Johannes Voll Kolstø

# Support Vector Machines on Riemannian Manifolds

Master's thesis in applied mathematics and physics
Supervisor: Ronny Bergmann
June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Johannes Voll Kolstø

# Support Vector Machines on Riemannian Manifolds

Master's thesis in applied mathematics and physics
Supervisor: Ronny Bergmann
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Norwegian University of Science and Technology

NTNU

Faculty of Information Technology and Electrical Engineering
Department of Mathematics

# Support Vector Machines on Riemannian Manifolds

Johannes Voll Kolstø

*Supervisor:* Assoc. Prof. Dr. rer. nat. habil. Ronny Bergmann

24.06.2022

**Johannes Voll Kolstø**

*Support Vector Machines*
*on Riemannian Manifolds*,
24.06.2022
Supervisor: Assoc. Prof. Dr. rer. nat. habil. Ronny Bergmann

**Norwegian University of Science and Technology**
Department of Mathematics
Faculty of Information Technology and Electrical Engineering
Trondheim

# Abstract

Support vector machines (SVM's) are useful tools used to perform binary classification of datasets on Hilbert spaces. However, in certain applications such as classifying the hippocampi belonging to people with schizophrenia, or separating brain signals due to specific stimuli, the data do not lie on Hilbert spaces. Instead, it is beneficial to model the data as belonging to generalized surfaces called Riemannian manifolds. In this thesis we present a novel classification model on Riemannian manifolds, inspired by the SVM model, called Distance SVM (DSVM). This model classifies data by a weighted sum of the Riemannian distance between support points, instead of computing their inner product. Variations on the Distance SVM model which produce linear separators on Hilbert spaces are also considered. In addition, we compare the Distance SVM models with three other existing manifold SVM models on real world data from the brain computer interface competition BCI-IV and show that the sparse DSVM models are competitive in test accuracy.

# Sammendrag

Støttevektormaskiner er nyttige verktøy brukt til binærklassifikasjon av datasett på Hilbertrom. I noen applikasjoner derimot, slik som å klassifisere hippocampuser til mennesker med schizofreni eller separare hjernesignaler basert på stimuli, ligger ikke dataen naturlig på et Hilbertrom. I stedet anser vi dataen for å ligge på generaliserte overflater kalt Riemannske mangfoldigheter. Vi presenterer en ny klassifiskasjonsmodell på Riemannske mangfoldigheter, inspirert av støttevektormaskiner, som kalles Avstands-støttevektormaskin. Denne modellen klassifiserer data ved å vekte den Riemannske avstanden til en mengde støttepunkt, i stedet for å beregne indreproduktet. Vi presenterer også variasjoner på denne modellen som produserer lineære skilleplan i Hilbertrom. Til sammenligning tester vi Avstands-støttevektormaskinmodellene og tre eksisterende modeller for klassifisering på Riemannske mangfoldigheter på ekte data fra hjernemaskingrensesnitt-konkurransen BCI-IV, og viser at de glisne Avstands-støttevektormaskinmodellene er konkuransedyktige hva gjelder treffsikkerhet på test-datasettet.

# Acknowledgement

The work done in this thesis would not be possible without the generous help of several people.

First, I would like to thank my supervisor Ronny Bergmann for all the good discussions, advice and feedback during the work on this thesis. I am grateful for your help and guidance with understanding and working on Riemannian manifolds, as it was quite a change of domain that began last autumn. And thank you for encouraging me to contribute to the MaGIC 2022 colloquium by giving a talk on the outlook of this Master's thesis, and for the help in preparing the presentation itself.

I would also like to thank all my study colleagues, for the entertaining lunch conversations, aid with spitballing ideas, and overall for making the time spent writing this thesis enjoyable. Thank you as well to all my friends and the student society in Trondheim, for keeping me engaged and motivated during my time at the Norwegian University of Science and Technology, and for making these last six year truly amazing.

Finally, a special thank you to my parents Stein Dankert and Ingrid for being a constant source of support and encouragement, and for reminding me that student life is about more than just studying.

# Contents

# Introduction 1

Support vector machines have a rich history in classification applications and function estimation problems [1], having been used to for example classify handwritten digits [2] and detect human faces in pictures [3]. However, the theory on support vector machines (SVMs) and their applications all work on Hilbert spaces with a vector space structure.

Some classes of data that we wish to classify are naturally restricted to specific subsets of e.g. $\mathbb{R}^n$, and through the framework of Riemannian geometry [4] we can imbue these subsets with a manifold structure. In some applications the assumption that data points lie on a vector space also ignores intrinsic information about the data, as is the case for Fletcher, Lu, et al. [5], who apply shape analysis to medical imaging seeking to improve the accuracy of medical diagnosis. The authors of [5] represent three-dimensional biological tissue as collections of *medial atoms*, which are points on $\mathbb{R}^4 \times \mathbb{R}^+ \times \mathbb{S}(2) \times \mathbb{S}(2)$, where $\mathbb{S}(2)$ denotes the two-sphere. For medial atoms the idea of adding two medial atoms is not meaningful in a vector space sense, and the data is better understood by considering the geometric structure of $\mathbb{R}^+$ and $\mathbb{S}(2)$. Styner, Lieberman, et al. [6], working in the field of neuroimaging, studied the viability of using medial atom representations of the hippocampus to classify shape abnormalities in schizophrenia. They seek to determine if there is a morphological change in the hippocampus of people with schizophrenia, to classify people at risk for schizophrenia as opposed to healthy structures. And again it would be inappropriate to apply the regular vector space SVM methods as the data domain is not a vector space.

Another field where the data is not always well describes as lying in a vector space is computer vision. Tuzel, Porikli, et al. [7] perform pedestrian detection in images by first constructing covariance feature matrices on a per-pixel basis, and then classify clusters of pixels as either covering a pedestrian or not. These covariance matrix features are guaranteed to be symmetric positive semi-definite matrices, and assuming they're positive definite they can be imbued with a manifold structure. Jayasumana, Hartley, et al. [8] further mention the set of 2D shapes and linear subspaces of $\mathbb{R}^n$ as sets of nonlinear data that are encountered in computer vision classification problems and can be imbued with a Riemannian manifold structure.

## 1.1 Related Work

We will call any classification model on manifolds which is inspired by the classical SVM model a *manifold svm* model. And the first idea which might come to mind when trying to generalize the classical SVM model to manifolds is to choose a reference point $p_{\text{ref}} \in \mathcal{M}$

on the manifold in question, and then map all the training points into its tangent vector space using the logarithmic mapping. Assuming that all the training points are within the injectivity radius of the reference point, we then have a vector space representation of our training points, and we can apply the standard SVM model. Such a tangent vector space based manifold SVM model has been presented by Barachant, Bonnet, et al. [9] and Tuzel, Porikli, et al. [10], who all use the above idea on the manifold of symmetric positive definite matrices. However, the choice of a reference point, or *the best* reference point, is not trivial, and we might lose geometric information by mapping into the tangent vector space and using Euclidean models which are blind to the inherent geometry of the manifold in question.

The second manifold SVM model used by e.g. Jayasumana, Hartley, et al. [11], and Yun, Gu, et al. [12], extends the kernel trick for classical SVM models to Riemannian manifolds with a specific class of Riemannian metrics. By considering the metric space structure of certain Riemannian manifolds they present an adapted Radial Basis Function (RBF) kernel which fulfills the requirements for implicitly mapping to a Hilbert space in which a linear separator can be constructed.

Finally, the third manifold SVM model we'll present works directly on the manifold itself instead of in a tangent vector space, thus avoiding the task of choosing a reference point to represent your training data in. Instead, Sen, Foskey, et al. [13] introduce two control points $c_+$ and $c_-$ on the manifold $\mathcal{M}$, one for each of the classes. Points on the manifold are then classified according to which of the control points they are closest to, and they optimize over $\mathcal{M}^2$ to find the pair of control points which minimizes misclassification of all the training points along with the squared distance between the two control points to promote uniqueness in the set of optimal control points.

## 1.2 Our contribution

In this thesis we present a novel classification model on Riemannian manifolds which does not map the training data into a tangent vector space or introduce new control points in the manifold. Instead, our model classifies points by computing the sign of a weighted sum of the squared distances to a set of support points on the manifold in question. We call this model Distance SVM, as its derivation is closely related to the classical SVM model, but relies on measuring the distance between points instead of their inner product. In addition, we present a variation on Distance SVM (DSVM) called Zero Curvature DSVM (ZCDSVM) which produces linear separators in Hilbert spaces, and in some cases the exact same linear separator as the classical SVM model.

Furthermore, we compare the three existing manifold SVM models mentioned in Section 1.1 with the DSVM models on real world data from the field of brain computer interfaces. Specifically we train and test the models to classify brain activity based on symmetric positive definite (SPD) covariance matrices generated from EEG measurements made available through the BCI-IV competition [39].

## 1.3 Thesis Structure

**Chapter 2, Optimization Theory**

As a preliminary for how to solve for the classical SVM classifier and other models which rely on solving constrained optimization problems, we present relevant concepts and results from constrained optimization theory.

