency of the groups is tallied and the mode of these groups is taken.

The reasoning is that, for a mask whose ECGs is not close enough to a particular training cluster, its assignment to such cluster will vary when the SVM dividing hyperplane's parameters are perturbed. Thus we propose the mode of $\mathbf{v}_\chi$ as the best possible assignment. The record of frequencies also tells us if a particular ECG is between two or more families. For instance, if the mode of $\mathbf{v}_\chi$ represents less than 85% of the vector's entries, it would stand to reason that further testing is necessary.

For the polynomial kernel case, the following results were obtained by considering all 72 possible combinations of parameters

$$\delta \in \{3, 5\}, \quad \gamma \in \{0.001\} \cup \{0.05, 0.15, \ldots, 1.05\}, \quad k \in \{-1, 0, 1\}.$$

5.7.1 Planar sweeps

It is worth mentioning that every item used for training was assigned to its corresponding group during the test phase. Thus the planar sweeps with polynomial kernel SVMs at least can distinguish the different training items.

(a) 73 **(b)** 76 **(c)** 87 **(d)** 89 **(e)** 94 **(f)** 103 **(g)** 125

Figure 5.5: Masks assigned to Set 02 after running 72 polynomial SVMs

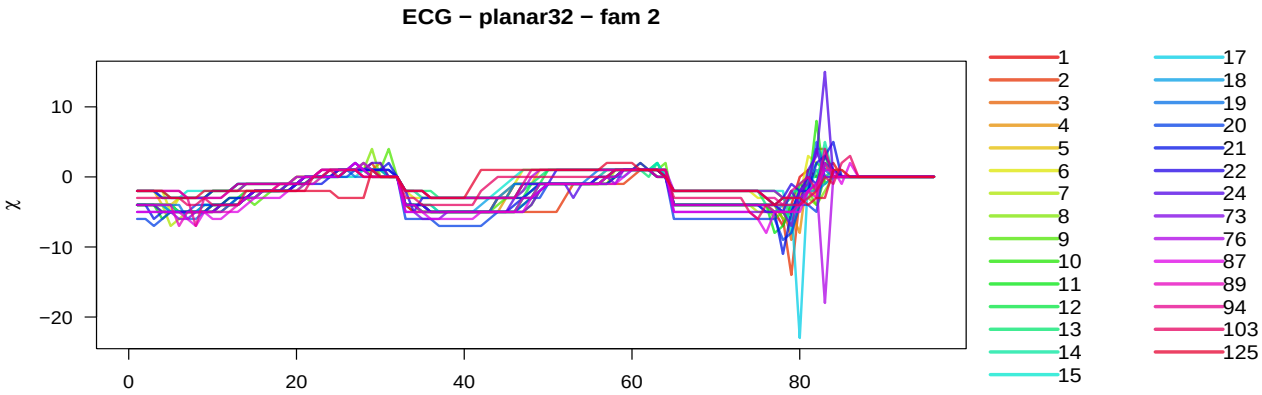
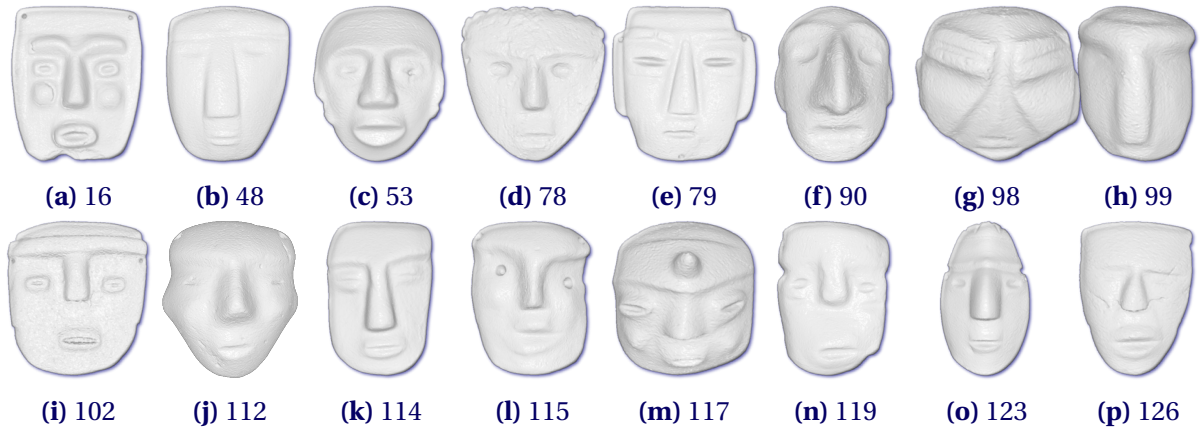


Figure 5.6: ECGs of newly assigned Set 02 based on planar sweeps f and $T = 32$

Using $T = 32$ thresholds

When considering a polynomial kernel, following the procedure described above, the Set 02 was assigned 7 new items from Set 10. 2 items, masks 16 and 23, from the original 02 failed to be assigned to this set. As discussed above, this masks were the only ones in 02 with no holes for either their eyes or mouth. Also, 6 of the 7 new items have holes as well. These can be seen in Figure 5.5. The ECGs seen in Figure 5.6 follow roughly the same pattern. As discussed in the Appendix A, the ECGs corresponding to masks 17 and 24 have large peaks near the end. Mask 73 has a trough near the end as well.



(a) 16 **(b)** 48 **(c)** 53 **(d)** 78 **(e)** 79 **(f)** 90 **(g)** 98 **(h)** 99

(i) 102 **(j)** 112 **(k)** 114 **(l)** 115 **(m)** 117 **(n)** 119 **(o)** 123 **(p)** 126

Figure 5.7: Masks assigned to Set 03 after running 72 polynomial SVMs

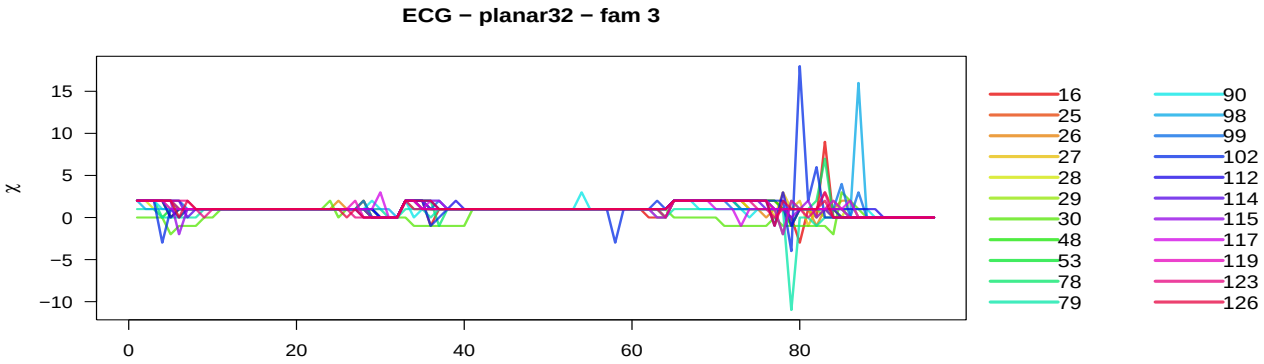


Figure 5.8: ECGs of newly assigned Set 03 based on planar sweeps f and $T = 32$

There were 13 masks from Set 10 that were assigned to 03. Mask 16 and masks 48 and 50, which were originally assigned to Sets 02 and 05 respectively, were excluded from the training set as discussed above. These three masks were now assigned to 03. The ECGs in Figure 5.8 reveal that masks 98 and 102 stand out with crests near the end. Mask 79 stands out as well with a trough near the end. All the assigned masks can be seen in Figure 5.7. Notice that they share certain physical characteristics with the originals from Set 03, such as elongated shapes (16, 48, 79, 90, 99, 114, 115, 119, 123, 126), defined and visible ears (53, 79, 90, 98, 112) and a clear T-shape formed by their nose and eyebrows (48, 53, 78, 79, 115, 119, 123, 126).

(a) 50 **(b)** 111

Figure 5.9: Masks assigned to Set 04 after running 72 polynomial SVMs

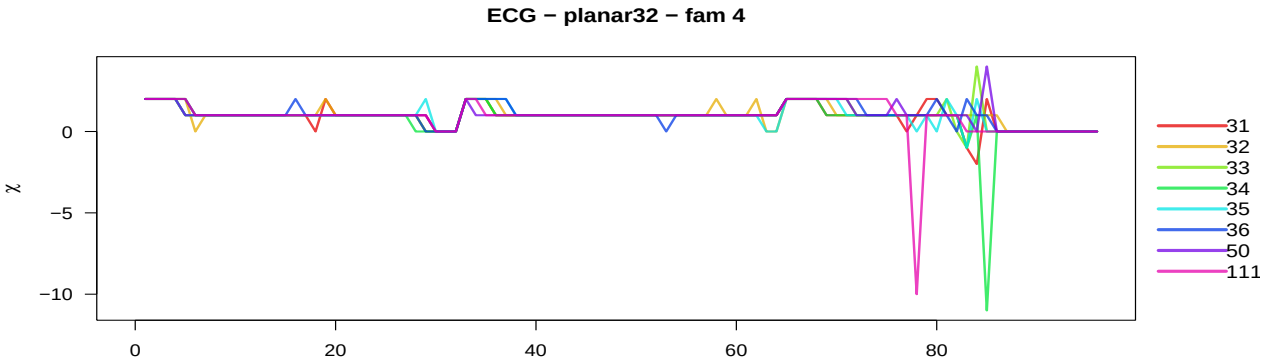
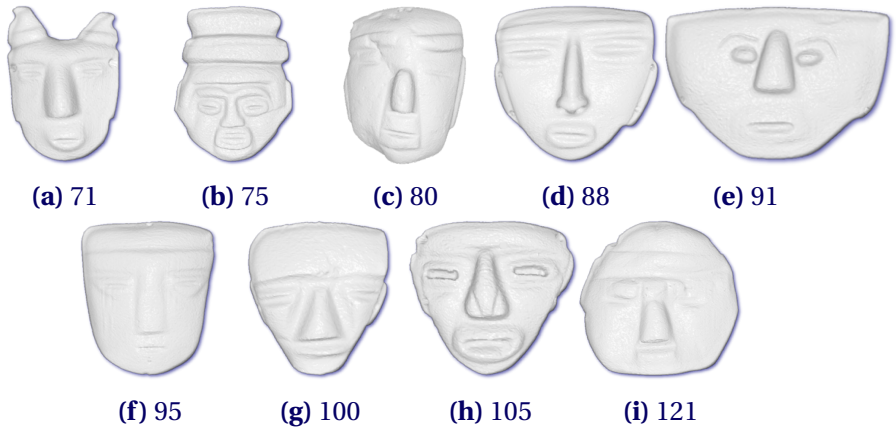


Figure 5.10: ECGs of newly assigned Set 04 based on planar sweeps f and $T = 32$

In the case of Set 04, two extra masks are assigned in it. One of them is an untrained mask originally from 05 and the other is from 10. Mask 111 stands out with a downwards acute peak in the last third of its ECG. As portrayed in Figure 5.9, mask 111 has little actual resemblance to the items in Set 04. Mask 50 is concave with a medium-sized nose, which might explain why its ECG suggests some similarity.



(a) 71 **(b)** 75 **(c)** 80 **(d)** 88 **(e)** 91

(f) 95 **(g)** 100 **(h)** 105 **(i)** 121

Figure 5.11: Masks assigned to Set 05 after running 72 polynomial SVMs

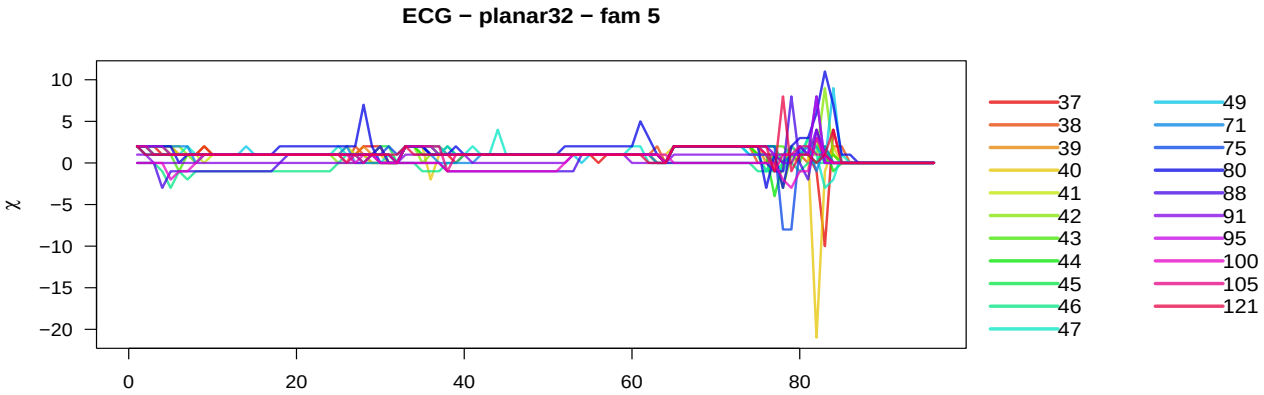


Figure 5.12: ECGs of newly assigned Set 05 based on planar sweeps f and $T = 32$

In the case of Set 05, it gained 9 masks from 10. The masks 40, 38, 70 have troughs while mask 80 stands out with the largest crest at the last third of their ECGs as seen in the Figure 5.12. These 9 masks also share physical similarities with masks in 05 used for training. For instance, they share triangular shaped noses (88, 91, 100, 105, 121) and slanted carved eyes (71, 80, 88, 95, 100, 105, 121). Observe that half of the masks in 05 have a triangular frame, which is found in masks 88, 91, 100, 105. Refer to Figure 5.11.

(a) 54 **(b)** 55 **(c)** 110

Figure 5.13: Masks in the new Set 06

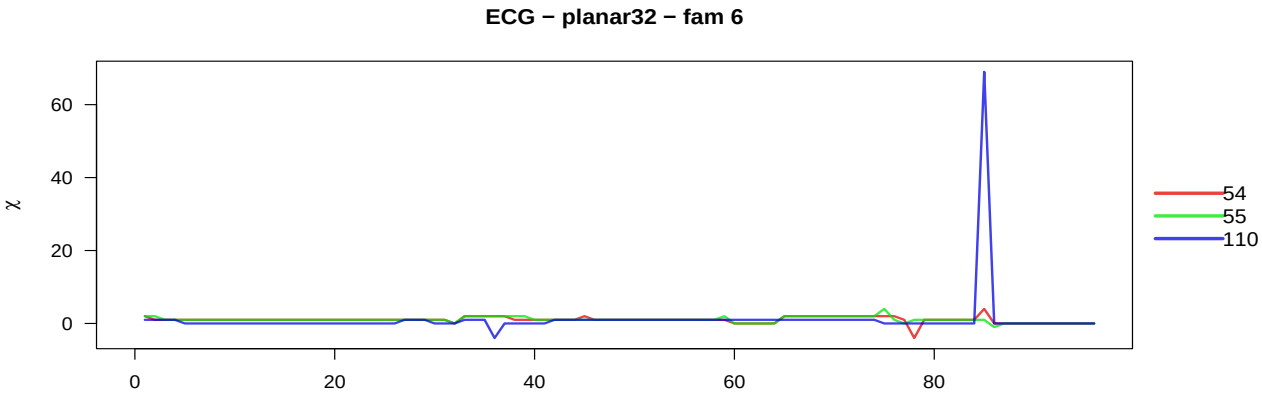


Figure 5.14: ECGs of newly assigned Set 06 based on planar sweeps f and $T = 32$

Set 06 was the smallest training set, with only two items. The polynomial SVM procedure assigned it only one extra mask from 10 as seen in Figure 5.13. The difference between the training ECGs and the ECG from mask 111, as shown in Figure 5.14, makes us conclude that the assignment ought to be due to 110 being least distant to 06 rather than close to it. It can also be said that they share almond-shaped, protruding eyes and the T-shaped eyebrows and nose.