**Chapter 3, Classical Support Vector Machines**

In this chapter we derive the classical SVM model along with how to solve for it. Additionally, we present the extension of the SVM model to non-linear classifiers by use of the kernel trick, and introduce the theory of positive definite functions to answer the question of which mappings are valid kernels.

**Chapter 4, Smooth Riemannian Manifolds**

To work on Riemannian manifolds and understand which kinds of structures and functions are available for generalizing the SVM classifier to manifolds, we give an introduction to manifold theory and parts of the theory of Riemannian manifolds.

**Chapter 5, Existing Models**

Here we explain the three existing manifold SVM models in greater detail, describing their motivations and derivation, and present how to find the classifiers for each model.

**Chapter 6, Distance SVM**

In this chapter we present the Distance SVM model, explaining its derivation and motivation. We also compare how the DSVM classifier looks in vector spaces with the classical SVM linear classifier, to get a deeper understanding of how the DSVM model differs from the classical SVM model.

**Chapter 7, Numerical Experiments**

In this chapter we first compare the resulting DSVM classifiers with the classical SVM classifiers on three toy academic datasets on $\mathbb{R}^2$. Then we compare the DSVM classifiers with the existing manifold SVM classifiers trained on two generated datasets on $\mathbb{S}(2)$. In the last section of this chapter we present a real world dataset and test all the DSVM models and existing manifold SVM on it.

**Chapter 8, Conclusion**

Lastly, we summarize the results of the numerical experiments, and discuss the benefits and drawbacks of the DSVM models as compared to the exist2ing manifold SVM models. And we point towards directions for further research on manifold SVM models.

# Optimization Theory 2

Having defined a model to classify data, we usually find the optimal classifier by solving an optimization problem, with or without constraints. To understand how we solve for the classical support vector machine model we present some general vector space optimization theory results, which are useful for some manifold SVM models as well.

## 2.1 Constrained Optimization

Following the notation of Nocedal and Wright [14], we write a general constrained optimization problem for a lower semi-continuous objective function $F\colon \mathbb{R}^d \to \mathbb{R}$ as

$$\min_{x \in \Omega} \quad F(x), \tag{2.1a}$$

$$\text{s.t.} \quad c_i(x) = 0, \quad i \in \mathcal{E}, \tag{2.1b}$$

$$c_i(x) \geq 0, \quad i \in \mathcal{I}. \tag{2.1c}$$

with constraint functions $c_i\colon \mathbb{R}^d \to \mathbb{R},\ i \in \mathcal{E} \cup \mathcal{I}$ indexed by $\mathcal{E}$ and $\mathcal{I}$ for equality and inequality constraints, respectively. Denoting the domain of a function $F$ by $\mathbf{dom}\, F$ the domain of the optimization problem is

$$\Omega = \left( \bigcap_{i \in \mathcal{E} \cup \mathcal{I}} \mathbf{dom}\, c_i \right) \cap \mathbf{dom}\, F \subseteq \mathbb{R}^d, \tag{2.2}$$

which we assume is non-empty. Given a *feasible point* $x \in \mathbb{R}^d$, i.e. a point satisfying both Eq. (2.1b) and Eq. (2.1c), it is useful to keep track of which constraints are met by equality, and to this end we define the *active set*:

**Definition 2.1** (Active Set [14, Def. 12.2])**.** *The* active set *at any feasible $x$ consists of the equality constraint indices $\mathcal{E}$ as well as the indices in $\mathcal{I}$ for which $c_i(x) = 0$ and is denoted $\mathcal{A}(x)$. That is,*

$$\mathcal{A}(x) = \mathcal{E} \cup \{ i \in \mathcal{I} \mid c_i(x) = 0 \}.$$

We furthermore say that a constraint $c_i$ is *active* at $x$ if $c_i(x) = 0$. Thus, for a feasible point $x$, all the constraints indexed by $\mathcal{A}(x)$ are active.

For unconstrained optimization problems with continuously differentiable objective functions $F(x)$, a necessary condition for a point $x^*$ to be a local minimizer is that the gradient $\nabla F(x^*)$ vanishes [14, Thrm. 2.2]. Analogous results hold for constrained optimization problems, but they depend on the constraint functions and their gradients satisfying certain conditions, or *constraint qualifications*. One such constraint qualification is the reasonably strict Linear Independence Constraint Qualification (LICQ).

**Definition 2.2** (LICQ [14, Def. 12.4]). *Given a point $x \in \Omega$ and the active set $\mathcal{A}(x)$, we say that the* Linear Independence Constraint Qualification (LICQ) *holds if the set of active constraint gradients $\{\nabla c_i(x) \mid i \in \mathcal{A}(x)\}$ is linearly independent.*

A construct which greatly aids in capturing the interplay between the objective function and the constraint functions of an optimization problem like Prob. (2.1) is the *Lagrangian* function defined as

$$\mathcal{L}\colon \mathbb{R}^d \times \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}^{|\mathcal{I}|} \to \mathbb{R}, \quad \mathcal{L}(x, \nu, \lambda) = F(x) - \sum_{i \in \mathcal{E}} \nu_i c_i(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x). \tag{2.3}$$

The variables $\nu$ and $\lambda$ are known as the Lagrange multipliers of their respective constraints, and will later play a central role in the *dual formulation* of optimization problems. Having defined the LICQ, the following theorem presents a number of conditions on the gradients of the objective, constraint function, and Lagrange multipliers in order for a point $x^*$ to be a local minimizer of Prob. (2.1).

**Theorem 2.1** (First Order Necessary KKT Conditions [14, Thrm. 12.1]). *Suppose that $x^*$ is a local minimizer of Prob. (2.1), that the functions $F$, $c_i$ in (2.1) are continuously differentiable, and the LICQ holds at $x^*$. Then there are Lagrange multipliers $(\nu^*, \lambda^*) \in \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}^{|\mathcal{I}|}$, such that the following conditions are satisfied at $(x^*, \nu^*, \lambda^*)$:*

$$\begin{aligned}
\nabla_x \mathcal{L}(x^*, \nu^*, \lambda^*) &= 0, && \text{(2.4a)}\\
c_i(x^*) &= 0, \ \forall\, i \in \mathcal{E}, && \text{(2.4b)}\\
c_i(x^*) &\geq 0, \ \forall\, i \in \mathcal{I}, && \text{(2.4c)}\\
\lambda_i^* &\geq 0, \ \forall\, i \in \mathcal{I}, && \text{(2.4d)}\\
\lambda_i^* c_i(x^*) &= 0, \ \forall\, i \in \mathcal{I}. && \text{(2.4e)}
\end{aligned}$$

The conditions in Eq. (2.4) are collectively known as the *Karush-Kuhn-Tucker* conditions, or KKT conditions for short. Conditions (2.4b) and (2.4c) enforce that $x^*$ is a feasible point. Condition (2.4e) is known as a *complementarity condition*, and ensures that Lagrange multipliers $\lambda_i^*$ can only be non-zero when the corresponding constraint $c_i$ is active at $x^*$. This condition also has implications for the vanishing gradient condition of (2.4a), which we can expand as

$$\nabla_x \mathcal{L}(x^*, \nu^*, \lambda^*) = \nabla F(x^*) - \sum_{i \in \mathcal{E}} \nu_i^* \nabla c_i(x^*) - \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* \nabla c_i(x^*) = 0, \tag{2.5}$$

where we've excluded the sum over inequality constraints with corresponding zero $\lambda_i^*$. This condition can be viewed a consequence of *Farkas' Lemma* [14, p. 326]. To give an intuitive explanation we first set $n = |\mathcal{E}|$ and $m = |\mathcal{A}(x^*) \cap \mathcal{I}|$ and consider the cone

$$\mathcal{C}(x^*) = \{J_\mathcal{E}\, \nu + J_\mathcal{I}\, \lambda \mid w \in \mathbb{R}^n, \lambda \in \mathbb{R}^m, \lambda \geq 0\}, \tag{2.6}$$

where the matrices $J_\mathcal{E} = [\nabla c_i(x^*)]_{i \in \mathcal{E}}$ and $J_\mathcal{I} = [\nabla c_i(x^*)]_{i \in \mathcal{A}(x^*) \cap \mathcal{I}}$ are the matrices whose columns are the gradients of the equality constraint functions and active inequality constraint functions, respectively. Farkas' Lemma applied to our situation then states that given the vector $\nabla F(x^*) \in \mathbb{R}^d$, exactly one of the following two alternatives is true. Either, $\nabla F(x^*) \in \mathcal{C}$, or there exists $g \in \mathbb{R}^d$ such that

$$\nabla F(x^*)^T g < 0, \quad J_\mathcal{E}^T g = 0, \quad J_\mathcal{I}^T g \geq 0. \tag{2.7}$$

In the former case, Eq. (2.5) is fulfilled for some $(\nu^*, \lambda^*), \lambda^* \geq 0$, as stated, whilst the latter case implies that there exists a feasible direction $g$ which reduces the objective value whilst staying in the feasible set to first order. For a local minimizer $x^*$ this cannot be the case, and thus the inclusion of $\nabla F(x^*)$ in the cone $\mathcal{C}(x^*)$ becomes a necessary first order condition for a local minimizer.