(a) 23 **(b)** 84 **(c)** 118

Figure 5.15: Masks assigned to Set 07 after running 72 polynomial SVMs

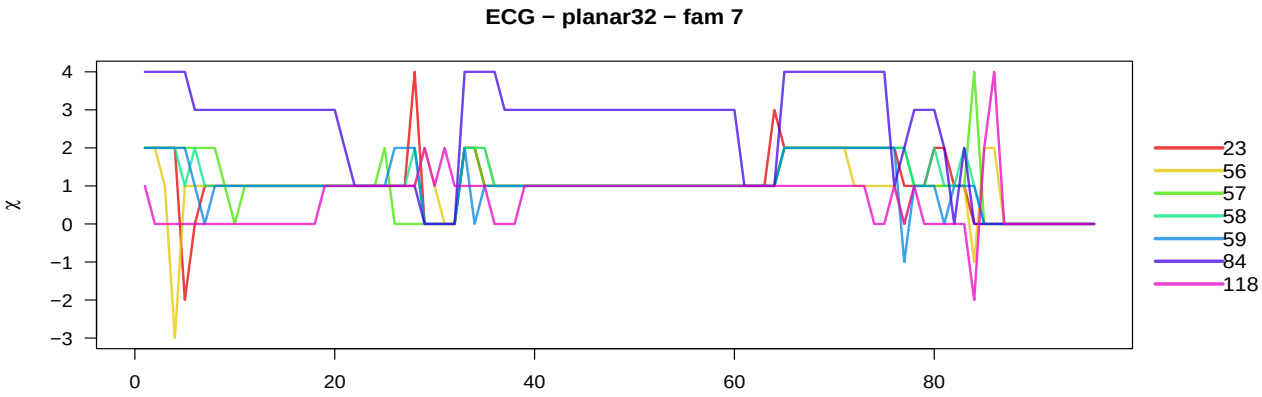


Figure 5.16: ECGs of newly assigned Set 07 based on planar sweeps f and $T = 32$

One of the untrained masks from Set 02 and 2 masks from 10 were assigned to Set 07. As depicted by Figure 5.15, aside from the rough aspect of mask 118, there are no similarities of the suggested additional items when compared to the original items in Set 07 depicted in Figure A.6. This lack of physical similarities in turn is reflected by no clear pattern followed by the ECGs as seen in Figure 5.16.

(a) 77 **(b)** 104 **(c)** 106 **(d)** 107 **(e)** 108 **(f)** 109

Figure 5.17: Masks assigned to Set 08 after running 72 polynomial SVMs

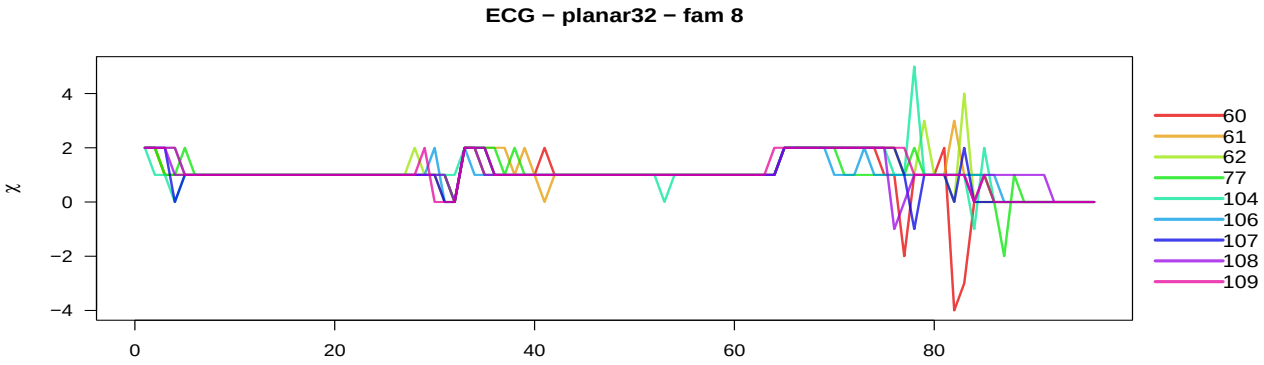
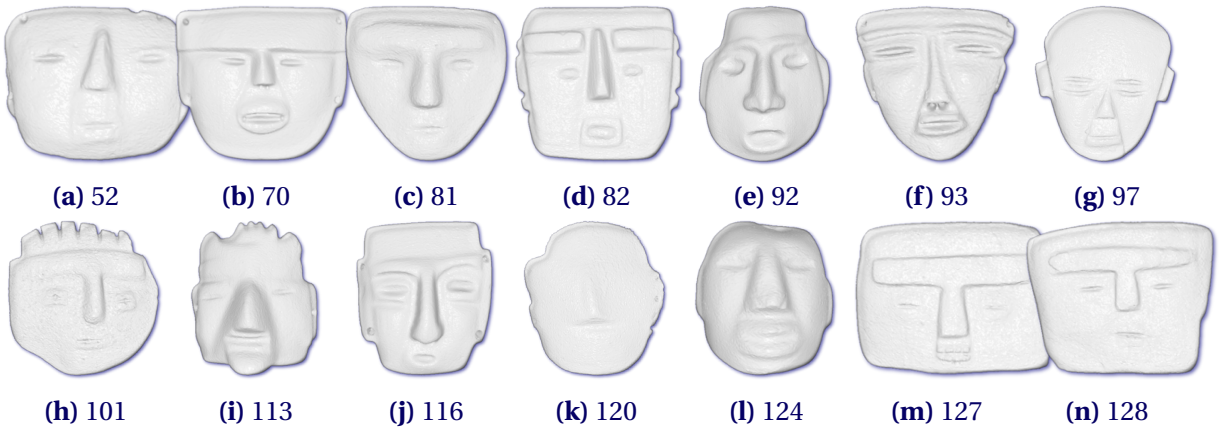


Figure 5.18: ECGs of newly assigned Set 08 based on planar sweeps f and $T = 32$

Set 08 gains 7 masks from Set 10. The ECG of masks 61, 77, 107, 108 present slight downwards spikes while masks 62, 104 have small peaks through the last third of the graph. Refer to Figure 5.18. There are some physical similarities between the training and test masks, such as triangular nose (77, 106, 107), round frame (104, 106, 108), and large slanted eyes with thick eyelids (104, 106, 108).



(a) 52 **(b)** 70 **(c)** 81 **(d)** 82 **(e)** 92 **(f)** 93 **(g)** 97

(h) 101 **(i)** 113 **(j)** 116 **(k)** 120 **(l)** 124 **(m)** 127 **(n)** 128

Figure 5.19: Masks assigned to Set 09 after running 72 polynomial SVMs

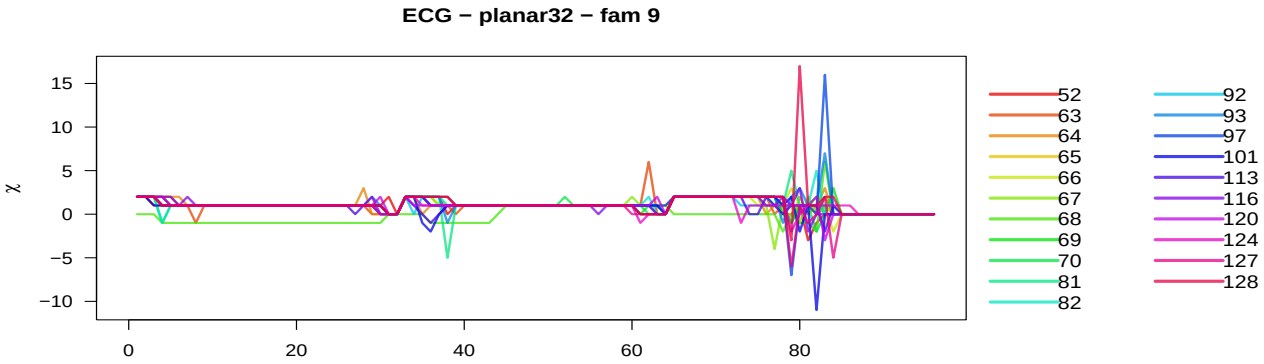


Figure 5.20: ECGs of newly assigned Set 09 based on planar sweeps f and $T = 32$

Finally, one of the untrained masks from Set 05 and 13 masks from 10 are assigned to Set 09, As seen in Figure 5.20, there is no definite pattern followed by these family. Mask 52 has a large peak towards the last third of its ECG, while mask 97 has both peaks and troughs in its final third as well. There is no definite common physical feature to be distinguished among all the new items in 09 although subsets of them share some similarities with the items from the training set. For instance, some of them have a clearly defined T-shaped nose and eyebrows (70, 81,82, 93, 101, 116, 127, 128), their eyes are carved into a squint (52, 70, 81, 93, 97, 101, 116, 127, 128) and some of them have a squared frame (52, 70, 82, 116, 127, 128, 113).

There were also eight masks (6.2% of the total dataset) with undecided assignment, as several of the different SVM tests would place them into different families. These are depicted in Figure 5.21, For instance, the mask 85 was assigned 35 times each to both

(a) 51 **(b)** 72 **(c)** 74 **(d)** 83 **(e)** 85 **(f)** 86 **(g)** 96 **(h)** 122

Figure 5.21: Masks assigned to different families by each SVM test

Sets 04 and 05. This could mean that mask 35 lies in between both classes and it is not clear which is the right assignment.

Observe that mask 72 has carved slanted eyes, T-shaped nose and eyebrows and defined ears, thus sharing physical similarities effectively to those in Set 03. This agrees with the fact that it was assigned to Set 03 most of the times (61 times). Mask 51 was found originally in Set 05 and excluded from the training set for testing purposes. It is interesting to find that it was labeled as 05 most of the times (51 times).

It is worth noting that mask 85 has holes in its eyes but the SVM never considered assigning it to Set 02. The reason might be that the holes in 85 appear not to be part of the original design of the mask. Thus the ECGs are sensitive enough to notice this difference.

Using $T = 64$ thresholds

Just as in the case above, when the three filter functions $\pi^{(i)}$ were run with $T = 64$ thresholds, all the training data was classified in its original family. It stands out that 20 masks in total (15.6% of the total dataset) were deemed undefined as less than 85% of the SVM tests agreed on a same family. Other observations stood out when the results are compared against the planar sweep filtering with $T = 32$ thresholds detailed above.

- Set 02 remained with the same items as in Figure 5.5, which indicates that the classifiers are still able to distinguish masks with holes apart from the rest.
- Set 02 still reports mask 125, although it has no holes.
- The Set 03 reports only 8 additional masks in total, all of them from Set 10, 9 masks less than before.
- No masks from Sets 02 or 05 were reported in Set 03 as before.

- Masks 101 and 102 were reported in Set 03 (refer to Figure A.10, although their squinted thick eyes resemble those found in the training Set 09.
- The Set 04 was assigned 6 additional masks in total, four items more than before.
- None of the assigned items to 04 share physical features (prominent nose) with the training masks in 04.
- No items were reported in Set 06. As 06 training set contains only two different items, it is possible that the data is too sparse (due to high dimensionality as mentioned in Subsection 4.6) to define a training label at all.
- Set 07 reports the same additional three masks as before, albeit none of them share strong physical features with the training items in 07.

Using $T = 128$ thresholds

When the three filter functions $\pi^{(i)}$ were run with $T = 128$ thresholds, 15 masks, 11.7% of the total, were deemed unidentified by the stated SVM procedure. All the training masks were assigned to their respective set during the test phase. A noticeable change is the fact that Set 09 grouped more than 35 items in total while its training set consists only of 7 different masks. At the same time, Set 05 reported only two additional masks despite using than 13 training items in this set. This might indicate some negative effects due to the high-dimensionality of the data, such as overfitting or sparsity as discussed in Section 4.6. Thus, we do not have as much confidence in these results as in the other cases. Nevertheless, for Set 02 the same 7 masks in Figure 5.5 remained assigned. Despite the high dimensionality effects, the holes for either eyes or mouth are still a powerful distinction captured by the ECGs.

5.7.2 Cylindrical Sweeps

Just as with the planar sweeps, every item used for training was assigned to its corresponding family when tested with the procedure initially described. The experiments with planar sweeps proved that increasing the number T of thresholds did not yield better results due to high dimensionality effects. For the cylindrical sweeps experiments, we decided thus to keep fixed the number of thresholds $T = 32$ and instead vary the M_g bound as referred in (5.3.2).

Using $T = 32$ and $M_g = \sqrt{2}$ thresholds

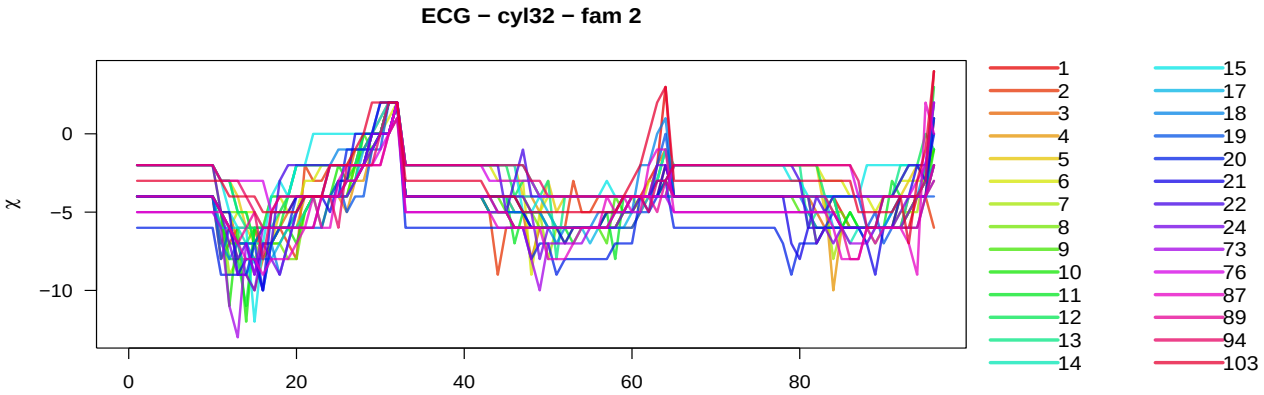
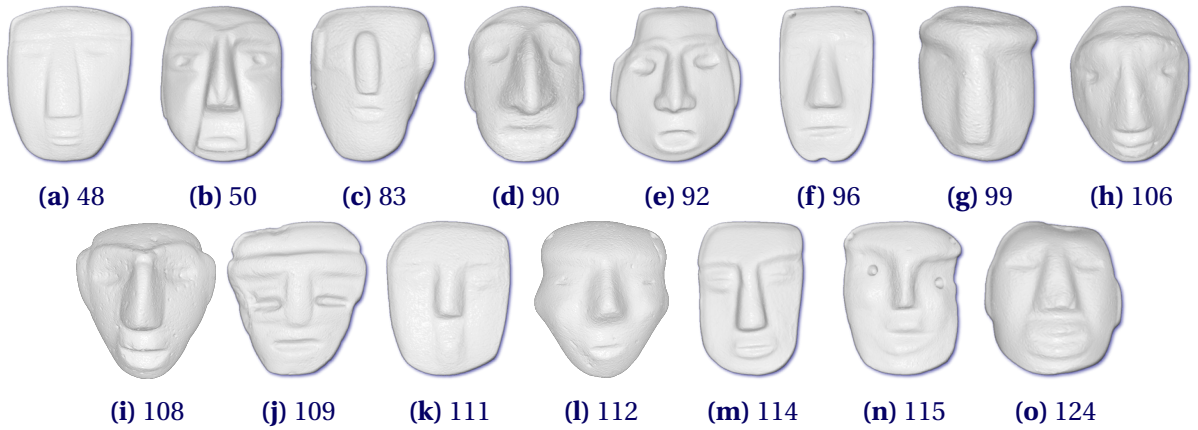


Figure 5.22: ECGs of newly assigned Set 02 based on cylindrical sweeps f and $T = 32$

When considering a polynomial kernel, following the procedure described above, the Set 02 was assigned 6 new items from Set 10. These were masks 72, 76, 87, 89, 94 and 103 as seen in Figure 5.5. Contrary to the planar sweep filter, mask 125 was excluded and now the only masks gathered in Set 02 are the ones with holes for their eyes or mouth. The ECGs seen in Figure 5.22 follow roughly the same pattern.



(a) 48 **(b)** 50 **(c)** 83 **(d)** 90 **(e)** 92 **(f)** 96 **(g)** 99 **(h)** 106

(i) 108 **(j)** 109 **(k)** 111 **(l)** 112 **(m)** 114 **(n)** 115 **(o)** 124

Figure 5.23: Masks assigned to Set 03 after running 72 polynomial SVMs

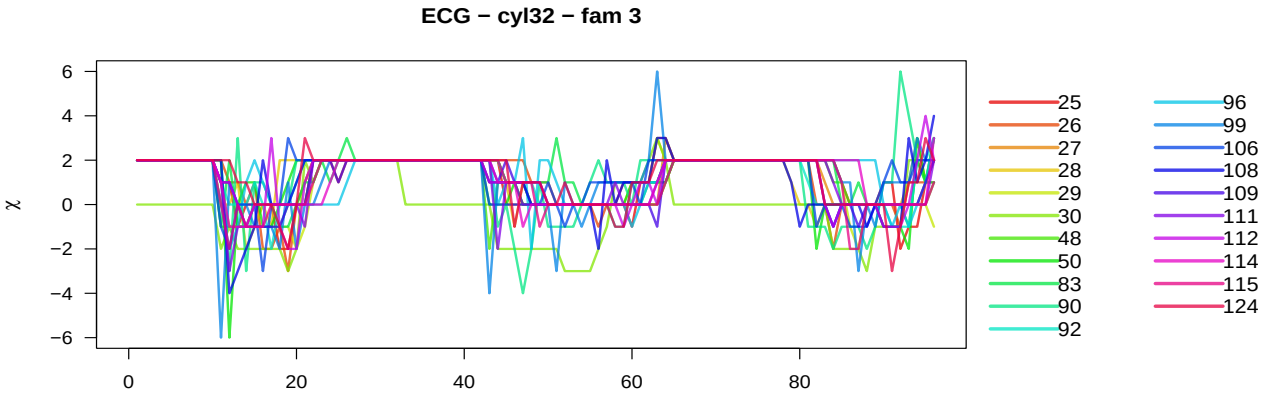


Figure 5.24: ECGs of newly assigned Set 03 based on cylindrical sweeps f and $T = 32$

There were 13 masks from Set 10 that were assigned to 03. Masks 48 and 50 (originally from 05) are assigned again to Set 03. The ECGs in Figure 5.24 reveal that masks 99 and 83 stand out slightly with peaks in the middle and last thirds respectively. All the assigned masks can be seen in Figure 5.23. Masks 48, 90, 99, 112, 114, and 115 were assigned once again to 03. Notice that they share certain physical characteristics with the originals from Set 03, such as an elongated frame (48, 50, 83, 90, 96, 99, 106, 114, 124), defined and visible ears (50, 90, 92, 108, 109, 112, 124) a clear T-shaped nose and eyebrows (48, 50, 83, 92, 96, 99, 102, 109, 111, 114, 115), trapezoidal nose (48, 50, 90, 92, 96, 99, 106, 114, 124) and carved eyes with out eyelids or no eyes at all (48, 83, 90, 92, 96, 99, 106, 115, 124).

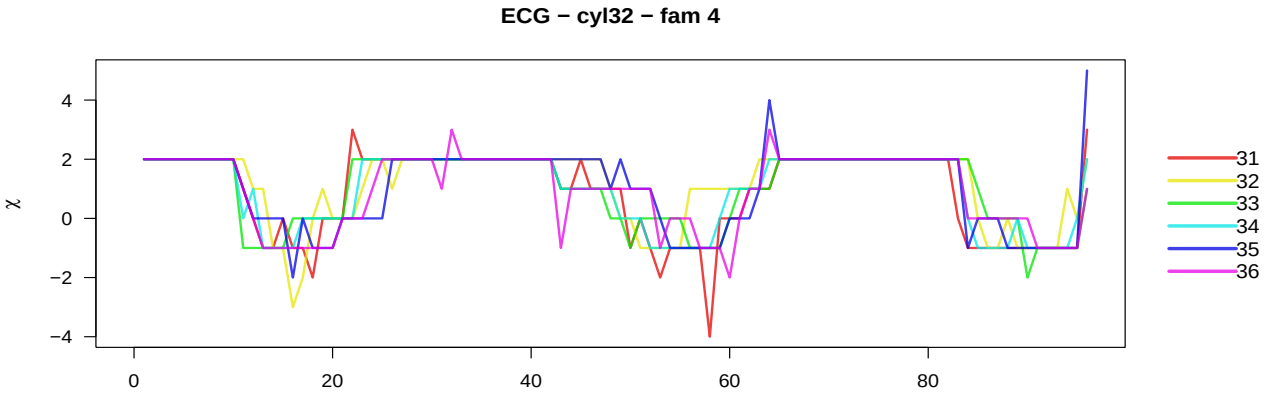
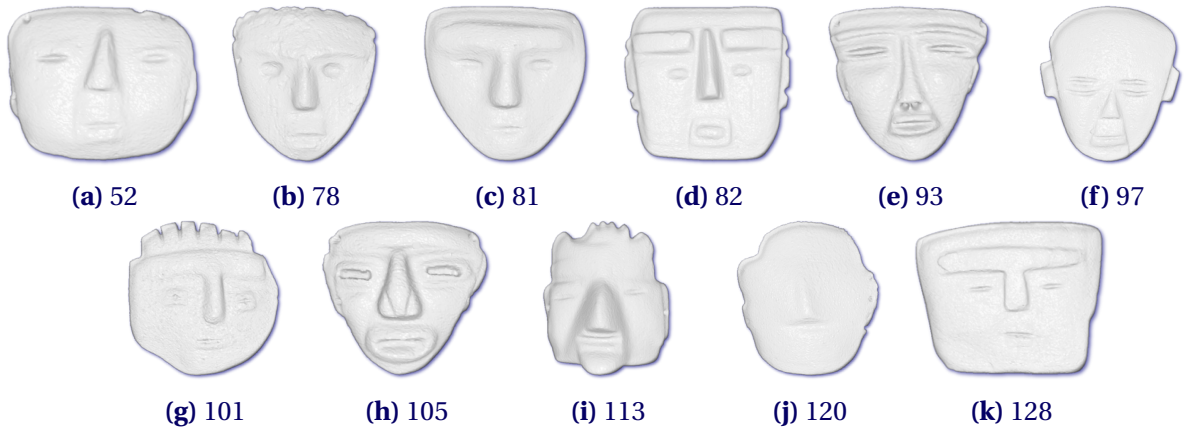


Figure 5.25: ECGs of newly assigned Set 04 based on cylindrical sweeps f and $T = 32$

In the case of Set 04, surprisingly no mask is assigned to this set except for its original items which were used for training. This is an encouraging result, as no other mask outside 04 shares their most prominent physical feature, which is a very protruding nose. We see that the ECGs from the training items in 04 follow all a certain pattern.



(a) 52 **(b)** 78 **(c)** 81 **(d)** 82 **(e)** 93 **(f)** 97

(g) 101 **(h)** 105 **(i)** 113 **(j)** 120 **(k)** 128

Figure 5.26: Masks assigned to Set 05 after running 72 polynomial SVMs

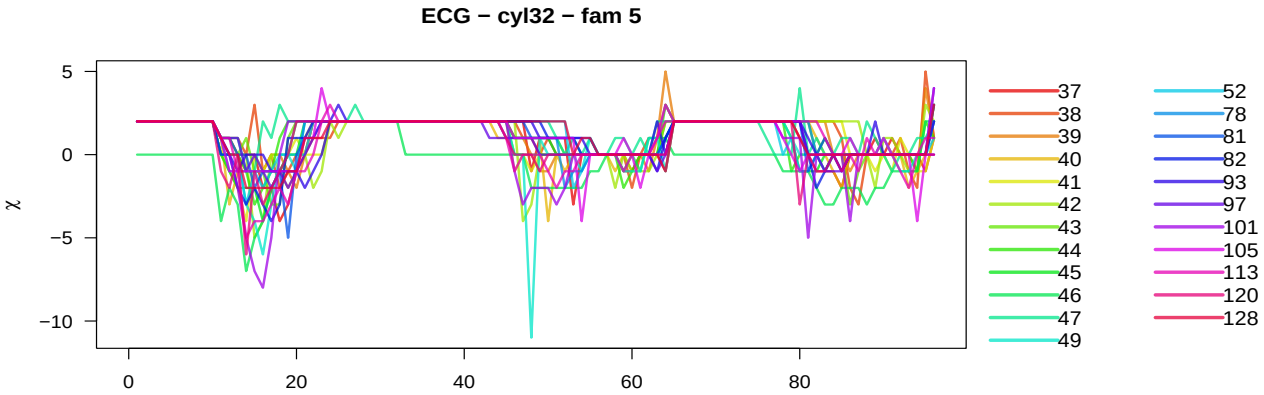


Figure 5.27: ECGs of newly assigned Set 05 based on cylindrical sweeps f and $T = 32$

In the case of Set 05, it gained 10 masks from 10. It also recovered mask 52, which was originally in 05 and excluded from training for testing purposes. Apart from a sharp trough during the middle third in the ECG of mask 49, all the ECGs show a certain resemblance as seen in the Figure 5.27. The proposed assortment in 05 with the cylindrical sweeps filter is completely different from the one proposed by the planar sweeps filter. The only non-training mask that both assignments have in common is mask 105. The 11 assigned masks also share physical similarities with masks in 05 used for training. For instance, they share triangular shaped noses (52, 78, 82, 93, 97, 105, 113, 120) and slanted carved eyes with eyelids (52, 93, 97, 105, 113), triangular-shaped chin (78, 81, 93, 101, 105) and defined ears (52, 78, 82, 93, 97, 105, 113). Refer to Figure 5.26.

(a) 54 **(b)** 55 **(c)** 84

Figure 5.28: Masks in the new Set 06

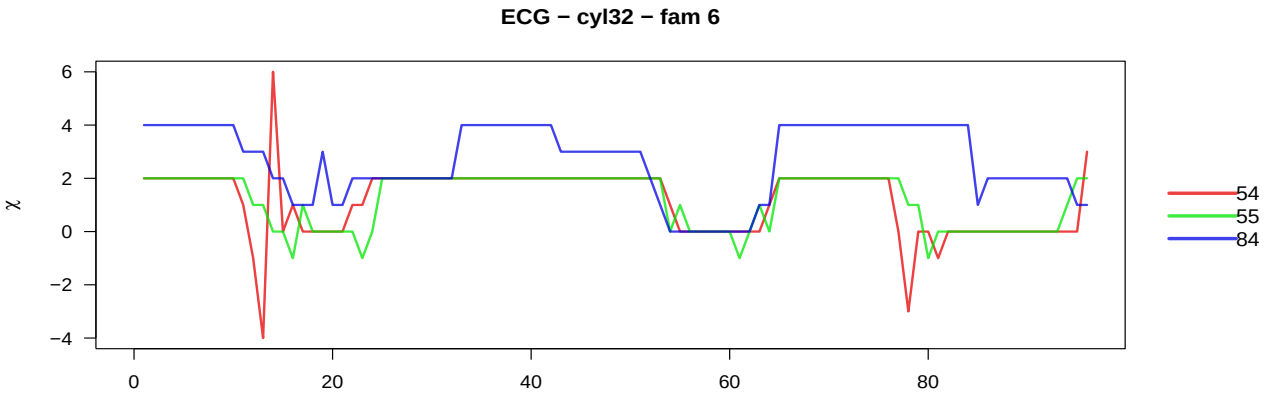


Figure 5.29: ECGs of newly assigned Set 06 based on cylindrical sweeps f and $T = 32$

The polynomial SVM procedure assigned Set 06 only one extra mask from 10 as seen in Figure 5.28. There is not a clear resemblance between the ECG of mask 84 and the training ECGs from 06 as shown in Figure 5.29. This poor resemblance in ECGs is also reflected in poor physical resemblance of masks. Observe that mask 84 is different from mask 111 proposed in the case of planar sweep filters (Figure 5.13).

(a) 80 **(b)** 123 **(c)** 126

Figure 5.30: Masks assigned to Set 07 after running 72 polynomial SVMs

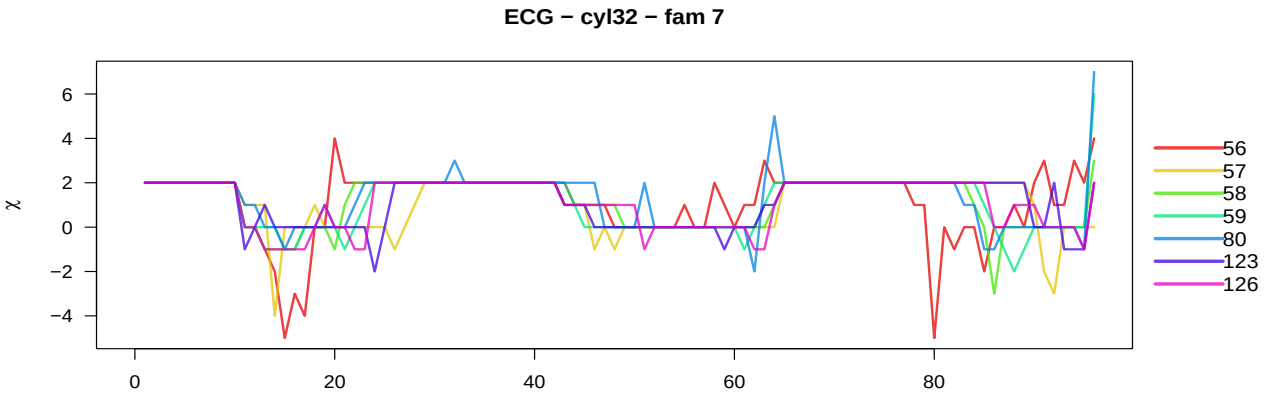


Figure 5.31: ECGs of newly assigned Set 07 based on cylindrical sweeps f and $T = 32$

Three masks from Set 10 were assigned to Set 07. Unlike the ECGs in Figure 5.16, as depicted by Figure 5.30, aside from some erratic behavior in the ECG of mask 56, there is a certain pattern followed by the items in this group. None of these three were considered by the planar sweep filter. There is also physical resemblance among the items. The three new items have an elongated and thin frame. The ECG seems to identify that the masks lack eyes and have few details overall as if left unfinished.

(a) 75 **(b)** 104 **(c)** 121

Figure 5.32: Masks assigned to Set 08 after running 72 polynomial SVMs

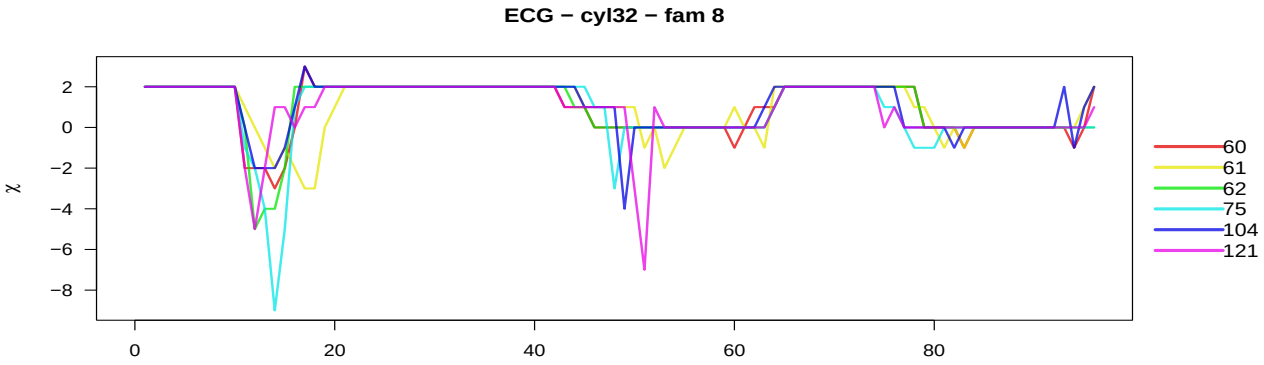
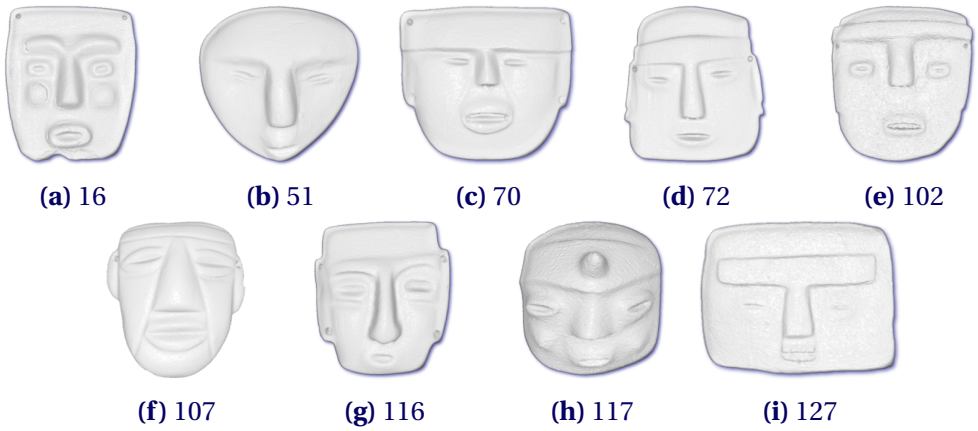


Figure 5.33: ECGs of newly assigned Set 08 based on cylindrical sweeps f and $T = 32$

For Set 08, it gains three masks from 10. The ECG of masks 75 and 121 present troughs during their first and middle third respectively, as seen in Figure 5.33. Mask 104 was assigned here once again. All of the masks share a round frame, however, it is unclear which other physical features they might share with the training items in 08.