The KKT conditions are central to the theory of constrained optimization, and for certain types of constrained optimization problems it is particularly useful that the KKT conditions become both necessary and sufficient for local minimizers under weaker constraint qualifications than LICQ.

## 2.2 The Lagrange Dual

In the previous section we defined the Lagrangian in Eq. (2.3) and treated it primarily as a function of the *primal* variable $x \in \mathbb{R}^d$. Now we use the Lagrangian to define the *Lagrange dual function* [15, p. 216],

$$\begin{aligned} &q \colon \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}^{|\mathcal{I}|} \to \mathbb{R} \cup \{-\infty\}, \\ q(\nu, \lambda) = \inf_{x \in \Omega} &\mathcal{L}(x, \nu, \lambda) = \inf_{x \in \Omega} \left( F(x) - \sum_{i \in \mathcal{E}} \nu_i c_i(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x) \right). \end{aligned} \tag{2.8}$$

The Lagrange dual is not well defined whenever $\mathcal{L}(x, \nu, \lambda)$ is unbounded from below, so we define the domain of $q$ as [14, p. 344]

$$\mathbf{dom}\, q = \{(\nu, \lambda) \mid q(\nu, \lambda) > -\infty\}. \tag{2.9}$$

Whenever $\lambda \geq 0$ and $(\nu, \lambda) \in \mathbf{dom}\, q$, we say that $(\nu, \lambda)$ is *dual feasible* [15, p. 216]. We denote the optimal objective value for Prob. (2.1) by $F^* := F(x^*)$, and an immediate consequence of the definition of the Lagrange dual is that for dual feasible $(\nu, \lambda)$

$$q(\nu, \lambda) \leq F^*, \tag{2.10}$$

and this property is known as *weak duality*. We can see that the property holds by noting that for any feasible point $\hat{x}$ and dual feasible $(\nu, \lambda)$

$$-\sum_{i \in \mathcal{E}} \nu_i c_i(\hat{x}) - \sum_{i \in \mathcal{I}} \lambda_i c_i(\hat{x}) \leq 0, \tag{2.11}$$

as $c_i(\hat{x}) = 0$ for $i \in \mathcal{E}$, and $c_i(\hat{x})$ are non-negative for $i \in \mathcal{I}$. Thus,

$$q(\nu, \lambda) = \inf_{x \in \Omega} \mathcal{L}(x, \nu, \lambda) \leq \mathcal{L}(\hat{x}, \nu, \lambda) \leq F(\hat{x}) \tag{2.12}$$

for all feasible points $\hat{x}$, and the inequality in Eq. (2.10) follows [15, p. 217]. Thus, any value $g(\nu, \lambda) > -\infty$ for dual feasible $(\nu, \lambda)$ is a lower bound for the optimal primal objective value $F^*$. As a follow-up question we can ask what the *best* lower bound on $F^*$ is. That can be answered by solving the *Lagrange dual problem* [15, p. 223]

$$\begin{aligned} q^* = \max_{\nu, \lambda} \quad & q(\nu, \lambda), \\ \text{s.t.} \quad & \lambda \geq 0. \end{aligned} \tag{2.13}$$

The quantity $F^* - q^*$ is called the *duality gap*, and if it's zero, we say that strong duality holds. Whenever strong duality holds, it has consequences for how the solutions to the primal and dual problem relate to one another. Let $x^*$ be a primal optimal point and $(\nu^*, \lambda^*)$ be dual optimal for an optimization problem where strong duality holds. Following Boyd and Vandenberghe [15, p. 242] we expand the strong duality condition as

$$\begin{aligned} F(x^*) &= q(\nu^*, \lambda^*), \\ &= \inf_{x \in \Omega} \left( F(x) - \sum_{i \in \mathcal{E}} \nu_i^* c_i(x) - \sum_{i \in \mathcal{I}} \lambda_i^* c_i(x) \right), \\ &\leq F(x^*) - \sum_{i \in \mathcal{E}} \nu_i^* c_i(x^*) - \sum_{i \in \mathcal{I}} \lambda_i^* c_i(x^*), \\ &\leq F(x^*), \end{aligned} \tag{2.14}$$

where the last inequality follows due to $(x^*, \nu^*, \lambda^*)$ being primal-dual feasible as in Eq. (2.11). We conclude that the inequalities in Eq. (2.14) are in fact equalities, which means that $x^*$ is a minimizer of $\mathcal{L}(x, \nu^*, \lambda^*)$. Furthermore, we see that $\sum_{i \in \mathcal{I}} \lambda_i^* c_i(x^*) = 0$, which implies $\lambda_i^* c_i(x^*) = 0, \;\; i \in \mathcal{I}$ because $\lambda_i^*, c_i(x^*)$ are all non-negative. This means that an analogous complementarity condition to Eq. (2.4e) holds for primal-dual optimal solutions $(x^*, \nu^*, \lambda^*)$ if strong duality holds.

## 2.3 Convex Problems

In general one cannot expect to find the global minimizer of a non-linear constrained optimization problem like Prob. (2.1), and instead one seek to find local minimizers characterized by first or second order conditions. For convex optimization problems however, one can show that any local minimizer is a global minimizer [15, pp. 136-139]. A set $\Omega \subset \mathbb{R}^d$ is convex if for all $x, y \in \Omega$ the cord connecting the two points is also included in $\Omega$, i.e. $(1-t)x + ty \in \Omega$ for $t \in [0,1]$. A function $f \colon \Omega \to \mathbb{R}$ is convex if its domain is convex, and if for all $x, y \in \Omega$

$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y), \; t \in [0,1]. \tag{2.15}$$

Positive sums of convex functions are also convex, i.e. for two convex functions $f_1$ and $f_2$, $\lambda_1 f_1 + \lambda_2 f_2$ is convex if $\lambda_1, \lambda_2 \geq 0$. Lastly, a function $g$ is concave if $-g$ is convex. Now we can state the requirements for an optimization problem to be convex.

**Definition 2.3** (Convex Optimization Problem).
*A* convex optimization problem *[14, p. 8] is one like in Prob.* (2.1) *where:*

- *The objective function $F(x)$ is convex,*
- *The equality constraints $c_i(x) = 0, i \in \mathcal{E}$ are affine,*
- *The inequality constraints $c_i(x) = 0, i \in \mathcal{I}$ are concave.*

For convex problems we can guarantee that strong duality holds and that the dual optimal value $q^*$ is attained by some dual feasible $(\nu^*, \lambda^*)$ under the relatively simple constraint qualification known as *Slater's constraint qualification* [15, pp. 226-227].

**Definition 2.4** (Slater's CQ [15, p. 226]). *For a constrained optimization problem of the form* (2.1)*,* Slater's constraint qualification *holds if there exists $x \in \mathbf{relint}\,\Omega$ such that*

$$c_i(x) = 0, \; \forall \; i \in \mathcal{E}, \quad c_i(x) > 0, \; \forall \; i \in \mathcal{I}. \tag{2.16}$$

To define the relative interior of a set $\Omega \subset \mathbb{R}^d$, denoted $\mathbf{relint}\,\Omega$, we first define the set of all affine combinations of points in a set $\Omega$, or its *affine hull* [15, p. 23]:

$$\mathbf{aff}\,\Omega = \{\theta_1 x_1 + \cdots + \theta_k x_k \mid x_1, \ldots, x_k \in \Omega, \; \theta_1 + \cdots + \theta_k = 1\}. \tag{2.17}$$

The relative interior of $\Omega$ is then defined as [15, Chap. 2.1.3]

$$\mathbf{relint}\,\Omega = \{x \in \Omega \mid B(x,r) \cap \mathbf{aff}\,\Omega \subseteq \Omega, \; \text{for some } r > 0\}, \tag{2.18}$$

where $B(x,r) = \{y \mid ||y - x|| < r\}$ for some norm $|| \cdot ||$ on $\mathbb{R}^d$.

For affine inequality constraints $c_j$, $j \in \mathcal{I}$ we can relax the requirement on those constraints in Slater's condition to the relaxed inequality $c_j(x) \geq 0$. Furthermore, if all the equality and inequality constraints are affine, then Slater's condition simplifies to the requirement that the primal feasible set is nonempty and $\mathbf{dom}\, F$ is open [15, pp. 227].

We also note that if Slater's condition holds for a convex optimization problem, then the KKT conditions are necessary and sufficient for optimality [15, p. 244]. Slater's condition implies that the duality gap is zero, and the dual optimum is attained. With the implication of Eq. (2.14) showing that a point $x^*$ which attains the infimum of $q(\nu, \lambda)$ is a minimizer of $F(x)$, we conclude that a point $x$ is optimal iff. there are $(\nu, \lambda)$ which together with $x$ fulfill the KKT conditions in Eq. (2.4).