(a) 16 **(b)** 51 **(c)** 70 **(d)** 72 **(e)** 102

(f) 107 **(g)** 116 **(h)** 117 **(i)** 127

Figure 5.34: Masks assigned to Set 09 after running 72 polynomial SVMs

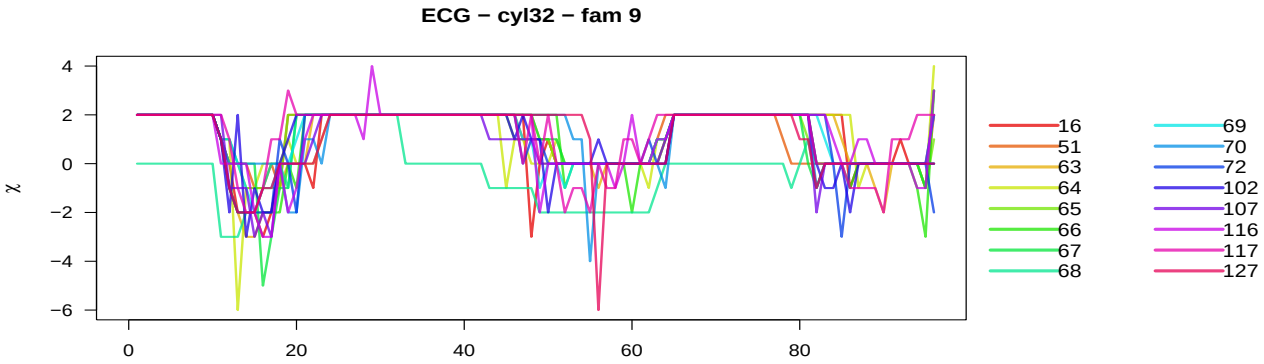
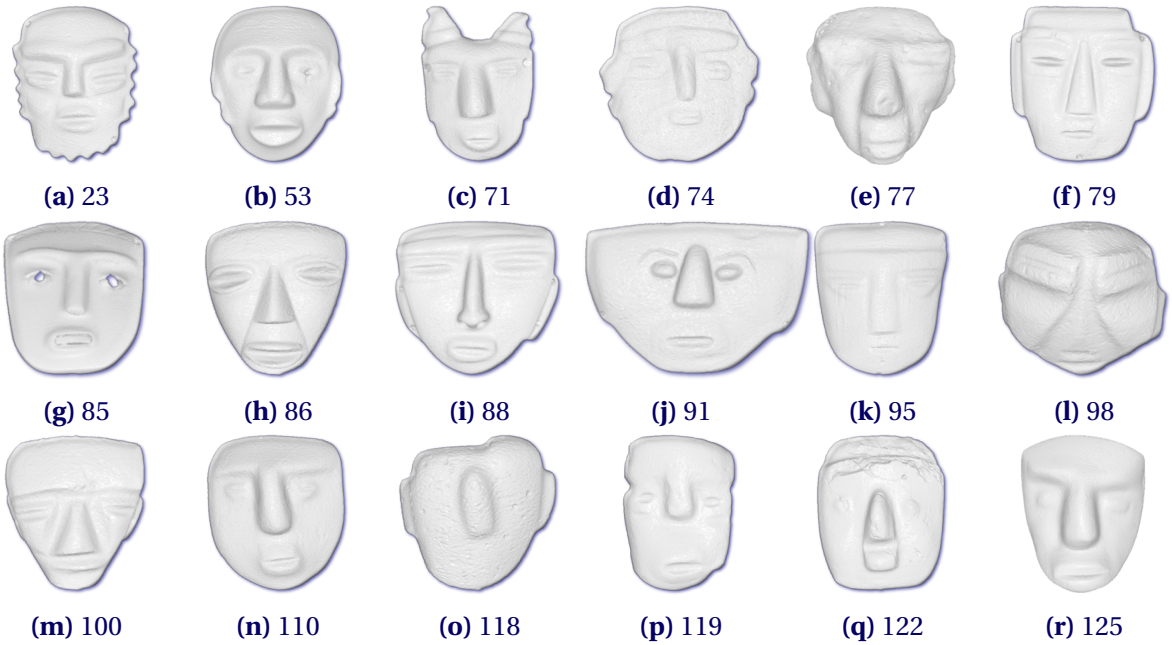


Figure 5.35: ECGs of newly assigned Set 09 based on cylindrical sweeps f and $T = 32$

Finally, one of the untrained masks from Set 02, another one untrained from 05 and 7 masks from 10 are assorted in Set 09, Masks 70, 116 and 127 are assigned to 09 once again. As seen in Figure 5.35, there is some pattern followed by these family. The masks also share some common physical features with the training items in 09. For instance, a clearly defined T-shaped nose and eyebrows (16, 72, 102, 107, 116, 117), their eyes are carved into a squint with thick protruding eyelids and lips (16, 72, 102, 116, 127) and some of them have a squared frame (16, 70, 102, 116, 127).

There were also 18 masks (14.1% of the total dataset) with undecided assignment, as several of the different SVM tests would place them into different families. Masks 74, 85, 86 and 122 were once again deemed with an uncertain labeling. These are depicted in Figure 5.36,

Observe that mask 119 has an elongated frame with rough facial features, so it is not



(a) 23 **(b)** 53 **(c)** 71 **(d)** 74 **(e)** 77 **(f)** 79

(g) 85 **(h)** 86 **(i)** 88 **(j)** 91 **(k)** 95 **(l)** 98

(m) 100 **(n)** 110 **(o)** 118 **(p)** 119 **(q)** 122 **(r)** 125

Figure 5.36: Masks assigned to different families by each SVM test

surprising that it was predominantly labeled as Set 07. Masks 71, 79, 95 and 125 have a T-shaped nose and eyebrows and clearly defined thick eyelids and lips, thus they were predominantly placed in Set 09. However, mask 79 also has a triangular nose, with slanted eyes and defined ears, thus it is reasonable to also be confused with Set 05. Masks 100 and 110 have both carved eyes into a squint with thick eyelids and lips. Mask 100 also has a triangular nose and a triangular chin, resembling items in 05, while mask 110 has a more defined T-shaped nose and eyebrows, resembling items in 09. However, for both masks 100 and 110, these physical features are less defined and protruding as with the training masks. Thus the SVMs are usually confused about their labeling between these two families.

It is worth noting again that mask 85 was never considered for assignment in Set 02 although it has holes in its eyes.

Using $T = 32$ thresholds with $M_g = 1$

With a tighter M_g bound, all the training items would still be classified in their respective sets during the training phase. The SVM procedure in this case is inconclusive for 20

masks (15.6% of the whole dataset), two masks more than the $M_g = \sqrt{2}$ case above. For the remainder of the subsection, all the comparisons mentioned here will be made against the SVM procedure above, with $T = 32$ thresholds and $M_g = \sqrt{2}$. Of all the new proposed Sets, none of them coincide with the classified Sets proposed before. This shows that changes in just one variable affect the overall classification of items in Set `10` during the test phase. Several observations were made regarding the results obtained in this case.

- The Set `02` was assigned the same six masks as before, which shows that the filter function still can distinguish holes from the masks eyes or mouth.
- Sets `03, 05` and `09` offer sensible results, where most of the additional proposed masks for each of these Sets share some physical features with the training items.
- The Set `07` reports a total addition of 7 items, four more masks than before. Other than rough, unfinished appearance, little is found in common with the training items.
- Less than two additional masks were assigned for Sets `04, 06` and `08`. This is encouraging, as none of the items in Set `10` share any physical resemblance with the training items from these sets.

Overall, the choice of parameters seems to produce sensible results that could be explored further. This result supports our hypothesis that tighter M_g bounds will produce more detailed ECGs. This gain of details could be due to the fact that the separation between threshold values is smaller, while at the same time the dimension of the produced ECG vector remains the same. Thus the ECG records finer changes while avoiding high dimensionality effects.

Using $T = 32$ thresholds, $M_g = 2$

By using the rougher bound $M_g = 2$, all the training items were still predominantly assigned to their respective sets during the test phase. However, rougher bounds for a fixed T value will imply that the threshold values are more distant from one another. This in

turn implies that the the resulting ECGs will be less detailed. This lack of detail is perhaps the reason on why 22 masks (17% of the total dataset) were deemed unidentified. Several observations stand out after carrying the SVM procedure.

- Set 02 reported the same outcome as in the last two procedures. Even though the M_g is rougher, the ECGs can still distinguish whether a mask has holes for eyes.
- Sets 02 and 05 report sensible assignments. The number of extra assignments, and the assignments themselves, are similar to the ones presented when $M_g = \sqrt{2}$ or $M_g = 1$.
- Sets 04 and 06 report less than 2 extra masks each, which is considered a good result as none of the masks in Set 10 share physical features with either of these two families.
- Sets 07 and 08 report six additional assignments, although each set has only 4 and 3 training items respectively. None of the extra items share strong physical features with the training items.
- Set 09 reports only 3 additional items, although there are several masks in Set 10 that have thick, squinted eyes and T-shaped nose and eyebrows, the most distinctive features of Set 09.

5.7.3 Spherical Sweeps

The spherical sweeps filter function h as described above was the only non-compound ECG filter function considered. Thus it seemed reasonable to use $T = 128$ and $T = 64$ thresholds for each ECG. The procedures were also carried with three different bounds M_h as in (5.3.2). Just as with the planar sweeps, every item used for training was assigned to its corresponding family when tested with the procedure initially described.

Using $T = 128$ thresholds and $M_h = \sqrt{2}$

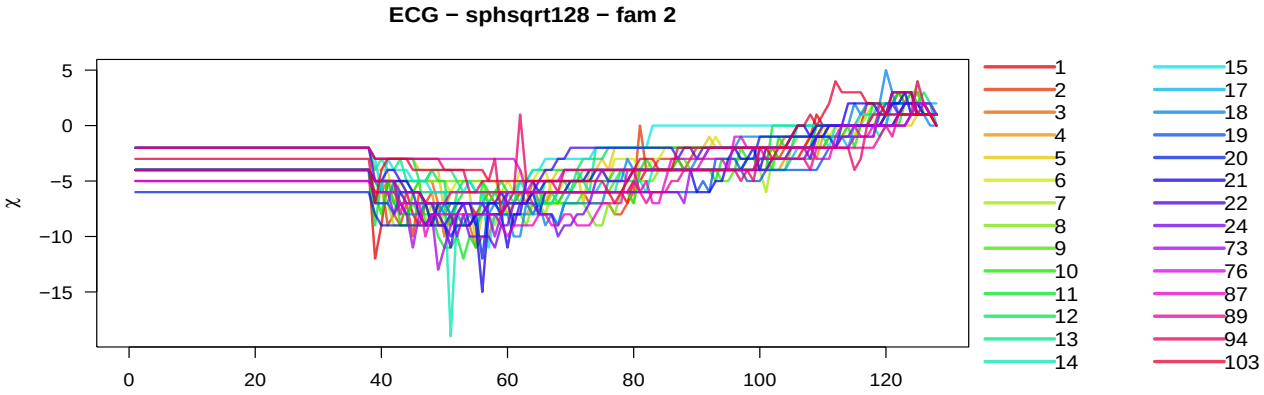
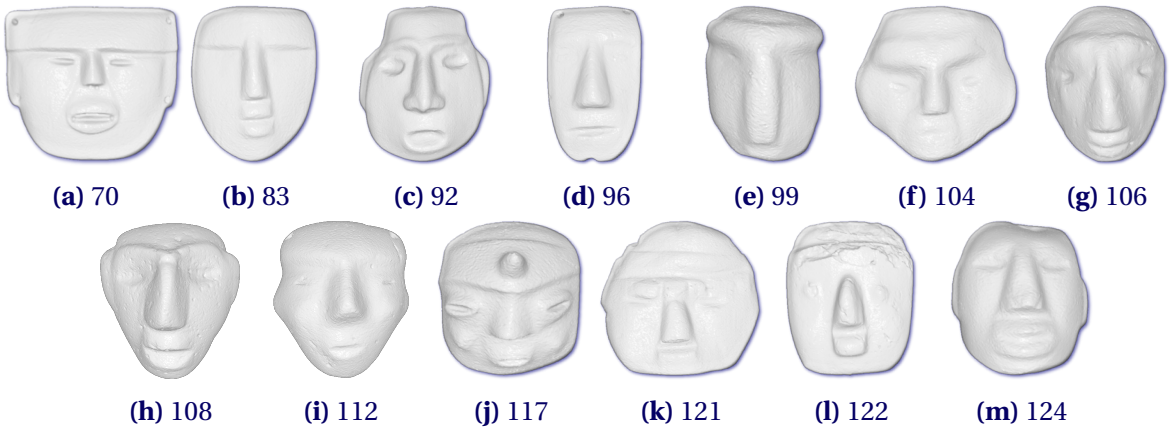


Figure 5.37: ECGs of newly assigned Set 02 based on spherical sweeps f and $T = 32$

When considering a polynomial kernel, following the procedure described above, the Set 02 was assigned again the same 6 items as in the cylindrical sweeps case in Figure 5.22. The ECGs seen in Figure 5.37 follow roughly the same pattern.



(a) 70 **(b)** 83 **(c)** 92 **(d)** 96 **(e)** 99 **(f)** 104 **(g)** 106

(h) 108 **(i)** 112 **(j)** 117 **(k)** 121 **(l)** 122 **(m)** 124

Figure 5.38: Masks assigned to Set 03 after running 72 polynomial SVMs

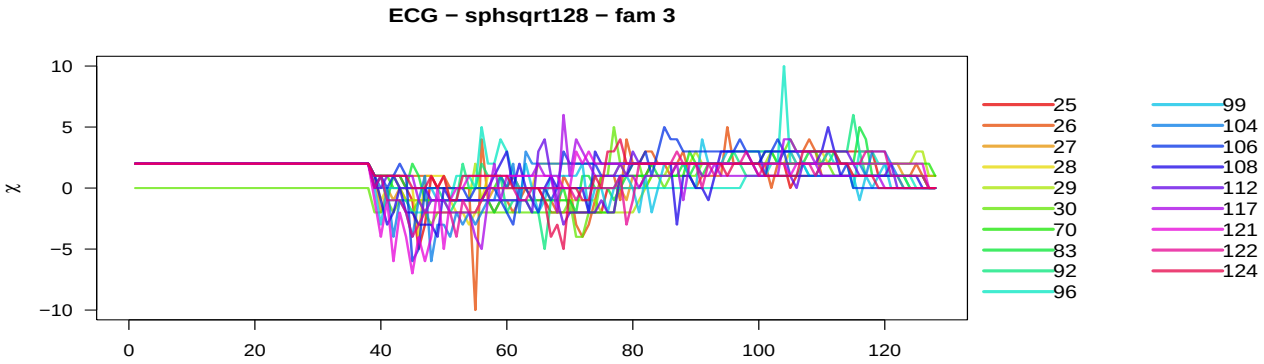


Figure 5.39: ECGs of newly assigned Set 03 based on spherical sweeps f and $T = 32$

There were 13 masks from Set 10 that were assigned to 03. Unlike the previous two filters, no untrained mask from Set 05 were assigned to 03 this time. All the assigned masks can be seen in Figure 5.38. Masks 83, 92, 96, 99, 106, 108, 112 and 124 were assigned once again to 03 as in the cylindrical sweeps filter. Notice that they share certain physical characteristics with the originals from Set 03. Except for masks 108 and 106, all of them share a triangular or trapezoidal nose. Except for mask 117, all the masks have carved, horizontal eyes with no eyebrows, or no eyes at all. Except for masks 99, 108, 112, all of them have a defined horizontal mouth and lips. Additionally, masks 70, 92, 104, 108, 112, 121 and 124 have defined ears. Masks 93, 96, 99, 106 and 122 have an oblong frame as well.

(a) 93 **(b)** 123

Figure 5.40: Masks assigned to Set 04 after running 72 polynomial SVMs

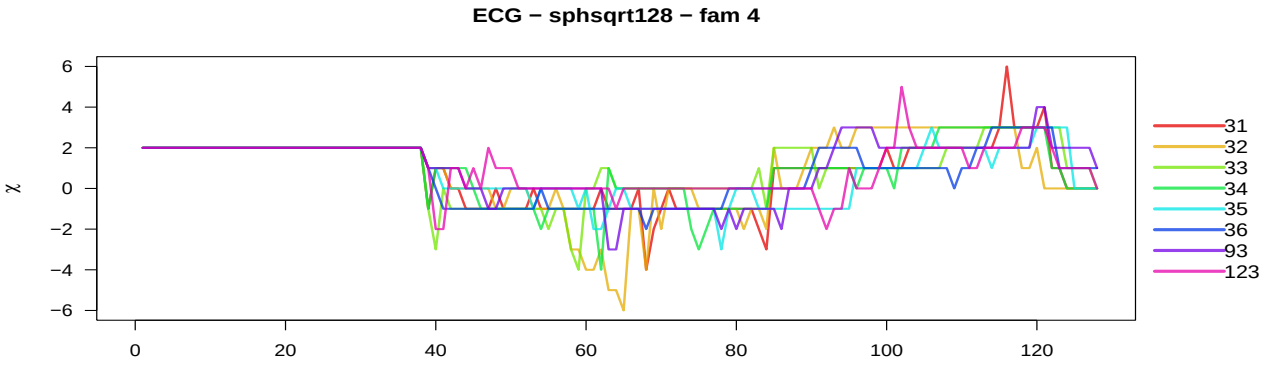
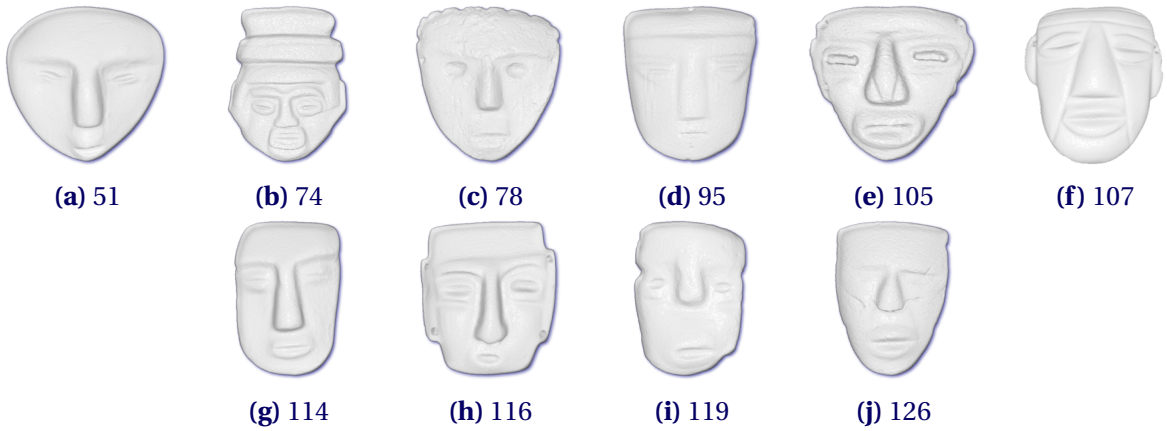


Figure 5.41: ECGs of newly assigned Set 04 based on spherical sweeps f and $T = 32$

In the case of Set 04, two masks from 10 were added, 98 and 123. None of these two had been considered as part of 04 before, although they share no physical resemblance with the training items. We see that the ECGs from the training items in 04 follow all a certain pattern.



(a) 51 **(b)** 74 **(c)** 78 **(d)** 95 **(e)** 105 **(f)** 107

(g) 114 **(h)** 116 **(i)** 119 **(j)** 126

Figure 5.42: Masks assigned to Set 05 after running 72 polynomial SVMs

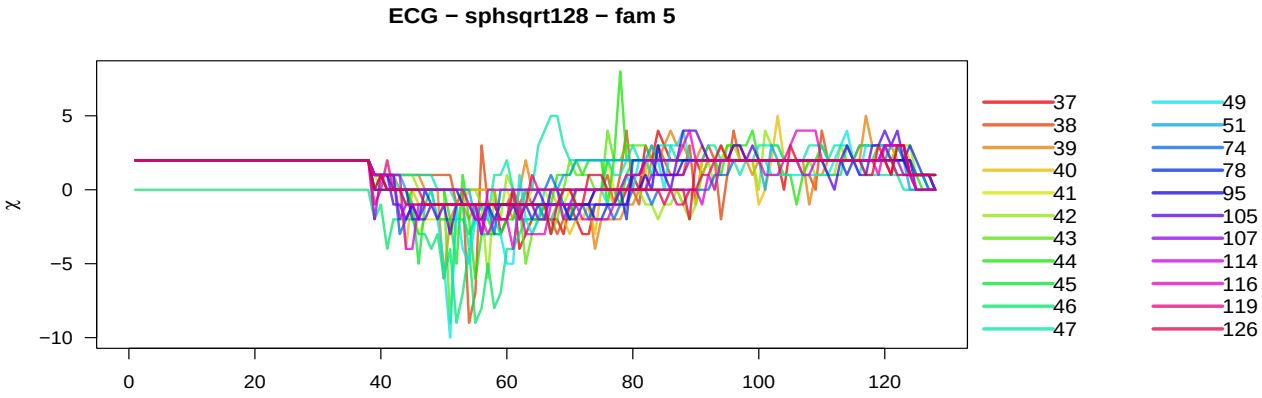


Figure 5.43: ECGs of newly assigned Set 05 based on spherical sweeps f and $T = 32$

In the case of Set 05, it gained 9 masks from 10. It also recovered the untrained mask 51. The proposed assortment in 05 with the cylindrical sweeps filter is very different as the proposed by the planar sweeps or cylindrical sweeps filters. The only non-training mask assignments that it has in common with the planar sweep filter are masks 95 and 105; the only assignments in common with the cylindrical sweeps filter are masks 78 and 105. All of the labeled masks have a trapezoidal nose. Except for masks 78 and 126, all of them have horizontally carved eyes with prominent eyelids and lips. Masks 74, 105, 107 and 116 have defined ears as well. Refer to Figure 5.42.

Figure 5.44: Mask 84, the only new assignment to set 06

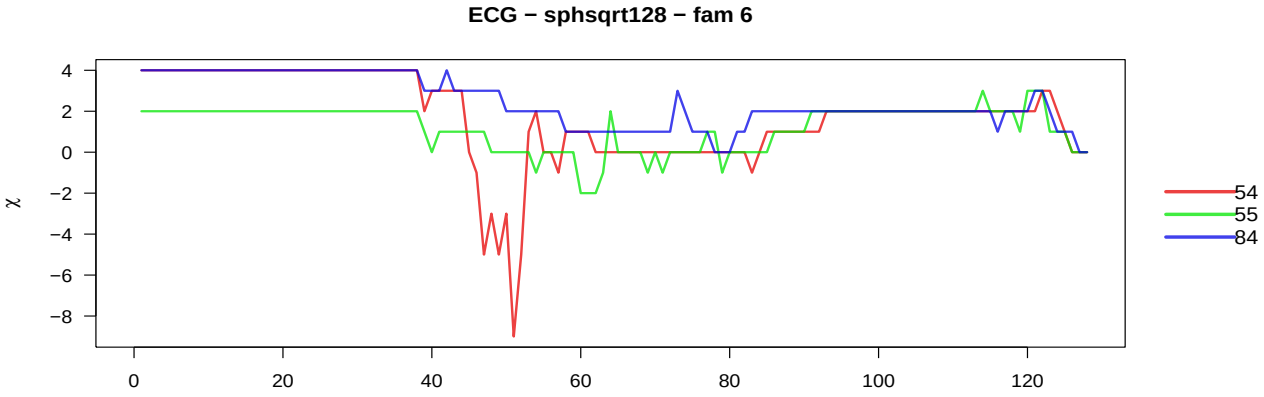
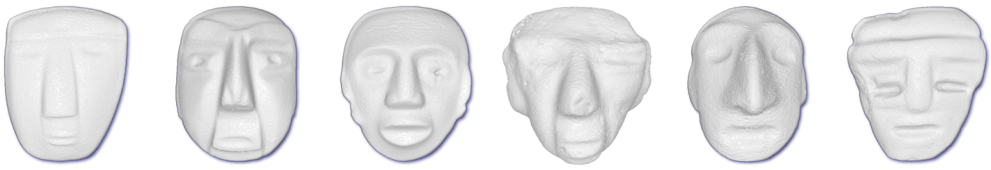


Figure 5.45: ECGs of newly assigned Set 06 based on spherical sweeps f and $T = 32$

Just as in the previous filter, the polynomial SVM procedure assigned Set 06 only one extra mask from 10 as seen in Figure 5.44. There is still not a clear resemblance between the ECG of mask 84 and the training ECGs from 06 as shown in Figure 5.45. As stated before, there is no physical resemblance either.



(a) 48 **(b)** 50 **(c)** 53 **(d)** 77 **(e)** 90 **(f)** 109

Figure 5.46: Masks assigned to Set 07 after running 72 polynomial SVMs

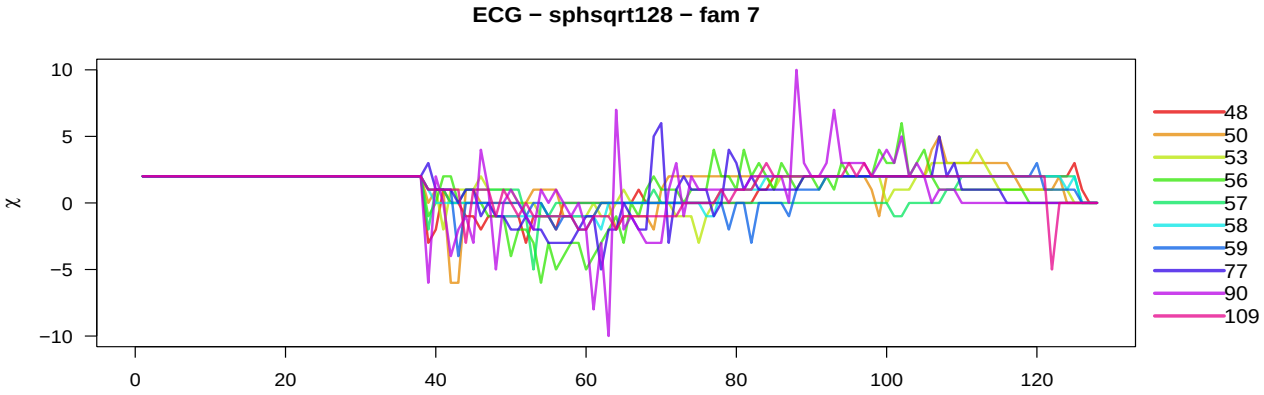


Figure 5.47: ECGs of newly assigned Set 07 based on spherical sweeps f and $T = 32$

Three masks from Set 05 and other three from Set 10 were assigned to Set 07. There little physical resemblance with the training items, expect for perhaps a rough, unfinished appearance (48, 77, 109) and straight cuts that delimit the area of the nose (50, 77). On the other hand, no additional mask was assigned in Set 08. As stated before, the training set 08 has only 3 items and no items in Set 10 resemble the training data.

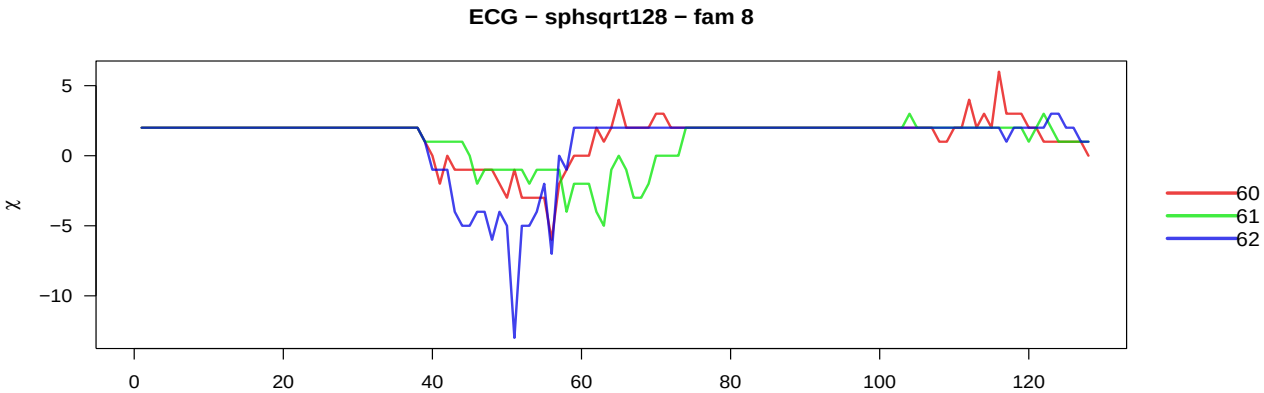
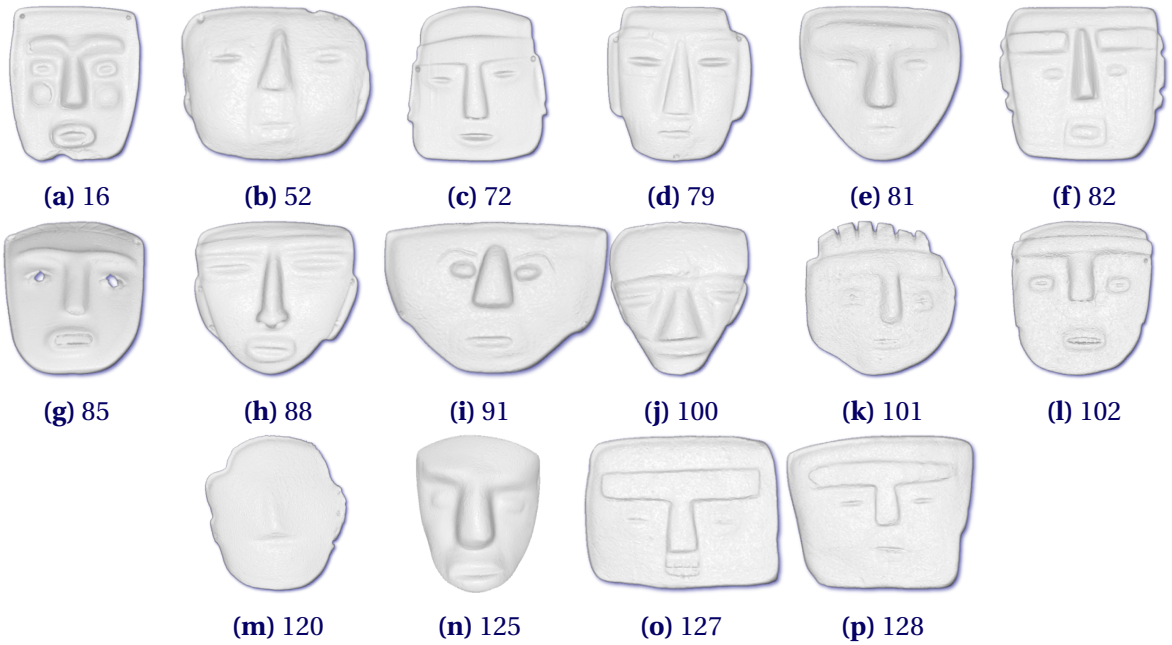


Figure 5.48: ECGs of newly assigned Set 08 based on spherical sweeps f and $T = 32$



(a) 16 **(b)** 52 **(c)** 72 **(d)** 79 **(e)** 81 **(f)** 82

(g) 85 **(h)** 88 **(i)** 91 **(j)** 100 **(k)** 101 **(l)** 102

(m) 120 **(n)** 125 **(o)** 127 **(p)** 128

Figure 5.49: Masks assigned to Set 09 after running 72 polynomial SVMs

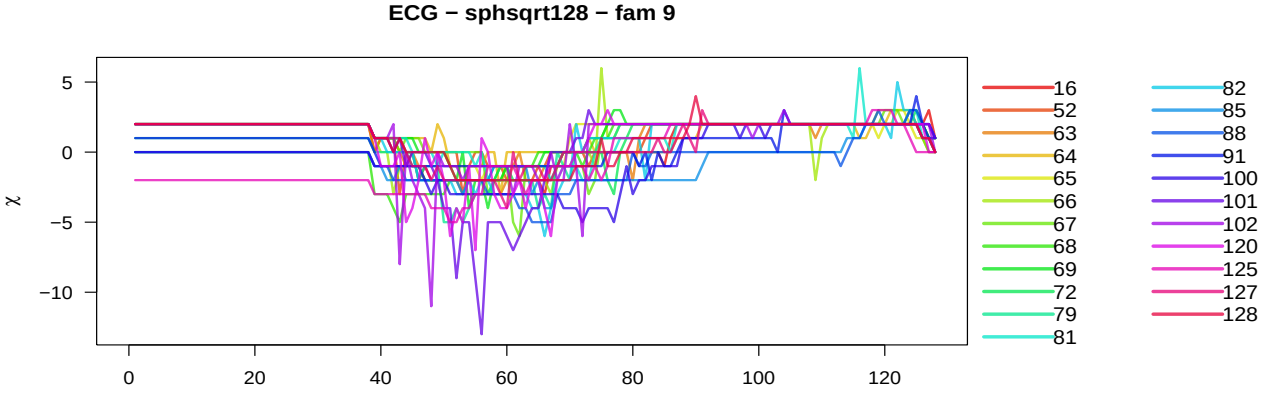
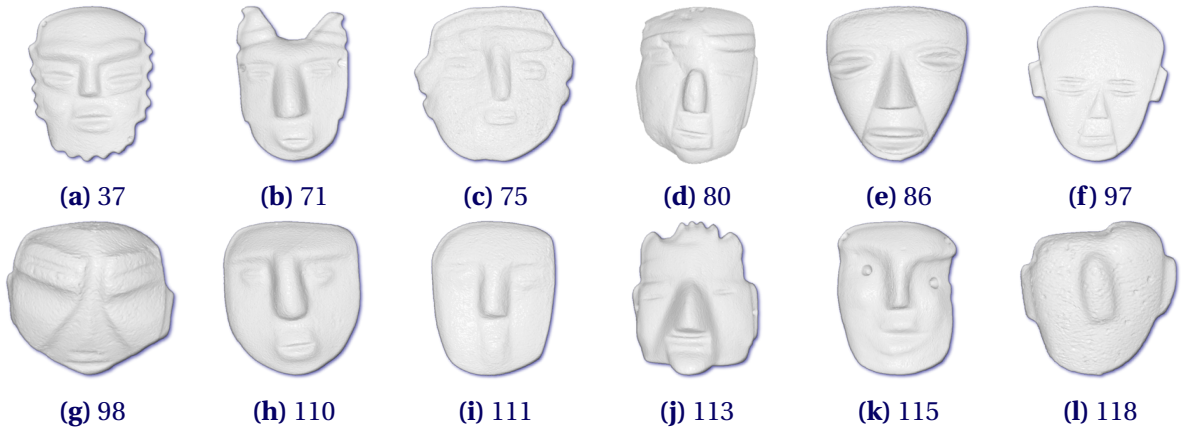


Figure 5.50: ECGs of newly assigned Set 09 based on spherical sweeps f and $T = 32$

Finally, once again, one of the untrained masks from Set 02, another one untrained from 05 were assigned to 09. Additionally, 14 masks from 10 are assorted in Set 09, Masks 16, 72, 102 and 127 are assigned to 09 once again. As seen in Figure 5.50, there is some pattern followed by these family. The masks also share some common physical features with the training items in 09. For instance, a clearly defined T-shaped nose and eyebrows except for masks 52, 91 and 120. All of the masks' eyes are horizontally carved in a squint with very prominent eyelids and eyebrows, except for masks 82, 85, 91 and 120.