## 2.4 The Wolfe Dual

Assuming that our optimization problem is convex with inequality constraints, we can construct the *Wolfe dual problem* [14, pp. 346-348]. The Wolfe dual problem consists of maximizing the Lagrangian of our problem over the primal and dual feasible variables, conditioned on its gradient w.r.t. the primal variables vanishing, as shown below.

$$\max_{x \in \Omega, \lambda \in \mathbb{R}^{|\mathcal{I}|}} \quad \mathcal{L}(x, \lambda) = F(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x), \tag{2.19a}$$

$$\text{s.t.} \quad \nabla_x \mathcal{L}(x, \lambda) = 0, \tag{2.19b}$$

$$\lambda \geq 0. \tag{2.19c}$$

This problem formulation can be useful for computing solutions to the Lagrange dual problem, although the equality constraint on the gradient of the Lagrangian is nonlinear in general and thus Prob. (2.19) can be non-convex.

Central to the usefulness of the Wolfe dual is its connection with the Lagrange dual problem in Eq. (2.13). Given dual feasible $\lambda$ the function $\mathcal{L}(\cdot, \lambda) \colon \Omega \to \mathbb{R}$ is convex for a convex optimization problem, because $-c_i$ is convex for $i \in \mathcal{I}$ and $\lambda \geq 0$. For any $\bar{x} \in \Omega$ which fulfills Eq. (2.19b) then,

$$\mathcal{L}(x, \lambda) \geq \mathcal{L}(\bar{x}, \lambda) + \nabla_x \mathcal{L}(\bar{x}, \lambda)^T (x - \bar{x}) = \mathcal{L}(\bar{x}, \lambda). \tag{2.20}$$

This means that the infimum $\inf_{x \in \Omega} \mathcal{L}(x, \lambda)$ is achieved at $\bar{x}$, and $q(\lambda) = \mathcal{L}(\bar{x}, \lambda)$.

Assuming that Slater's condition holds and $(x^*, \lambda^*)$ are a primal-dual optimal solution to Prob. (2.13), the KKT conditions, which are necessary for primal-dual optimality, ensure that all the constraints of Prob. (2.19) are met by $(x^*, \lambda^*)$. For any other feasible $(x, \lambda)$ then, we can follow [14, Thrm. 12.14] to show that

$$\begin{aligned} \mathcal{L}(x^*, \lambda^*) &= F(x^*), \\ &\geq F(x^*) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x^*), \\ &= \mathcal{L}(x^*, \lambda) \\ &\geq \mathcal{L}(x, \lambda) + \nabla_x \mathcal{L}(x^*, \lambda)^T (x - x^*), \\ &= \mathcal{L}(x, \lambda), \end{aligned} \tag{2.21}$$

meaning that $(x^*, \lambda^*)$ maximizes the Lagrangian under the required constraints, and thus solves the Wolfe dual problem.

# Classical Support Vector Machines 3

Before we expand on the SVM idea to classify manifold valued data, we first present the classical vector space SVM model. Following the presentation of Hastie, Tibshirani, et al. [16, Chap. 12] and Burges [17, pp. 128-136], the idea of the Support Vector Machine is to find a linear separator between two classes of points sampled on a feature space $\mathbb{R}^d$. We'll develop the model under the assumption that the two classes are *linearly separable*, a term which will be defined below, but relax this assumption later to allow for classification of non-linearly separable datasets. Let the training data be tuples of points $x \in \mathbb{R}^d$ and class labels $y \in \{1, -1\}$, i.e. $\mathcal{X}_\mathbb{R} = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \{1, -1\}$. Then define a hyperplane in $\mathbb{R}^d$ by its normal vector $\beta \in \mathbb{R}^d$, and bias $\beta_0$,

$$H(\beta, \beta_0) = \{x \in \mathbb{R}^d \mid f_{\text{SVM}}(x \mid \beta, \beta_0) := \beta_0 + \beta^T x = 0\}. \tag{3.1}$$

Fig. 3.1 illustrates binary classification of a dataset on $\mathbb{R}^2$ by linear separators, with three different hyperplanes overlaid. Only two of the three hyperplanes successfully separate the two classes, with H$_3$ having the greatest margin to any training point. We call the function $f_{\text{SVM}} \colon \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ the *classifier*, and

$$\text{sign}(f_{\text{SVM}}(x \mid \beta, \beta_0)) \in \{1, -1\} \tag{3.2}$$

is used as the classifying rule to separate the two different classes.

The value $f_{\text{SVM}}(x \mid \beta, \beta_0)$ also has a geometric interpretation as $||\beta||_2$ times the *signed distance* between $x$ and $H(\beta, \beta_0)$. Letting $x_1 \in H(\beta, \beta_0)$, the distance between the hyperplane $H(\beta, \beta_0)$ and $x \in \mathbb{R}^d$ can be computed as the length of the difference $x - x_1$ projected onto the hyperplane normal vector $\frac{1}{||\beta||}\beta$:

$$\left| \frac{1}{||\beta||} \beta^T (x - x_1) \right| = \frac{1}{||\beta||} \left| \left( \beta^T x + \beta_0 \right) \right| = \frac{1}{||\beta||} \left| f_{\text{SVM}}(x \mid \beta, \beta_0) \right|, \tag{3.3}$$

where we've used that $\beta^T x_1 = -\beta_0$. Leaving out the absolute value from the above expression, the signed distance $\frac{1}{||\beta||} f_{\text{SVM}}(x \mid \beta, \beta_0)$ has magnitude equal to the distance between $H(\beta, \beta_0)$ and $x$. When $f_{\text{SVM}}(x \mid \beta, \beta_0)$ is positive we say that $x$ is on the positive side of $H(\beta, \beta_0)$, and on the negative side in the opposite case.

The dataset $\mathcal{X}_\mathbb{R}$ is called linearly separable if there exists $\beta \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$ s.t.

$$y_i f_{\text{SVM}}(x_i \mid \beta, \beta_0)) > 0, \ \forall \, i. \tag{3.4}$$

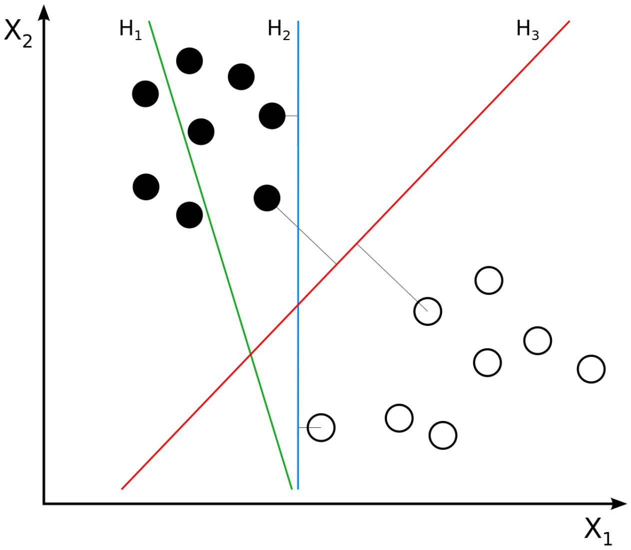

**Fig. 3.1.:** Illustrative example of a two class dataset in $\mathbb{R}^2$ with three linear separators. The line (1D hyperplane) H$_1$ does not separate the full circles from the open circles. H$_2$ does separate the two classes narrowly, whilst H$_3$ is the maximum margin separator. Meaning it has the maximum minimal distance to any training point. Source: [40]

There are infinitely many linear separators of a linearly separable dataset, but we can specify a unique linear separator which is also robust to perturbations in the data by finding the *maximum margin* linear separator. The *margin* between a linear separator $f(\cdot \mid \beta, \beta_0)$ with $||\beta|| = 1$ and training data $\mathcal{X}_\mathbb{R}$ is defined as

$$\mathrm{M} = \min_{(y,x)\in\mathcal{X}_\mathbb{R}} y f_{\mathrm{SVM}}(x \mid \beta, \beta_0), \tag{3.5}$$

as $f_{\mathrm{SVM}}(x \mid \beta, \beta_0)$ is the signed distance to $H(\beta, \beta_0)$ when $||\beta||_2 = 1$. To find this maximum margin separator we need to solve the optimization problem

$$\max_{\beta, \beta_0} \quad \mathrm{M}, \tag{3.6a}$$

$$\text{s.t.} \quad y_i \, f_{\mathrm{SVM}}(x_i \mid \beta, \beta_0) \geq \mathrm{M} \; \forall \, i, \tag{3.6b}$$

$$||\beta|| = 1. \tag{3.6c}$$

Hastie, Tibshirani, et al. [16, p. 132], show that the norm-one requirement on $\beta$ can be discarded by noting that $|f_{\mathrm{SVM}}(x \mid \beta, \beta_0)| \propto ||\beta||$, and rescaling the l.h.s. of the inequalities in Eq. (3.6b) by $1/||\beta||$.

$$\frac{1}{||\beta||} y_i \, f_{\mathrm{SVM}}(x_i \mid \beta, \beta_0) \geq \mathrm{M} \quad \Rightarrow \quad y_i \, f_{\mathrm{SVM}}(x_i \mid \beta, \beta_0) \geq ||\beta|| \mathrm{M} = 1, \tag{3.7}$$

where we've set $\mathrm{M} = 1/||\beta||$. With this rescaling Prob. (3.6) can be restated as the equivalent convex quadratic minimization problem

$$\min_{\beta,\,\beta_0} \quad \frac{1}{2}||\beta||^2, \tag{3.8a}$$

$$\text{s.t.} \quad y_i\, f_{\text{SVM}}(x_i \mid \beta, \beta_0) \geq 1,\ \forall\, i. \tag{3.8b}$$

This problem is convex as per Def. 2.3 the objective is convex and all the inequality constraints, $c_i(\beta, \beta_0) = y_i(\beta_0 + \beta^T x_i) - 1$, are linear in $\beta$ and $\beta_0$. However, the feasible set for Prob. (3.8) is only non-empty if the training data $\mathcal{X}_{\mathbb{R}}$ is linearly separable. To allow for some under classification/misclassification of training points the standard approach [16, p. 419] is to introduce non-negative slack variables $\xi \in \mathbb{R}^N$, $\xi \geq 0$, and modify the classification inequalities in Eq. (3.8b) to

$$y_i\, f_{\text{SVM}}(x_i \mid \beta, \beta_0) \geq 1 - \xi_i,\ \forall\, i. \tag{3.9}$$

For each training point $x_i$ then, the corresponding $\xi_i$ measures the proportional amount w.r.t. the margin $M$ by which the point $x_i$ can be under classified, and misclassifications will happen when $\xi_i > 1$. To control the balance between the size of the classification margin and the average under/misclassification we add the term $(1/N)\sum_{i=1}^{N} \xi_i$ scaled by the hyperparameter $C > 0$ to the objective in Eq. (3.8a)

$$\min_{\beta,\,\beta_0,\,\xi} \quad \frac{1}{2}||\beta||^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i, \tag{3.10a}$$

$$\text{s.t.} \quad y_i\, f_{\text{SVM}}(x_i \mid \beta, \beta_0) \geq 1 - \xi_i,\ \forall\, i, \tag{3.10b}$$

$$\xi \geq 0. \tag{3.10c}$$

This optimization problem neatly generalizes the linearly separable case, as we can reduce back to the linearly separable case in Prob. (3.8) by letting $C \to \infty$.

## 3.1 Solving for the Classical SVM Model

In order to solve Prob. (3.10) we could proceed directly with the primal formulation as stated, but this would require that we find a feasible starting point $(\tilde{\beta}, \tilde{\beta}_0, \tilde{\xi})$, and handle the relatively intricate interplay between the objective function and the classification inequalities mediated by the slack variables $\xi$. The more common way to solve Prob. (3.8) is to transform the problem to its Wolfe dual from Sec. 2.4, and solve a relatively easy quadratic maximization problem over the Lagrange multipliers corresponding to the classification inequalities [16, pp. 420-421], [17, pp. 128-132].

The Lagrangian of Prob. (3.8) with Lagrangian multipliers $\lambda, \mu \in \mathbb{R}^N$ for the constraints in Eq. (3.10b) and Eq. (3.10c) respectively, is

$$\mathcal{L}_{\text{SVM}}(\beta, \beta_0, \xi, \lambda, \mu) = \frac{1}{2}||\beta||^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i(y_i\, f_{\text{SVM}}(x_i \mid \beta, \beta_0) - 1 + \xi_i) - \sum_{i=1}^{N}\mu_i\xi_i. \tag{3.11}$$

Writing out the expression for the classifier $f_{\text{SVM}}(x_i \mid \beta, \beta_0)$ and rearranging terms the Lagrangian can be expressed as

$$\mathcal{L}_{\text{SVM}} = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{N}\lambda_i y_i x_i^T\beta + \sum_{i=1}^{N}\lambda_i - \beta_0\sum_{i=1}^{N}\lambda_i y_i + \sum_{i=1}^{N}(C/N - \lambda_i - \mu_i)\xi_i. \tag{3.12}$$

All the inequality constraints in Prob. (3.8) are affine and the domain of the objective function is open, meaning Slater's condition reduces to determining whether there exists a feasible point $(\tilde{\beta}, \tilde{\beta}_0, \tilde{\xi})$. Due to the addition of the slack variables $\xi$ there always exists a feasible point as we can simply increase any $\xi_i$ until the relevant constraint is satisfied for any $(\beta, \beta_0)$. The KKT conditions are thus both necessary and sufficient for optimality of Prob. (3.8) and they are

$$\nabla_\beta\mathcal{L}_{\text{SVM}} = \beta - \sum_{i=1}^{N}\lambda_i y_i x_i = 0, \tag{3.13a}$$

$$\partial_{\beta_0}\mathcal{L}_{\text{SVM}} = -\sum_{i=1}^{N}\lambda_i y_i = 0, \tag{3.13b}$$

$$\partial_{\xi_i}\mathcal{L}_{\text{SVM}} = C/N - \lambda_i - \mu_i = 0,\ \forall\, i, \tag{3.13c}$$

$$y_i\, f_{\text{SVM}}(p_i \mid \beta, \beta_0) - 1 + \xi_i \geq 0,\ \forall\, i, \tag{3.13d}$$

$$\lambda_i(y_i\, f_{\text{SVM}}(x_i \mid \beta, \beta_0) - 1 + \xi_i) = 0,\ \forall\, i, \tag{3.13e}$$

$$\mu_i\xi_i = 0,\ \forall\, i, \tag{3.13f}$$

$$\xi, \lambda, \mu \geq 0. \tag{3.13g}$$

The vanishing gradient conditions of Eq. (3.13a) and Eq. (3.13b) imply that

$$\beta = \sum_{i=1}^{N}\lambda_i y_i x_i, \quad \text{and} \quad \sum_{i=1}^{N}\lambda_i y_i = 0 \tag{3.14}$$

at critical points of the Lagrangian. Combining the vanishing gradient condition (3.13c) with the non-negativity constraints on $\xi$, $\lambda$, and $\mu$ allows us to eliminate $\mu$ by constraining $\lambda$ to $0 \leq \lambda \leq C/N$ and implicitly setting $\mu_i = C - \lambda_i$. The complementarity condition in Eq. (3.13f) ensures that $\xi_i = 0$ if $\lambda_i < C/N$, and for any $0 < \lambda_i < C/N$ the complementarity condition in Eq. (3.13e) ensures that the corresponding point $x_i$ is exactly on the margin, i.e. $y_i f(x_i) = 1$. We can use this to solve for the bias $\beta_0$ as

$$\beta_0 = y_i - \beta^T x_i, \tag{3.15}$$

where $x_i$ is the corresponding training point to $\lambda_i$. Numerically one usually takes the average of the $\beta_0$ computed from all margin points with $0 < \lambda_i < C/N$. Crucially, $\lambda_i$ is only non-zero if the corresponding training point $x_i$ is exactly on the classification margin, or under/misclassified by $\xi_i > 0$. This generally leads to a sparse optimal $\lambda$.

Inserting the vanishing gradient requirements back to the expression for the Lagrangian results in the Wolfe dual objective

$$\begin{aligned}\mathcal{L}_{\mathrm{SVM}}(\beta, \beta_0, \xi, \lambda, \mu) &= \frac{1}{2}\left\|\sum_{i=1}^{N} \lambda_i y_i x_i\right\|^2 - \sum_{i=1}^{N} \lambda_i y_i x_i^T \left(\sum_{j=1}^{N} \lambda_j y_j x_j\right) + \sum_{i=1}^{N} \lambda_i, \\ &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i y_i x_i^T x_j \lambda_j y_j + \sum_{i=1}^{N} \lambda_i, \\ &= -\frac{1}{2}\lambda^T (\mathbf{YKY})\, \lambda + \lambda^T \mathbb{1}.\end{aligned} \tag{3.16}$$

Here we've introduced the *class label matrix* $\mathbf{Y} = \mathrm{diag}(y_1, \ldots, y_N)$ and the *kernel matrix* $\mathbf{K} \in \mathbb{R}^{N \times N}$, whose entries are the pairwise inner products between our training data. That is, $(\mathbf{K})_{i,j} = \langle x_i, x_j \rangle$. The naming of this matrix will be explained later, but for now it's noteworthy that K is a *Gram* matrix [18, p. 441], and all Gram matrices are positive semi-definite. For any $\alpha \in \mathbb{R}^N$

$$\alpha^T \mathbf{K} \alpha = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \langle x_i, x_i \rangle \alpha_j = \left\langle \sum_{i=1}^{N} \alpha_i x_i, \sum_{i=1}^{N} \alpha_i x_i \right\rangle \geq 0. \tag{3.17}$$

After inserting the vanishing gradient conditions and box constraints for $\lambda$, the Wolfe dual to Prob. (3.10) becomes

$$\hat{\lambda} = \underset{0 \leq \lambda \leq \frac{C}{N}}{\arg\max} \quad \mathcal{D}_{\mathrm{SVM}} := -\frac{1}{2}\lambda^T (\mathbf{YKY})\, \lambda + \lambda^T \mathbb{1}. \tag{3.18}$$

The dual objective $\mathcal{D}_{\mathrm{SVM}}$ is concave, and as such $\min_\lambda -\mathcal{D}(\lambda)$ with box constraints on $\lambda$ is a convex quadratic optimization problem. Numerically, Prob. (3.18) can be solved using optimization algorithms like the interior point based *Ipopt* algorithm [19], or the *SCS* algorithm [20] for convex problems. And having solved for $\hat{\lambda}$, we recover the primal solution $\hat{\beta}$ from Eq. (3.14) and $\hat{\beta}_0$ by averaging Eq. (3.15) applied to all margin points identified by $0 < \hat{\lambda}_i < C/N$.