(a) 37 **(b)** 71 **(c)** 75 **(d)** 80 **(e)** 86 **(f)** 97

(g) 98 **(h)** 110 **(i)** 111 **(j)** 113 **(k)** 115 **(l)** 118

Figure 5.51: Masks unassigned to any set after running 72 polynomial SVMs

There were also 12 masks with undecided assignment (9.3% of the total dataset), as several of the different SVM tests placed them into different families. Masks 23, 71, 86, 98, 110 and 118 were once again given an uncertain labeling. These can be seen in Figure 5.51.

Masks 71, 75 and 110 have common features with Set `09`: a T-shaped nose and eyebrows, with squinted eyes and thick eyelids and lips. However, mask 71 also has horn-like appendages which might explain the confusion between being labeled both as `04` and `09`. Mask 75 has a very round frame, which might confuse the SVM procedure between labels `08` and `09`. Mask 110 is confused between labels `05` and `09`. Mask 111 was confused between labels `03` and `05`, although masks 110 and 111 look very much alike.

Using $T = 64$ thresholds and $M_h = \sqrt{2}$

When a smaller number of thresholds is used, the SVM procedure assigns all the training items to their respective groups during the test phase. Moreover, most of the test masks are assigned to the same Sets they were assigned during the $T = 128$ procedure above. The procedure was deemed inconclusive for 19 masks (14.8% of the total dataset), 7 masks more that before with $T = 128$. Other changes observed were as follows.

- Sets `03`, `05` and `09` were assigned the same number of masks. Most of these masks are the same from the $T = 128$ with two or three interchanges when compared to the previous procedure.

- Set 07 reports only two additional items, four items less than before. It is more sensible to suggest 2 additional items or less, as no other mask in Set 10 shares strong resemblance with the training items in 07.

Overall, the classification is very similar to the one suggested with $T = 128$ thresholds above. This fact in turn suggests that it may not be necessary to use a large number of thresholds, as the ECG is able to pick up basic differences with few threshold values. The use of small T values in turn lessens negative effects due to high dimensionality and concentration of measure.

Using $T = 64$ thresholds and $M_h = 1$

Based on the SVM procedure run previously with $T = 64$ and $M_h = \sqrt{2}$, it is suggested that $T = 64$ is good enough to pick up relevant features of the masks. Thus we will work with $T = 64$ onwards and rather tune the M_h bound value. When a smaller number of thresholds and a tighter M_h bound are used, the SVM procedure is inconclusive for 11 masks (8.6% of the total dataset), which is a better result than before. However, it stands out that Set 05 now reported an addition of 24 masks, making it the largest family with 37 items in total. Although the vectors have a lower dimension and are supposedly more detailed than with the $T = 128, M_h = \sqrt{2}$ procedure, the ECGs seem to cluster around Set 05. This unusual clustering of masks suggests that we should not have as much confidence with these results as in the previous procedures.

Surprisingly, Set 02 now reports only five additional items as opposed to the six items as in the previous procedures. Item 103 was now excluded and deemed unidentified and confused between Sets 04 and 09. Mask 103 was the most elongated mask of all the masks with holes for eyes or mouth, which might explain the discrepancy. This discrepancy might also deter us to trust the results obtained with the $M_g = 1$ bound.

Using $T = 64$ thresholds and $M_h = 2$

When a smaller number of thresholds and a softer M_h bound are used, it is expected that ECG will convey only the a rough picture of the objects. All but one of the training items were classified in their respective set during the test phase, indicating the the

rough ECGs are still good enough. It is noted that the SVM procedure is inconclusive for 22 masks (17.1% of the total dataset), 10 masks more than in the $T = 128, M_h = \sqrt{2}$ case, which might provide evidence of such rough information. All the comparisons refer to the results obtained with $T = 128, M_h = \sqrt{2}$. Observation made from the procedure results are as follows:

- Set `02` reported the same additional 6 masks as in previous procedures, which suggests that ECGs with rough M_h bounds still convey sufficient information regarding holes in the masks' eyes.
- Set `03`, `05` and `09` report a sensible number of extra assignments and items. At least half of the additional items assigned to each of these sets had been considered before in the $T = 128$ case.
- Set `07` actually missed one of its training items, mask 58. This mask was classified as `07` only 60 times in total. That is, 83.3% of the times rather than the required minimum of 85%.
- Set `08` reports 7 additional items, while it only has three training items. Moreover, none of the suggested new items share any physical feature with the training items.

Overall, the $M_h = 2$ provides a sensible classification, although the number of unassigned items and the misclassification mentioned in Set `07` might suggest that a larger M_h bound will result in a dubious classification.

5.8 Results with unsupervised SVMs

As it was described in Chapter 4 (Algorithm 3), the (i, j)-the Furthest Hyperplane Problem assumes that the two different training vectors $\mathbf{x}_i, \mathbf{x}_j$ define the two possible different families. Remember that the FHP algorithm assumes that the origin is at midpoint $\mathbf{m}_{i,j}$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ and that all the training data has Euclidean norm less than 1. The FHP returns an optimal hyperplane normal vector $\mathbf{w}_{i,j}$ which was afterwards unscaled by a factor $s_{i,j}$, defining thus an unscaled margin $\bar{\theta}_{i,j}$. That is

$$\bar{\theta}_{i,j} := \min_{1 \leq k \leq n} |\langle \mathbf{w}_{i,j}, \mathbf{x}_k \rangle| \, s_{i,j}. \tag{5.8.1}$$

Remember that the optimal unscaled margin of the MMP procedure is the maximum of all the margins $\bar{\theta}_{i,j}$, that is,

$$\bar{\Theta} := \max_{1\leq i<j\leq n} \bar{\theta}_{i,j}. \tag{5.8.2}$$

During the computation of $\bar{\theta}_{i,j}$ it was also computed the sample mean $\mu_{i,j}$ and sample variance $\sigma_{i,j}$ of the distances $\{|\langle \mathbf{w}_{i,j}, \mathbf{x}_k\rangle|\}_{k=1}^n$. The sample mean and sample variance for all the means and variances were computed as well. That is, the following values were computed

$$\mu_{i,j} := \frac{1}{n}\sum_{1\leq k\leq n} |\langle \mathbf{w}_{i,j}, \mathbf{x}_k\rangle|\, s_{i,j} \quad , \quad \sigma_{i,j}^2 := \frac{1}{n}\sum_{1\leq k\leq n} (|\langle \mathbf{w}_{i,j}, \mathbf{x}_k\rangle| s_{i,j} - \mu_{i,j})^2 \tag{5.8.3a}$$

$$M(\mu) := \binom{n}{2}^{-1} \sum_{1\leq i<j\leq n} \mu_{i,j} \quad , \quad \Sigma(\mu)^2 := \binom{n}{2}^{-1} \sum_{1\leq i<j\leq n} (\mu_{i,j} - M(\mu))^2 \tag{5.8.3b}$$

$$M(\sigma^2) := \binom{n}{2}^{-1} \sum_{1\leq i<j\leq n} \sigma_{i,j}^2 \quad , \quad \Sigma(\sigma^2)^2 := \binom{n}{2}^{-1} \sum_{1\leq i<j\leq n} (\sigma_{i,j}^2 - M(\sigma^2))^2. \tag{5.8.3c}$$

We will denote by $\hat{i}$ and $\hat{j}$ the pair of indices i, j which yield the optimal unscaled margin $\bar{\Theta}$. We will refer to $\mu(\bar{\Theta})$ and $\sigma^2(\bar{\Theta})$ as the mean distance and variance from every data point $\mathbf{x}_i$ respectively to the optimal hyperplane with maximum margin $\bar{\Theta}$. That is,

$$\bar{\Theta} = \bar{\theta}_{\hat{i},\hat{j}}, \quad \mu(\bar{\Theta}) := \mu_{\hat{i},\hat{j}}, \quad \sigma^2(\bar{\Theta}) := \sigma^2_{\hat{i},\hat{j}}. \tag{5.8.4}$$

If $\sigma^2(\bar{\Theta})$ is small with respect to $\mu(\bar{\Theta})$ might suggest that all the data points are roughly at the same distance from the optimal hyperplane. This would imply that all data points lie within a spherical shell, suggesting effects of high-dimensionality as shown by (4.6.6).

As mentioned above in Section 5.5, the methodology consisted of evaluating ECGs from two of the eight different training sets. Given two fixed families, our main aim was to recover the members of each family based on an unsupervised approach. The four benchmark families considered for this procedure are detailed in Table 5.1.

Sets `06`, `07` and `08` were not considered as each of them has less than 4 items in total. Set `10` was not considered as there is ongoing discussion with respect the best way to label its items.

As stated earlier, the computation of singular values is a tolling computation dependent on the number of dimensions, It was implemented then an standard Principal

EXP.	FAMILIES TO COMPARE	1ST NO. ITEMS	2ND NO. ITEMS
a.	Set 02 vs. Set 05	24	17
b.	Set 02 vs. Set 09	24	7
c.	Set 03 vs. Set 04	6	6
d.	Set 04 vs. Set 05	6	17

Table 5.1: Benchmark comparisons for the MMP procedure

Component Analysis procedure based on Singular Value Decomposition (cf. [Shl14]). The PCA technique was used to reduce the dimension of the ECGs to 6, 8, 10 and 12 dimensions before running the MMP procedure as suggested in Algorithm 3.

5.8.1 Set 02 vs Set 05

Planar sweeps

The ECGs used were obtained with $T = 32$ thresholds for each of the planar sweep functions described in (5.3.3). A $M_\pi = 2$ bound was considered as well. Even though the variances in (5.8.3) were large, the MMP procedure would cluster every mask but one as a unique family. Mask 40 from Set 05 (refer to Figure A.4) would be constantly excluded from the rest regardless of any reduction in dimension. This failure suggests that in an unsupervised setting, the planar sweeps provide little information to distinguish masks with holes apart from masks without holes.

Cylindrical sweeps

First the ECGs used were obtained with $T = 32$ thresholds for each of the cylindrical sweep functions described in (5.3.4). A $M_g = \sqrt{2}$ was considered as well. Second, the procedure was run considering ECGs computed with $T = 32$ thresholds and a tighter bound $M_g = 1$. The ECGs were reduced to 6 and 12 dimensions prior to start the MMP procedure. In all cases, the optimal hyperplane would split the data in two disjoint subsets

$$S_1 := \{1, 2, \ldots, 24\} \setminus \{16, 23\}, \quad S_2 = \{37, 38 \ldots, 53\} \cup \{16, 23\}. \tag{5.8.5}$$

Observe that masks 16 and 23 are the only masks in Set 02 which have no holes for their eyes or mouth. Refer to Figure A.1. It is also interesting that mask 46 in Set 05 was not considered in S_1, even though it has a hole on the left side of its forehead. Thus the cylindrical sweeps suggest that are sensitive enough to differentiate if a mask has holes in its eyes or mouth. This differentiation is possible even after aggressive dimension reduction, suggesting that the distinction can be robust.

The details for each case regarding the optimal unscaled margin (5.8.2) of the optimal dividing hyperplane and its mean distance from each point and its variance (5.8.4) are as follows:

M_g	dim	$\bar{\Theta}$	$\hat{i}$	$\hat{j}$	$\mu(\bar{\Theta})$	$\sqrt{\sigma^2(\bar{\Theta})}$
$\sqrt{2}$	6	9.8	24	47	23.9	6.3
$\sqrt{2}$	12	10.3	11	43	23.5	6.1
1	6	9.4	15	46	23.2	6.2
1	12	9.9	15	46	22.4	6.1

Data in the $M_g = \sqrt{2}$ case seemed slightly more separated than the $M_g = 1$ case, as the margins, mean values and variances were slightly larger.

Spherical sweeps

First the ECGs used were obtained with $T = 128$ thresholds and $M_g = \sqrt{2}$ for each of the cylindrical sweep functions described in (5.3.5). Secondly, the procedure was run considering ECGs computed with $T = 64$ thresholds and a tighter bound $M_g = 1$. Prior to the MMP procedure, the ECGs dimension was reduced to $6, 8, 10$ and 12 for each case. For every case, the procedure split the training set in the disjoint sets S_1 and S_2 mentioned in (5.8.5). Thus the spherical sweeps prove to be a sensitive choice as well to distinguish holes in the masks' eyes.

The details for each case regarding the optimal unscaled margin (5.8.2) of the optimal dividing hyperplane and its mean distance from each point and its variance (5.8.4) are as follows.

M_g	dim	$\bar{\Theta}$	$\hat{i}$	$\hat{j}$	$\mu(\bar{\Theta})$	$\sqrt{\sigma^2(\bar{\Theta})}$
$\sqrt{2}$	6	10.3	3	47	27.3	8.2
$\sqrt{2}$	12	10.8	6	16	26.7	8.8
1	6	8.2	15	46	18.5	6.0
1	12	8.7	15	46	18.1	5.2

As in the cylindrical sweeps ECGs above, large means and variances when $T = 128$ and $M_g = \sqrt{2}$ suggest that the ECGs obtained through spherical filtering are spread out. This in turn makes the distinction between Sets 02 and 05 clear to pick up. Observe that both the mean distance and variances when $M_g = 1$ decreases. This might suggest that ECGs are closer together in this case, even when it is expected that such ECGs encode more detailed information on the topological changes of each object.

5.8.2 Set 02 vs 09

Planar sweeps

As in the previous case involving the planar filters (5.3.3) and $T = 32$ thresholds, the MMP procedure couldn't find any significant differences between sets 02 and 09. Algorithm 3 clustered all the masks but one in a large cluster. Mask 17 from Set 02 (refer to Figure A.1) would be constantly excluded from the rest of the mask. This result was observed even when the ECGs were reduced to 6 and 12 dimensions via PCA. Thus we may confirm that the planar sweeps (5.3.3) are not sensitive enough to even tell apart masks with holes from masks without holes.

Cylindrical sweeps

First, as in the previous test, we consider filters (5.3.4) with $T = 32$ thresholds and $M_g = \sqrt{2}$. Second, we compute the ECGs with $T = 32$ and $M_g = 1$. Via PCA, we reduce the dimension of the ECGs to 6 and 12 prior to run the Algorithm 3. The MMP procedure was able to distinguish masks with holes and masks with no holes. This distinction is reflected by the partition of data in two disjoint sets

$$S_1 := \{1, 2, \ldots, 24\} \setminus \{16, 23\}, \quad S_3 = \{63, \ldots, 69\} \cup \{16, 23\}. \tag{5.8.6}$$

Thus the cylindrical sweeps are again sensitive enough to distinguish holes in masks' eyes despite aggressive dimension reduction. The details for each case regarding the optimal unscaled margin (5.8.2) of the optimal dividing hyperplane and its mean distance from each point and its variance (5.8.4) are as follows.