## 3.2 The Kernel Trick

The idea behind the *kernel trick* comes from realizing that the only manner in which our training data points enter the Wolfe dual in Prob. (3.18) is through pairwise inner products as elements of $\mathbf{K}$ [17, pp. 138-143][16, Chap. 12.3]. That means we can map our training data from the feature space $\mathbb{R}^d$ into another Hilbert space $\mathcal{V}$ by a mapping $\Phi\colon \mathbb{R}^d \to \mathcal{V}$. Then we construct the kernel matrix element wise as

$$(\mathbf{K})_{i,j} = \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{V}}, \tag{3.19}$$

and Prob. (3.18) optimizes for the maximum margin linear separator in the possibly infinite dimensional Hilbert space $\mathcal{V}$. The separator value for $x \in \mathbb{R}^d$ is computed by mapping into $\mathcal{V}$ through $\Phi$ as

$$f_{\text{SVM}}(\Phi(x) \mid \beta, \beta_0) = \beta_0 + \sum_{i=1}^{N} \lambda_i y_i \langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{V}}, \tag{3.20}$$

where we've expanded the hyperplane normal vector $\beta \in \mathcal{V}$ as $\beta = \sum_{i=1}^{N} \lambda_i y_i \Phi(x_i)$. However, for higher dimensional $\mathcal{V}$, it can be prohibitively expensive or numerically infeasible to compute the explicit mappings $x \mapsto \Phi(x) \in \mathcal{V}$. And the explicit value of $\Phi(x_i)$ is only ever used to compute its inner product with another element $\Phi(x_j)$. Therefore, we can compose the mapping $\Phi \times \Phi$ with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ to construct the corresponding *kernel*

$$K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \quad K(\tilde{x}, \bar{x}) = \langle \Phi(\tilde{x}), \Phi(\bar{x}) \rangle_{\mathcal{V}}, \tag{3.21}$$

which first applies the mapping $\Phi$ to both its arguments before computing their inner product in $\mathcal{V}$. This perspective also opens the question of what properties we must require of a function $K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ in order for it to implicitly define a mapping $\Phi\colon \mathbb{R}^d \to \mathcal{V}$ and be a valid kernel. It turns out that as long as our kernel is *positive definite*, a term defined below, the implicit mapping into a Hilbert space exists. And by using a positive definite kernel we perform linear separation in a possibly very high dimensional space.

We state the following definitions of *positive definite* and *negative definite* kernels, due to Berg, Christensen, et al. [21]:

**Definition 3.1** (Positive and negative definite kernels [21, Chap. 3, Def. 1.1]). *Let $\mathcal{X}$ be a nonempty set. A symmetric function $\phi\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a* positive definite kernel *if and only if*

$$\sum_{i,j=1}^{N} c_i c_j \phi(x_i, x_j) \geq 0, \tag{3.22}$$

*for all $N \in \mathbb{N}$, $\{x_i\}_{i=1}^{N} \subseteq \mathcal{X}$, and $\{c_i\}_{i=1}^{N} \subseteq \mathbb{R}$. A symmetric function $\psi\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called* conditionally negative definite *if and only if*

$$\sum_{i,j=1}^{N} c_i c_j \psi(x_i, x_j) \leq 0, \tag{3.23}$$

*for $N \geq 2$, $\{x_i\}_{i=1}^{N} \subseteq \mathcal{X}$, and $\{c_i\}_{i=1}^{N} \subseteq \mathbb{R}$, with $\sum_{i=1}^{N} c_i = 0$.*

Note the additional requirement on $\{c_i\}_{i=1}^N$ for a function $\psi$ to be a conditionally negative definite kernel, i.e. $\sum_{i=1}^N c_i = 0$, which explains the requirement $N \geq 2$.

The above definition of a positive definite kernel is equivalent to requiring that the matrix $K_\phi \in \mathbb{R}^N,\ (K_\phi)_{i,j} = \phi(x_i, x_j)$ be positive *semi*-definite. However, for historical reasons one call kernels which satisfy the requirement in Eq. (3.22) with a strict inequality a *strictly positive definite kernel*, and the same prefix applies for any conditionally negative definite kernel which satisfies Eq. (3.23) with a strict inequality [21, p. 67]. Berg, Christensen, et al. [21] omit the word "conditionally" when defining conditionally negative definite kernels, but due to the extra constraint on the coefficients $\{c_i\}_{i=1}^N$ for which a function $\psi$ needs to satisfy Eq. (3.23), we've chosen to define them as *conditionally* negative definite kernels to avoid confusion.

Positive definite kernels enjoy a plethora of different properties. A useful first result is that if $h\colon \mathcal{X} \to \mathcal{X}$ is a bijection, then $\phi$ is a positive (resp. conditionally negative) definite kernel iff. $\phi \circ (h \times h)$ is a positive (resp. conditionally negative) definite kernel [21, p. 67]. Furthermore, given an arbitrary function $g\colon \mathcal{X} \to \mathbb{R}$ the kernel $\phi(x_1, x_2) = g(x_1)g(x_2)$ is positive definite [21, p. 69], as

$$\sum_{i,j=1}^N c_i c_j \phi(x_i, x_j) = \left| \sum_{i,j=1}^N c_i g(x_i) \right|^2 \geq 0. \tag{3.24}$$

Any positive definite kernel $\phi$ necessarily satisfies the property that

$$\phi(x, x) \geq 0 \;\forall\; x \in \mathcal{X}, \tag{3.25}$$

and by considering the requirement in Eq. (3.22) for any $x_1, x_2 \in \mathcal{X}$ and arbitrary $c_1, c_2 \in \mathbb{R}$, we find the necessary requirement

$$\det\left(\begin{bmatrix} \phi(x_1, x_1) & \phi(x_1, x_2) \\ \phi(x_2, x_1) & \phi(x_2, x_2) \end{bmatrix}\right) \geq 0, \;\Rightarrow\; |\phi(x_1, x_2)|^2 \leq \phi(x_1, x_1)\phi(x_2, x_2), \tag{3.26}$$

for any positive definite kernel $\phi$ [21, p. 69]. The following Theorem due to Berg, Christensen, et al. [21], also shows that positive definite kernels are closed under pointwise multiplication.

**Theorem 3.1** (Multiplication of pos. def. kernels [21, Chap. 3, Thrm. 1.12]). *Let $\phi_1, \phi_2\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be positive definite kernels. Then the kernel*

$$(\phi_1 \cdot \phi_2)\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \quad (\phi_1 \cdot \phi_2)(x_1, x_2) = \phi_1(x_1, x_2) \cdot \phi_2(x_1, x_2), \tag{3.27}$$

*is positive definite too.*

The proof of Theorem 3.1 uses properties of positive semi definite matrices, specifically the fact that a matrix $A \in \mathbb{R}^{n \times n}$ is positive semi definite iff. it can be realized as the

Gram matrix of a set of vectors $\{b_i\}_{i=1}^n, b_i \in \mathbb{R}^k$, i.e. $(A)_{i,j} = b_i^T b_j$ [18, Thrm. 7.2.10]. And building on top of Theorem 3.1, one can show that if $\phi$ is positive definite, the composition $\exp \circ \phi \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel [21, Chap. 3, Cor. 1.14]. Finally, the next theorem connects conditionally negative definite kernels to positive definite kernels through composition with the exponential function.

**Theorem 3.2** (Exp. of neg. def. kernel [21, Chap. 3, Thrm. 2.2]). *Let $\mathcal{X}$ be a non-empty set, and let $\psi \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel. Then $\psi$ is conditionally negative definite iff. $\exp(-t\psi)$ is positive definite for all $t > 0$.*

Using the above properties of positive definite kernels, we can prove that the well known Radial Basis Function (RBF) kernel is positive definite.