M_g	dim	$\bar{\Theta}$	$\hat{i}$	$\hat{j}$	$\mu(\bar{\Theta})$	$\sqrt{\sigma^2(\bar{\Theta})}$
$\sqrt{2}$	6	11.4	15	42	24.1	7.1
$\sqrt{2}$	12	11.7	2	39	23.3	6.6
1	6	11.3	15	42	24.3	7.1
1	12	11.3	22	23	24.2	6.9

The mean distance, variance and optimal margin values are similar in all cases, which suggests that data looks very similar in both cases, $M_g = \sqrt{2}$ and $M_g = 1$. The large optimal margin and large mean distance value for all cases suggest that the two compared families lie clearly separated.

Spherical sweeps

Just as in the previous case, first we consider spherical filters (5.3.5) with $T = 32$ thresholds and $M_h = \sqrt{2}$. Second we consider ECGs computed with $T = 64$ and $M_h = 1$. The ECGs are reduced to 6 and 12 dimensions prior to the Algorithm 3. For all cases, the computed optimal hyperplane separated the data in disjoint sets S_1 and S_3 as in (5.8.6).

Thus the spherical sweeps are again sensitive enough to distinguish holes in masks' eyes. The details for each case regarding the optimal unscaled margin (5.8.2) of the optimal dividing hyperplane and its mean distance from each point and its variance (5.8.4) are as follows.

M_g	dim	$\bar{\Theta}$	$\hat{i}$	$\hat{j}$	$\mu(\bar{\Theta})$	$\sqrt{\sigma^2(\bar{\Theta})}$
$\sqrt{2}$	6	13.9	22	37	29.1	9.2
$\sqrt{2}$	12	14.2	22	37	29.0	9.1
1	6	10.7	12	23	19.6	6.2
1	12	11.0	12	23	19.6	6.1

The larger margins, mean distances and variances in the first case, $M_g = \sqrt{2}$ indicate that data is much more spread out and separable than in the second case, $M_g = 1$.

5.8.3 Set 03 vs 04

The MMP failed to recognize the difference between training items in Sets 03 and 04. The planar and cylindrical sweeps filters used in the comparisons 02 vs. 09 above grouped all the masks but one in a single large cluster. The mask excluded in every case was mask 30 from Set 03 (refer to Figure A.2.) The spherical sweeps also failed to split the two Sets and clustered all but one mask in one single large group as well. The excluded mask in this case was mask 32 from 04 (refer to Figure A.3).

The mean distances and variances, as in (5.8.3b) and (5.8.3c), for the several filters used are as follows:

g	M	dim	$\bar{\Theta}$	$M(\mu)$	$\sqrt{\Sigma(\mu)}$	$M(\sigma^2)$	$\sqrt{\Sigma(\sigma^2)}$
planar	2	6	5.4	5.1	1.8	1.3	1.6
planar	2	12	5.4	5.2	1.6	0.9	0.4
cylinder	$\sqrt{2}$	6	7.9	7.4	2.5	0.7	0.5
cylinder	$\sqrt{2}$	12	8.2	8.2	1.9	0.6	0.3
cylinder	1	6	7.6	8.0	1.3	0.8	0.3
cylinder	1	12	7.8	7.2	2.0	2.0	1.5

It is interesting the fact that the optimal margin Θ is close to the sample mean of mean distances. Moreover, the fact that the sample standard deviation is relatively small suggests that all the points are on average at the same distance from every splitting hyperplane. This last observation is also supported by the small values of mean variances and the standard deviation from sample variances. In other words, the clustering might be caused by high dimensionality effects as mentioned in Section 4.6. Observe that the results were computed after the ECGs had been reduced to 6 and 12 dimensions. Thus the MMP procedure proves to be unfit to distinguish finer details such as larger noses or triangular frames.

5.8.4 Set 04 vs 05

The MMP failed to recognize any significant difference between these two training sets. Just as above, the MMP clustered all but one mask into a single large group. The excluded mask depended on the filter function. For planar filters, mask 40 was excluded.

For cylindrical filters, mask 46 was excluded. Finally, for spherical filters, mask 47 was excluded.

Just as in the previous case, when observing the mean average distances and variances, as in (5.8.3b) and (5.8.3c), it was observed that the optimal margin $\bar{\Theta}$ is very close to $M(\mu)$. The mean distance variance $\Sigma(\mu)$ was small as well. This suggests that the ECGs are affected by high dimensionality. This negative effects are still present when the dimension of the ECGs is lowered via PCA.

5.9 Conclusions and future directions

As discussed in the Section 3.6, the assignments obtained from a SVM could be due to two reasons

1. The ECG of the mask suggests that the given mask is similar to a certain group and ought to be assigned to such group.
2. The ECG of the mask is very distant from the rest of training ECGs and the SVM picks the least distant group.

Part of the difficulties of the project is to determine which case occurred for each assignment. A closer inspection to ECGs could perhaps help to explain which of those cases occurred for each classification. For instance, the ECG of mask 110 yielded from planar sweeps seen in Figure 5.14 has a very prominent peak near the last third. The peak sets this ECG apart from the other two ECGs of training items in Set 06. The fact that mask 110 does not resemble the training items could be an example of the SVM classifying based on the "least distant" set rather than the "closest" set. This reasoning might explain the classification of certain items in Set 10 that do not resemble any of the training sets, as would be the case of masks 92 or 98.

The Euler Characteristic, as shown in Chapter 1, is a summary of the Betti numbers $\beta_q(K)$ of a simplicial complex K. Remember that the Betti numbers represent the number of homologically different q-dimensional holes of a complex. Thus it makes sense that the ECG is powerful descriptor to tell apart if a mask has holes in its eyes or mouth. During both the supervised and unsupervised SVM procedures, the ECGs consistently

assigned all the masks with holes in their eyes and mouth in Set 02. This distinction was possible even when the ECGs were high-dimensional vectors and despite the low number of training items.

A more detailed study of the ECGs may reveal further geometrical information of each mask. For instance, a closer inspection of the graphs in Figure 5.25 versus graphs is Figure 5.27 might reveal where and how exactly does the ECG tell apart a large nose. Ideally, we would like to be able to distinguish information about the masks' frames and thickness based solely on their ECGs. This in turn might help to understand better the ECG as a topological descriptor.

Overall, with the right parameters, some of the supervised SVM procedures produced some sensible results. This is surprising considering the low number of training masks and the high number of different possible labels. Apart from the masks with holes, it was observed that some non-training masks were consistently classified in the same set, even when the SVM procedures took different filter functions and parameters. For instance, mask 105 was consistently labeled as Set 05 in several tests. It could be proposed that if a mask is consistently assigned to the same set, then it should be considered as part of the training set. The SVMs are run again and the process could thus be reiterated. This dynamical training set might help to reduce the number of non-training items for the supervised SVM procedures.

The low number of training items however presents important difficulties to validate the results with several standard statistical techniques such as cross-reference validation. A possible approach would be a *jackknife resampling*. That is, a training point is removed and the SVM is run with the rest of the data. Later we verify if the removed point is assigned to its corresponding family. This procedure is repeated for every training point. Unfortunately, this method is computationally expensive, as it requires to calculate a large number of SVMs. Moreover, the procedure we used depends on computing 72 SVMs with different parameters each.

As stated previously in Chapter 2, there are several mathematical goals to accomplish. Mainly, we would like to establish some results regarding regularity of the ECGs. We would like to defining more precisely how close the ECGs must be to each other to then imply that they come from similar simplicial complexes. Several mathematical

questions arises, such as defining "close" and "similar" in the previous statement. We must also study if we require additional conditions on the simplicial complexes to establish a certain regularity result. The classifications presented above suggests that it is possible to attain such regularity results.

Remember that the computation of the ECG of a fixed complex K is linear on the number of vertices. Despite that each mesh model had on average $70,000$ different vertices, each ECG could be computed instantly without requiring any special hardware. Moreover, more traditional techniques of topological data analysis, such as the computation of persistence diagrams, are computationally quite expensive. The ECG proves a new technique to extract topological information from very large objects in a quite feasible manner. This efficiency may be exploited in the future in real-time applications such as augmented reality.

The curse of dimensionality most likely has a negative effect on some proposed sortings as it was pointed with the ECGs yielded from planar sweeps filters and $T = 128$ thresholds. This curse was shown as most of the masks would be clustered in a single large set. This curse seems to be specially present in the unsupervised setting. None of the unsupervised procedures managed to tell apart the triangular frames and large noses in Set 04 from the oblong frames and wide noses in Set 03. This failure is most likely due to the lack of training data, although more experimentation is necessary.

The use of tighter M bounds (as in (5.3.2)) was proposed as a partial solution to this dimensionality conundrum. Tighter M bounds imply shorter distances between threshold values. Thus, the ECGs would be able to capture more detailed changes of the Euler Characteristic without increasing its dimension T. In the case of cylindrical and spherical sweeps, for a fixed number of thresholds, the tighter bound $M = \sqrt{2}$ proved to yield sensible results.

However, further exploration is needed on the particular effects of tightening or relaxing these M bounds. For instance, when the spherical filtration was considered the tight bound $M_h = 1$, a considerable number of masks clustered in Set 05. The unsupervised SVM procedures also reveal that the ECGs computed with spherical filters (5.3.5) and $M_h = 1$ are actually closer together. This assertion is suggested from the fact that the mean distance and variance from each point to the optimal separation hyperplane is

smaller than in the $M_h = \sqrt{2}$ case. In other words, the tightest possible M bound might not be the best choice.

A second problem on tightening M bounds is the fact that they are highly dependent on the dataset. For instance, $M_g = 2$ and $M_h = 3$ are bounds that work for every complex embedded in the $[-1, 1]^3$ cube, as it was the case with the 128 meshes dataset. Tighter M_g and M_h bounds were found through direct experimentation, and it is not clear that they will work if additional models are considered. Thus tightening M bounds may risk expandability of some results to larger data sets.

Filtrations based on curvature and (Euclidean) distance to the k-th nearest neighbor were also considered as suggested in [RW14] to analyze archaeological data. The ECGs yielded failed to produce sensible results, as most of the masks would be clustered in a single large group. Our hypothesis to explain such failure is the fact that the 3D mesh models are extremely details. Each model has on average $70,000$ vertices. Thus, locally, the model looks the same everywhere as the vertices are spread out homogeneously throughout the whole model. This homogeneity in turn produces small differences based on curvatures or nearest neighbors which are difficult to pick up by our proposed method. An attempt was made to lower the number of vertices of each mask to less than 5000 while still preserving its basic shape and topology. This vertex reduction was carried out using *MeshLab v1.3.2* [Cig+08] and its inbuilt *Quadratic Edge Collapse Decimation* tool [PTC10]. Even after aggressively reducing the vertices of each model, the ECGs still failed to produce sensible results.

In conclusion, more experimentation is needed. The results above show that the proposed classification depends heavily on the fixed filter function and its M bound. Further discussion with the archaeological community is necessary to determine the best way to tune the parameters.

Bibliography

[Arm83] M.A. Armstrong. *Basic Topology*. New York: Springer-Verlag, 1983.

[Bar05] Alexander Barvinok. *Measure Concentration*. Lecture Notes. 2005. URL: http://www.math.lsa.umich.edu/~barvinok/total710.pdf (visited on 04/02/2018).

[BHK17] Avrim Blum, John Hopcroft, and Ravi Kannan. *Foundations of Data Science*. June 2017. URL: https://www.microsoft.com/en-us/research/publication/foundations-of-data-science-2/.

[Bur98] Christopher J.C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Mining and Knowledge Discovery* 2.2 (June 1998), pp. 121–167. ISSN: 1573-756X. DOI: 10.1023/A:1009715923555. URL: https://doi.org/10.1023/A:1009715923555.

[CL11] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A Library for Support Vector Machines". In: *ACM Trans. Intell. Syst. Technol.* 2.3 (May 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27. ISSN: 2157-6904. DOI: 10.1145/1961189.1961199. URL: http://doi.acm.org/10.1145/1961189.1961199.

[Cig+08] Paolo Cignoni et al. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008, pp. 129–136. ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[CEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. “Stability of Persistence Diagrams”. In: *Discrete & Computational Geometry* 37.1 (Jan. 2007), pp. 103–120. ISSN: 1432-0444. DOI: 10.1007/s00454-006-1276-5. URL: https://doi.org/10.1007/s00454-006-1276-5.

[Cov61] Miguel Covarrubias. *Arte indígena de México y Centroamérica (Spanish)*. 1st ed. México: Universidad Nacional Autónoma de México (UNAM), 1961.

[Fle00] Roger Fletcher. *Practical Methods of Optimization*. 2nd ed. Chichester: Wiley-Interscience Publication, 2000.

[Gal+17] Mark Galassi et al. *GNU Scientific Library*. 2.4. June 2017. URL: https://www.gnu.org/software/gsl/.

[Gay67] Carlo T. E. Gay. *Mezcala stone sculpture; the human figure*. English. New York Graphic Society Greenwich, Conn, 1967.

[Hat02] Allen Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002. ISBN: 9780521795401. URL: https://www.math.cornell.edu/~hatcher/AT/ATpage.html.

[HL02] Chih-Wei Hsu and Chih-Jen Lin. “A Comparison of Methods for Multiclass Support Vector Machines”. In: *IEEE Transactions on Neural Networks* 13.2 (Mar. 2002), pp. 415–425. ISSN: 1045-9227. DOI: 10.1109/72.991427. URL: http://dx.doi.org/10.1109/72.991427.

[JRM12] Diego Jiménez Badillo, Salvador Ruíz Correa, and Omar Méndoza Montoya. *Analyzing formal features of archaeological artefacts through the application of Spectral Clustering*. Seen online 2018-03-20. Lecture of the Digital Classicist Seminar. Deutsches Archäologisches Institut, Berlin, Tuesday, 06 November 2012. URL: http://hdl.handle.net/11858/00-1780-0000-000B-216A-E.

[Kar+12] Zohar Karnin et al. “Unsupervised SVMs: On the Complexity of the Furthest Hyperplane Problem”. In: *Proceedings of the 25th Annual Conference on Learning Theory*. Ed. by Shie Mannor, Nathan Srebro, and Robert C. Williamson. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: PMLR, 25–27 Jun 2012, pp. 2.1–2.17. URL: http://proceedings.mlr.press/v23/karnin12.html.

[KT51] H. W. Kuhn and A. W. Tucker. “Nonlinear Programming”. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA: University of California Press, 1951, pp. 481–492. URL: https://projecteuclid.org/euclid.bsmsp/1200500249.

[Lux07] Ulrike von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17.4 (Dec. 2007), pp. 395–416. ISSN: 1573-1375. DOI: 10.1007/s11222-007-9033-z. URL: https://doi.org/10.1007/s11222-007-9033-z.

[Mey+17] David Meyer et al. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-8. 2017. URL: https://CRAN.R-project.org/package=e1071.

[OG86] Bertina Olmedo Vera and Carlos Javier González. “Presencia del estilo mezcala en el Templo Mayor : una clasificación de piezas antropomorfas (Spanish)”. Bachelor Thesis in Archaeology. México: Escuela Nacional de Antropología e Historia, 1986.

[PR16] Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Vol. 152. Cambridge studies in advanced mathematics. Cambridge University Press, 2016. ISBN: 9781107104099.

[PTC10] N. Pietroni, M. Tarini, and P. Cignoni. “Almost Isometric Mesh Parameterization through Abstract Domains”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.4 (July 2010), pp. 621–635. ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.96.

[R C18] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2018. URL: https://www.R-project.org/.

[RW14] Eitan Richardson and Michael Werman. “Efficient classification using the Euler characteristic”. In: *Pattern Recognition Letters* 49 (2014), pp. 99–106. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2014.07.001. URL: http://www.sciencedirect.com/science/article/pii/S0167865514002050.

[Roj15] Raúl Rojas. *The Curse of Dimensionality*. Feb. 2015. URL: https://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/tutorials/dimensionality.pdf (visited on 04/02/2018).

[Rot99] Joseph Rotman. *An Introduction to the Theory of Groups*. 4th ed. Vol. 148. Graduate Texts in Mathematics. New York: Springer-Verlag, 1999. ISBN: 9780387942858. URL: https://www.springer.com/us/book/9780387942858.

[RC11] Radu Bogdan Rusu and Steve Cousins. "3D is here: Point Cloud Library (PCL)". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, May 2011.

[Ser73] M. C. Serra Puche. "Intento de seriación en esculturas de Guerrero: cronología del estilo Mezcala [Spanish]". In: *XIII Mesa Redonda (Xalapa, Veracruz)*. Vol. 2. México: Sociedad Mexicana de Antropología, 1973, pp. 337–348.

[Shl14] Jonathon Shlens. *A Tutorial on Principal Component Analysis*. 2014. URL: http://arxiv.org/abs/1404.1100 (visited on 04/02/2018).

[Spr14] Vincent Spruyt. *The Curse of Dimensionality in Classification*. 2014. URL: http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/ (visited on 04/02/2018).