**Theorem 3.3** (The Radial Basis Function Kernel). *Let $\mathcal{V}$ be Hilbert space with inner product $\langle \cdot, \cdot, \rangle_{\mathcal{V}}$. Then the RBF kernel*

$$K_{RBF} \colon \mathcal{V} \times \mathcal{V} \times \mathbb{R}^+ \to \mathbb{R}, \quad K_{RBF}(x_1, x_2 \mid \sigma^2) = \exp\left(-\frac{||x_1 - x_2||_{\mathcal{V}}^2}{2\sigma^2}\right), \tag{3.28}$$

*is positive definite for all $\sigma^2 > 0$.*

*Proof.* As $\mathcal{V}$ is a Hilbert space, we can express the squared distance function as

$$||x_1 - x_2||_{\mathcal{V}}^2 = \langle x_1 - x_2, x_1 - x_2 \rangle_{\mathcal{V}}$$

and factorize $K_{\text{RBF}}(x_1, x_2)$ as

$$K_{\text{RBF}}(x_1, x_2) = \exp\left(-\frac{||x_1||_{\mathcal{V}}^2}{2\sigma^2}\right) \exp\left(-\frac{||x_2||_{\mathcal{V}}^2}{2\sigma^2}\right) \exp\left(\frac{\langle x_1, x_2 \rangle_{\mathcal{V}}}{\sigma^2}\right). \tag{3.29}$$

If we define $g(x) = \exp\left(-\frac{||x||_{\mathcal{V}}^2}{2\sigma^2}\right)$, the kernel $K_{\text{RBF}}^1(x_1, x_2) = g(x_1)g(x_2)$ is positive definite by the property described in Eq. (3.24). As we've shown in Eq. (3.17), any inner product is a positive definite kernel, and remains so when composed with the bijection $h(x) = x/\sigma$. Therefore, $K_{\text{RBF}}^2(x_1, x_2) = \exp\left(\frac{\langle x_1, x_2 \rangle_{\mathcal{V}}}{\sigma^2}\right)$ is positive definite, being the exponential of a positive definite kernel. Finally, due to Thrm. 3.1 the pointwise multiplication of the two positive definite kernels $K_{\text{RBF}}^1, K_{\text{RBF}}^2$ results in another positive definite kernel, $K_{\text{RBF}} = (K_{\text{RBF}}^1 \cdot K_{\text{RBF}}^2)$. □

## 3.3 Reproducing Kernel Hilbert Spaces

In order to show that any positive definite kernel $\phi$ implicitly defines a mapping $\Phi \colon \mathcal{X} \to \mathcal{H}$ where $\mathcal{H}$ is a Hilbert space such that $\phi(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle_{\mathcal{H}}$, we refer to the notion of a Reproducing Kernel Hilbert Space (RKHS) associated with any positive definite kernel [21, Chap. 3, §3].

Aronszajn [22, Thrm. 4, p. 344] states that for every positive definite kernel there exists a unique corresponding Hilbert space $H$, with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. To give some intuition for the above statement, we follow Berg, Christensen, et al. [21, Chap. 3, 3.1] and let $\phi \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel. Let $\mathcal{H}_0$ be the linear subspace of $\mathbb{R}^{\mathcal{X}}$ generated by the functions $\{\phi_x \mid x \in \mathcal{X}\}$, where $\phi_x \colon \mathcal{X} \to \mathbb{R}$, $\phi_x(y) = \phi(x, y)$ [21, p. 81]. If $f = \sum_i c_i \phi_{x_i}$ and $g = \sum_j d_j \phi_{y_j}$ belong to $\mathcal{H}_0$, then the quantity

$$\sum_{i,j} c_i d_j \phi(x_i, y_j) = \sum_j d_j f(y_j) = \sum_i c_i g(x_i), \tag{3.30}$$

does not depend on the (possibly non unique) representations of $f$ and $g$, and we denote it $\langle f, g \rangle_{\mathcal{H}_0}$. By setting $g = \phi_x$ we see from Eq. (3.30) that

$$\langle f, \phi_x \rangle_{\mathcal{H}_0} = f(x), \ \forall f \in \mathcal{H}_0, x \in \mathcal{X}. \tag{3.31}$$

The above relation in called the *reproducing* property of $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$, and specifically implies that $\langle \phi_x, \phi_y \rangle_{\mathcal{H}_0} = \phi(x, y)$. As $\phi$ is positive definite

$$\langle f, f \rangle_{\mathcal{H}_0} = \sum_{i,j} c_i c_j \phi(x_i, x_j) \geq 0. \tag{3.32}$$

Using the reproducing property and the nonnegative determinant property of positive definite kernels from Eq. (3.26) we see that

$$|f(x)|^2 = |\langle f, \phi_x \rangle_{\mathcal{H}_0}|^2 \leq \langle f, f \rangle_{\mathcal{H}_0} \cdot \phi(x, x), \tag{3.33}$$

implying that $\langle f, f \rangle_{\mathcal{H}_0} = 0$ iff. $f$ is identically zero. Therefore, $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ defines an inner product over $\mathcal{H}_0$ forming a pre-Hilbert space $(\mathcal{H}_0, \langle \cdot, \cdot \rangle_{\mathcal{H}_0})$, and its completion is the aforementioned Hilbert space $\mathcal{H}$ [21, p. 81]. In conclusion, we state the following theorem:

**Theorem 3.4** (RKHS [22, Thrm. 4, p. 344][21, p. 82]). *Let $\mathcal{X}$ be a nonempty set and $\phi \colon \mathcal{X} \times \mathcal{X} \to$ a positive definite function. Then there exists a unique Hilbert space $\mathcal{H}$ corresponding to $\phi$ and a mapping $\Phi \colon \mathcal{X} \to \mathcal{H}, x \mapsto \phi_x$ such that*

$$\phi(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}, \tag{3.34}$$

*where $\phi_x \colon \mathcal{X} \to \mathbb{R}, \phi_x(y) = \phi(x, y)$.*

# Smooth Riemannian Manifolds 4

Consider a set $\mathcal{M}$. Given a subset $\mathcal{U} \subset \mathcal{M}$ we call a bijection $\varphi \colon \mathcal{U} \to \mathbb{R}^d$ onto an open subset of $\mathbb{R}^d$ a $d$-dimensional *chart* of the set $\mathcal{M}$, denoted by the tuple $(\mathcal{U}, \varphi)$. When clear we suppress the explicit mention of the domain $\mathcal{U}$, and simply write $\varphi$ to represent the relevant chart $(\mathcal{U}, \varphi)$. For any $p \in \mathcal{U}$, the value $\varphi(p) \in \mathbb{R}^d$ is called the *coordinates* of $p$ in the chart $\varphi$ [23, p. 18].

Charts $\varphi \colon \mathcal{U} \to \mathbb{R}^d$ and their inverses $\varphi^{-1} \colon \varphi(\mathcal{U}) \to \mathcal{U}$ function as local handles on the set $\mathcal{M}$. For example, they allow us to study functions $f \colon \mathcal{U} \to \mathbb{R}$ by instead considering $f \circ \varphi^{-1} \colon \varphi(\mathcal{U}) \to \mathbb{R}$ which we can analyze using tools from real analysis. However, we generally require a collection of several charts to cover the set $\mathcal{M}$, and wherever the domains of two different charts $(\mathcal{U}_1, \varphi_1)$ and $(\mathcal{U}_2, \varphi_2)$ overlap, the properties of structures defined on the set $\mathcal{M}$ should not be dependent on the specific choice of chart. The notion of a *good* collection of charts to cover a manifold in a compatible way is codified into the concept of an *atlas*.

**Definition 4.1** (Atlas [23, p. 19]). *A (smooth)* atlas $\mathcal{A}$ of $\mathcal{M}$ into $\mathbb{R}^d$ *is a collection of charts* $\{(\mathcal{U}_\alpha, \varphi_\alpha)\}$ *of the set* $\mathcal{M}$ *such that:*

1. $\bigcup\limits_{\alpha \in I} \mathcal{U}_\alpha = \mathcal{M}$.
2. *For any pair* $\alpha, \beta$ *with* $\mathcal{U}_\alpha \cap \mathcal{U}_\beta = \mathcal{C} \neq \emptyset$*, the images* $\varphi_\alpha(\mathcal{C})$ *and* $\varphi_\beta(\mathcal{C})$ *are open sets in* $\mathbb{R}^d$*, and the chart transition function* $\varphi_\alpha \circ \varphi_\beta^{-1} \colon \varphi_\beta(\mathcal{C}) \to \mathbb{R}^d$ *is differentiable (*$C^\infty$ *smooth) over its domain* $\varphi_\beta(\mathcal{C})$.

Essentially we require that the collection of charts covers the whole set $\mathcal{M}$, and the *change of charts* function $\varphi_\alpha \circ \varphi_\beta^{-1}$ must be differentiable for any pair of overlapping charts, or $C^\infty$ smooth for a smooth atlas. Two atlases $\mathcal{A}_1, \mathcal{A}_2$ over $\mathcal{M}$ are equivalent if the union of both collections of charts $\mathcal{A}_1 \cup \mathcal{A}_2$ is still an atlas. That is, for any chart $(\mathcal{U}, \varphi) \in \mathcal{A}_2$, the collection of charts $\mathcal{A}_1 \cup (\mathcal{U}, \varphi)$ is still a chart according to Def. 4.1. Furthermore, given an atlas $\mathcal{A}$ we define the *maximal atlas* generated by $\mathcal{A}$ as the set of charts $(\mathcal{U}, \varphi)$ such that $(\mathcal{U}, \varphi) \cup \mathcal{A}$ is still an atlas, and denote it $\mathcal{A}^+$. Connecting back to the idea of equivalent atlases, two atlases $\mathcal{A}_1, \mathcal{A}_2$ are equivalent iff. they generate the same maximal atlas, i.e. $\mathcal{A}_1^+ = \mathcal{A}_2^+$ [23, p. 19].