[Wol61] Philip Wolfe. "A duality theorem for non-linear programming". In: *Quarterly of Applied Mathematics* 19 (1961), pp. 239–244.

[PAGE LEFT INTENTIONALLY BLANK]

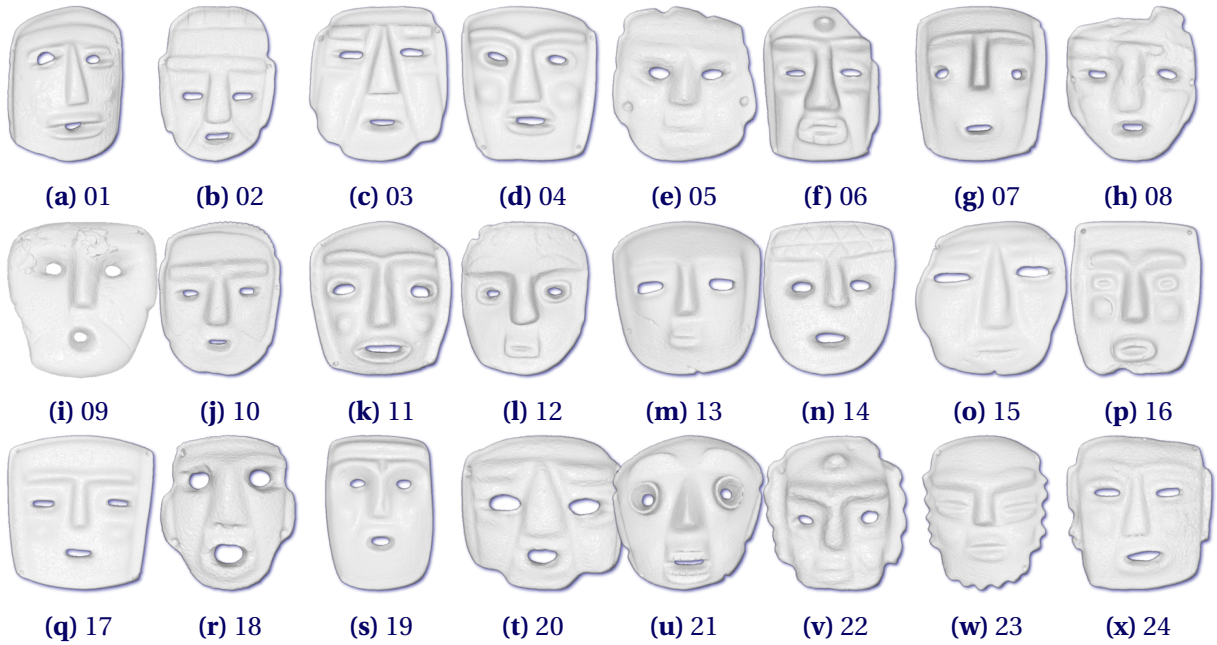
Appendix A

Original dataset

The dataset used consists of 128 digitized 3D meshes of the masks which are part of the 162 masks found at the Sacred Precinct of Tenochtitlán. The 34 remaining masks were not considered as they are heavily damaged, incomplete and possibly unfinished. The data was provided by Diego Jiménez and the National Institute of Anthropology and History (INAH). The meshes of the masks were originally grouped in 9 different sets numbered from 02 to 10. The items from each original set are numbered from 01 to 128.

A.1 Set 02

The main distinctive feature of the set picked by the ECGs are the holes in the eyes or the mouth. 22 items out of 24 do have at least two holes. As they are the only set with holed masks, their ECGs are quite distinctive from the rest. Most of them also have a distinctive T-shape formed by their eyebrows and nose. It is the largest set of the eight well-distinguished set.



(a) 01 **(b)** 02 **(c)** 03 **(d)** 04 **(e)** 05 **(f)** 06 **(g)** 07 **(h)** 08

(i) 09 **(j)** 10 **(k)** 11 **(l)** 12 **(m)** 13 **(n)** 14 **(o)** 15 **(p)** 16

(q) 17 **(r)** 18 **(s)** 19 **(t)** 20 **(u)** 21 **(v)** 22 **(w)** 23 **(x)** 24

Figure A.1: Masks in the original set 02

A.2 Set 03

(a) 25 **(b)** 26 **(c)** 27 **(d)** 28 **(e)** 29 **(f)** 30

Figure A.2: Masks in the original set 03

Masks in Set 03 have a wide, trapezoidal nose; carved slanted eyes with no eyebrows, or even no eyes at all; carved horizontal mouth with clear lips; defined ears and an oblong frame.

A.3 Set 04

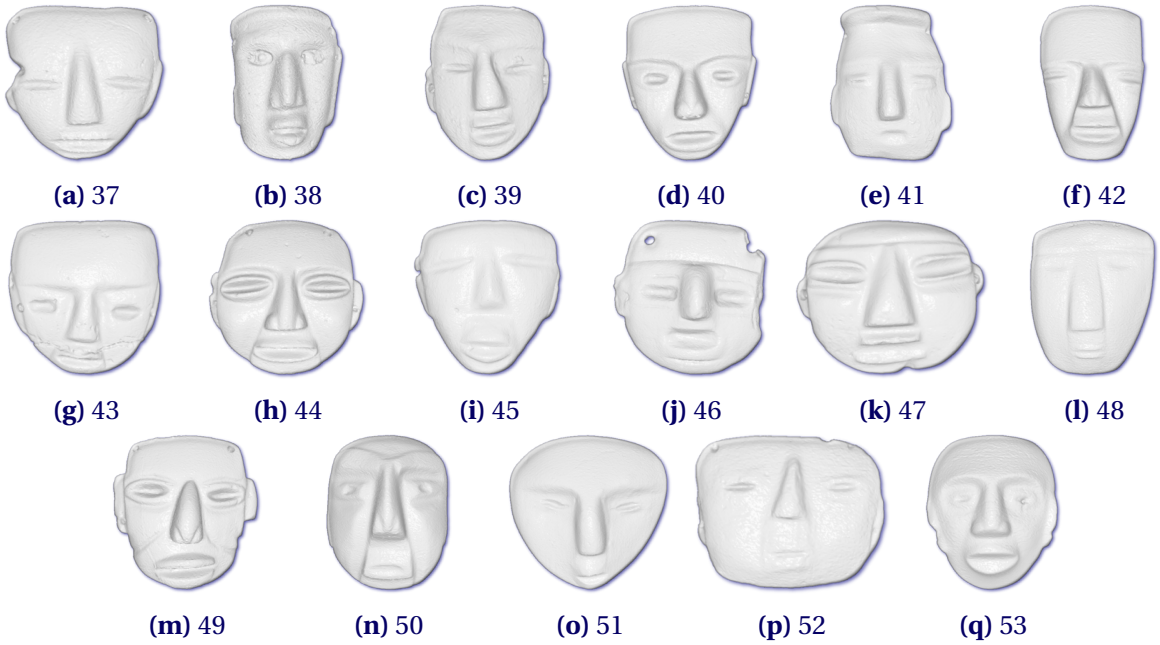
The most distinctive feature for masks in Set 04 is their very prominent, triangular, sharp nose. All of them have carved, slanted eyes with no eyelids as well. Except for mask 32,

all of them have a triangular frame. Except for mask 36, all of them have defined, thick eyebrows.

(a) 31 **(b)** 32 **(c)** 33 **(d)** 34 **(e)** 35 **(f)** 36

Figure A.3: Masks in the original set 04

A.4 Set 05



(a) 37 **(b)** 38 **(c)** 39 **(d)** 40 **(e)** 41 **(f)** 42

(g) 43 **(h)** 44 **(i)** 45 **(j)** 46 **(k)** 47 **(l)** 48

(m) 49 **(n)** 50 **(o)** 51 **(p)** 52 **(q)** 53

Figure A.4: Masks in the original set 05

Masks from Set 05 have a triangular, thin nose. They also have carved, slanted eyes, and most of the masks do have clear eyelids. They do have a carved, horizontal open mouth and defined lips. Most of them have clearly defined ears. Roughly half of the masks do have a triangular, shaped chin.

A.5 Set 06

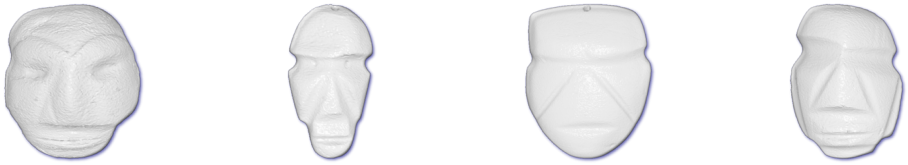
Set 06 is the smallest training set of all the 8 training sets. It just contains two items. Their main feature seems to be their round, bulky frame, with heavy cheeks. They do have a T-shaped nose and eyebrows. Their eyes are carved in a squint, with very prominent eyelids. They do have very prominent lips as well.

(a) 54 **(b)** 55

Figure A.5: Masks in the original set 06

A.6 Set 07

Masks in Set 07 have a rough appearance, as if unfinished. With three clear cuts their triangular nose is defined, and with two additional horizontal wedge cuts their eyes are represented. Except for mask 58, one additional cut represents their horizontal mouth. Except for mask 56, they have an elongated frame.



(a) 56 **(b)** 57 **(c)** 58 **(d)** 59

Figure A.6: Masks in the original set 07

A.7 Set 08

Masks in Set 08 main feature is their triangular nose defined by two deep, straight cuts. Their eyes and mouth are carved horizontally with well defined eyelids and lips. They also have a round frame.

(a) 60 **(b)** 61 **(c)** 62

Figure A.7: Masks in the original set 08

A.8 Set 09

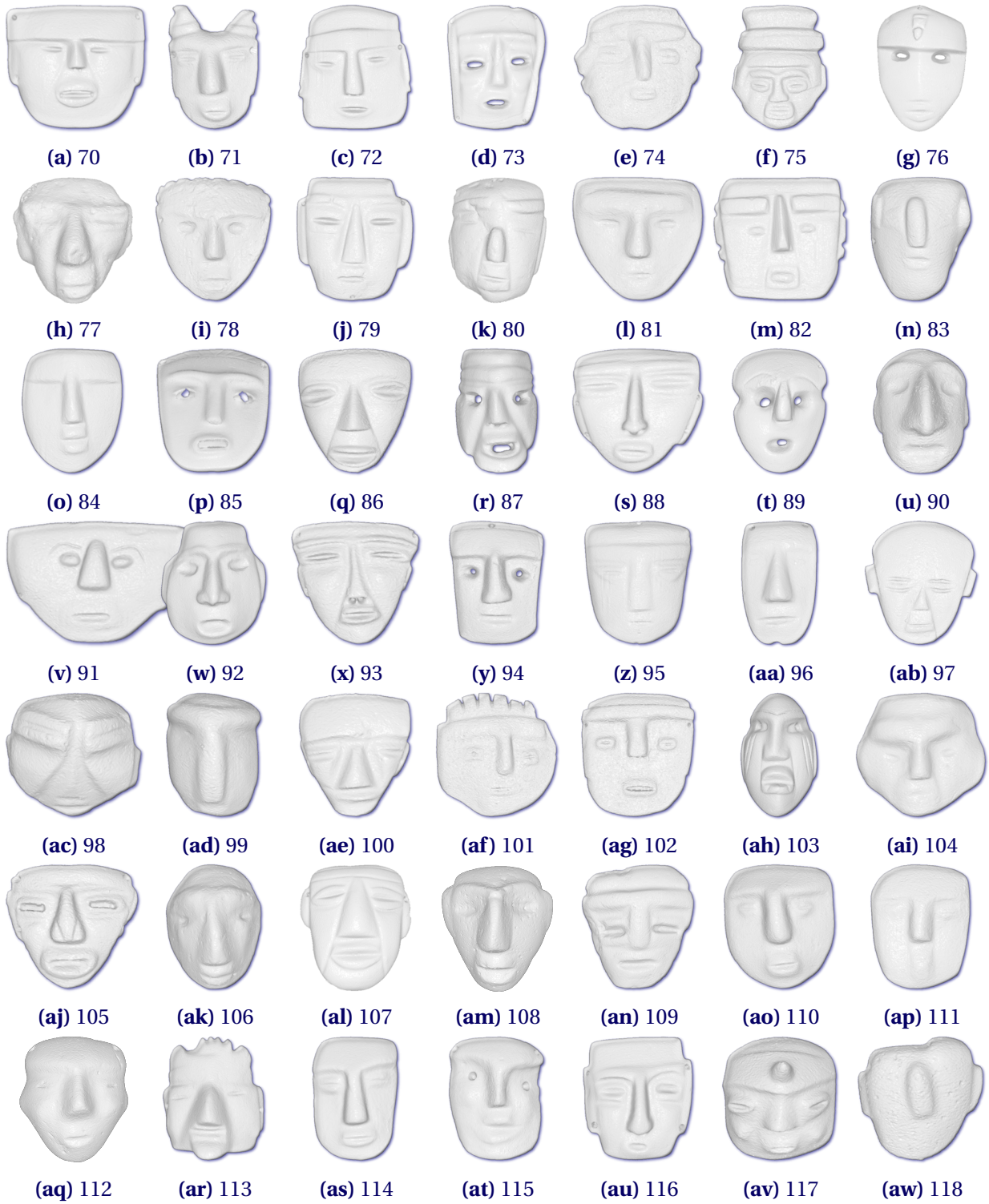
Masks in Set 09 have a prominent T-shaped nose and eyebrows. They do have slitted eyes and mouth as if squinting, with prominent eyelids and lips. Except for mask 65, all of them have square-like frame.

(a) 63 **(b)** 64 **(c)** 65 **(d)** 66 **(e)** 67 **(f)** 68 **(g)** 69

Figure A.8: Masks in the original set 09

A.9 Set 10

The Set 10 consists of masks whose origins are unclear. Further discussion is needed among the archaeological community to establish and distinguish the origin, style, and classification of each of these items. As expected, there is no clear pattern followed by all the 59 masks in the Set. Observe that almost half of the models of pre-Columbian masks are in Set 10. This implies that only half of our total dataset is possible to use as training data for any supervised classification approach.



(a) 70 **(b)** 71 **(c)** 72 **(d)** 73 **(e)** 74 **(f)** 75 **(g)** 76

(h) 77 **(i)** 78 **(j)** 79 **(k)** 80 **(l)** 81 **(m)** 82 **(n)** 83

(o) 84 **(p)** 85 **(q)** 86 **(r)** 87 **(s)** 88 **(t)** 89 **(u)** 90

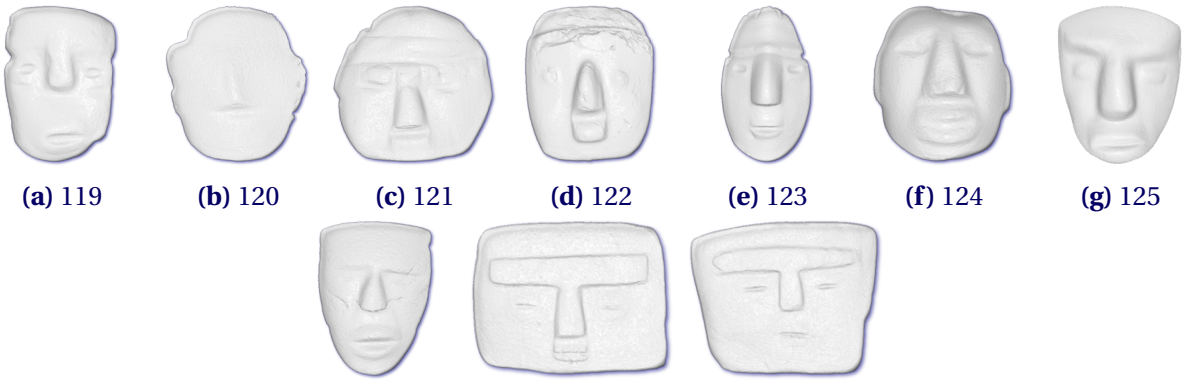
(v) 91 **(w)** 92 **(x)** 93 **(y)** 94 **(z)** 95 **(aa)** 96 **(ab)** 97

(ac) 98 **(ad)** 99 **(ae)** 100 **(af)** 101 **(ag)** 102 **(ah)** 103 **(ai)** 104

(aj) 105 **(ak)** 106 **(al)** 107 **(am)** 108 **(an)** 109 **(ao)** 110 **(ap)** 111

(aq) 112 **(ar)** 113 **(as)** 114 **(at)** 115 **(au)** 116 **(av)** 117 **(aw)** 118

Figure A.9: Mask in the original Set 10



(a) 119 **(b)** 120 **(c)** 121 **(d)** 122 **(e)** 123 **(f)** 124 **(g)** 125

(h) 126 **(i)** 127 **(j)** 128

Figure A.10: Masks in the original set 10