A collection of charts from a maximal atlas $\mathcal{A}^+$ over $\mathcal{M}$ induces a topology over $\mathcal{M}$, called the *atlas topology* of $\mathcal{M}$. In this topology a set $\mathcal{V} \subset \mathcal{M}$ is open iff. for any chart $(\mathcal{U}, \varphi)$, $\varphi(\mathcal{U} \cap \mathcal{V})$ is an open subset of $\mathbb{R}^d$. Throughout this thesis we will be referring to the atlas topology when talking about open sets on a manifold. With the concepts of charts and atlases over the set $\mathcal{M}$ defined, we are ready to state the definition of a differentiable/smooth manifold.

**Definition 4.2** ($d$-dimensional Manifold [4, Def. 2.1])**.** *A differentiable (smooth) manifold of dimension $d$ is a set $\mathcal{M}$ along with an atlas $\mathcal{A}^+$ over $\mathcal{M}$ such that*

- *$\mathcal{A}^+$ is a (smooth) maximal atlas of $\mathcal{M}$ into $\mathbb{R}^d$.*
- *The atlas topology induced by $\mathcal{A}^+$ is* Hausdorf *and* second countable.

For a topology to be *Hausdorff* means that for any two distinct points $x, y \in \mathcal{M}$, there exists open subsets $V, W$ of $\mathcal{M}$ such that $x \in V,\ y \in W$ and $V \cap W = \emptyset$ [24, p. 85]. In essence, all distinct points in a set with a Hausdorff topology are distinguishable by non-overlapping open sets. Furthermore, for a topology on $\mathcal{M}$ to be *second countable* there must exist a countable collection of open sets $\mathcal{T} = \{V_i\}_{i\in\mathcal{K}}, V_i \subset \mathcal{M}$ such that all open sets $W \subset \mathcal{M}$ can be written as a union of sets from $\mathcal{T}$ over a sub-sequence $\mathcal{K}' \subset \mathcal{K}$, i.e. $W = \bigcup_{j\in\mathcal{K}'} V_j$ [24, p. 84].

Both Absil, Mahony, et al. [23] and do Carmo [4] state that the conditions on the atlas $\mathcal{A}^+$ for a tuple $(\mathcal{M}, \mathcal{A}^+)$ to be a maximal atlas along with inducing a Hausdorff atlas topology on $\mathcal{M}$ is included for technical reasons to avoid unconventional topologies. If these conditions were not present, one would allow for atlas topologies on manifolds where convergent sequences of points may not have a single limit point [23, p. 19]. A maximal atlas over a set $\mathcal{M}$ which satisfies the requirements of Def. 4.2 is called a *manifold structure* or a *differentiable structure* on $\mathcal{M}$. To construct a manifold structure on a set $\mathcal{M}$ we do not need to define a maximal atlas however, it is enough to define an atlas $\mathcal{A}$ whose maximal atlas generates the manifold structure on $\mathcal{M}$. And any such atlas is called an *atlas of the manifold* $(\mathcal{M}, \mathcal{A}^+)$ [23, p. 20]. When $(\mathcal{M}, \mathcal{A}^+)$ is a manifold, we sometimes say "the manifold $\mathcal{M}$" when referring to an implicit manifold structure, and "the set $\mathcal{M}$" when referring to the underlying set $\mathcal{M}$ without the manifold structure.

## 4.1 Vector Space Manifolds

An important class of sets with a “trivial” manifold structure are vector spaces $\mathcal{E}$. Although trivial in a sense, some definitions and concepts can be unified by considering $\mathbb{R}^d$ or $\mathbb{R}^{m\times n} \cong \mathbb{R}^{mn}$ as having a manifold structure in their own right. Let $\mathcal{E}$ be a $d$ dimensional vector space with basis $\{p_i\}_{i=1}^d$ such that for any $p \in \mathcal{E}, p = \sum_{i=1}^d a_i\, p_i$. Then we can define a chart with global domain $\mathcal{U} = \mathcal{E}$ as

$$\varphi_{\mathcal{E}} \colon \mathcal{E} \to \mathbb{R}^d, \quad \varphi_{\mathcal{E}}(p) \mapsto [a_1, ..., a_n], \tag{4.1}$$

and the atlas with the single chart $\mathcal{A} = (\mathcal{U}, \varphi_{\mathcal{E}})$. Then $\mathcal{A}$ is an atlas of the manifold $(\mathcal{E}, \mathcal{A}^+)$, generating a manifold structure over the vector space $\mathcal{E}$ [23, pp. 22-23]. These vector space manifolds are subsequently useful as building blocks when viewing manifolds as embedded within vector spaces, like for example constrained sets of matrices in $\mathbb{R}^{m\times n}$, where the precise meaning of an *embedded manifold* will be specified later. The vector space $\mathbb{R}^{m\times n}$ can be also be realized as a Hilbert space with element-wise addition and scaling, along with the inner product

$$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}, \quad \langle \mathbf{X}, \mathbf{Y} \rangle = \operatorname{tr}(\mathbf{X}^T \mathbf{Y}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{X}_{i,j} \mathbf{Y}_{i,j}. \tag{4.2}$$

This inner product is simply the regular Euclidean inner product applied element wise to matrices, and induces the Frobenius norm $||\mathbf{X}||_{\mathrm{Fr}} = \sqrt{\operatorname{tr}(\mathbf{X}^T \mathbf{X})}$, $\mathbf{X} \in \mathbb{R}^{m\times n}$.

Any open set $\mathcal{M} \subset \mathcal{E}$ admits a differentiable structure inherited from the vector space $\mathcal{E}$, and intuitively we can reason that any chart $(\mathcal{U}, \varphi)$ over $\mathcal{E}$ is a chart over $\mathcal{M}$ if we restrict the domain of $\varphi$ to $(\mathcal{U} \cap \mathcal{M}, \varphi)$. Such a manifold $\mathcal{M} \subset \mathcal{E}$ is called an *open submanifold*.

**Definition 4.3** (Open submanifolds Boumal [25, Def. 3.6])**.** *Let $\mathcal{M}$ be an open subset of a vector space $\mathcal{E}$. Then $\mathcal{M}$ admits a smooth manifold structure, and we call it an* open submanifold. *If $\mathcal{M} = \mathcal{E}$, we call it a* linear manifold. *The set $\mathcal{E}$ is called the* embedding space *or the* ambient space *of $\mathcal{M}$.*

## 4.2 Differentials on Manifolds

For both scalar and manifold valued mappings the concept of differentiability of mappings is crucial to both the analysis of manifolds and the optimization of objective functions $f\colon \mathcal{M} \to \mathbb{R}$. We can view $\mathbb{R}$ as a vector space manifold in the sense of section 4.1, and as such we may view $f$ as a mapping between two manifolds as well. Here we'll first introduce a coordinate based definition of the differential of a mapping, and later we'll introduce the concept of tangent vectors which generalize directional derivatives of scalar valued functions on manifolds.

Using charts $(\mathcal{U}_1, \varphi_1)$, $\mathcal{U}_1 \subset \mathcal{M}_1$ and $(\mathcal{U}_2, \varphi_2)$, $\mathcal{U}_2 \subset \mathcal{M}_2$ we can define a coordinate representation of a mapping $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ as

$$\hat{F}\colon \varphi_1(\mathcal{U}_1) \to \varphi_2(\mathcal{U}_2), \quad \hat{F}(x) = \varphi_2 \circ F \circ \varphi_1^{-1}(x). \tag{4.3}$$

One can verify that this coordinate representation of $F$ is chart independent [23, p. 24], and therefore we can investigate properties about the original mapping $F$ from its coordinate representation. Specifically we can examine its differential properties through the charts. For the aforementioned mapping $F$ with its coordinate representation $\hat{F}$, we say that the mapping $F$ is *smooth at a point* $p \in \mathcal{M}$ if the chart differential of $\hat{F}$,

$$\mathrm{D}\hat{F}[\varphi_1(p)]\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}, \tag{4.4}$$

is smooth at $p$. Furthermore, we say that a mapping $F$ is *smooth* if it is smooth at all points in $\mathcal{M}_1$. As the chart differential of $\hat{F}$ is a linear operator between $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$ we can discuss its *rank* as in a linear algebra setting, where $\operatorname{rank}(F) = \dim(\operatorname{range}(\mathrm{D}\,\hat{F}[\varphi_1(p)]))$. With the *rank* of a mapping between two manifolds in mind, we define the concept of a *submersion* between two manifolds.

**Definition 4.4** (Submersion and Regular Values). *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be manifolds of dimension $d_1$ and $d_2$ respectively. Then a submersion between the two manifolds is a differentiable mapping $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ such that $\operatorname{rank}(F) = d_2$ for all points $p \in \mathcal{M}_1$. Additionally, a point $q \in \mathcal{M}_2$ is denoted a regular value of $F$ if $\operatorname{rank}(F) = d_2$ for all $p \in F^{-1}(q) = \{p \in \mathcal{M}_1 | F(p) = q\}$.*

Inspired by Boumal [25, Sec. 7.7], we can intuitively think of the full rank requirement on the mapping $F$ as ensuring that the gradients of the constraints defining the manifold through $F(p) = q$ remain linearly independent. In this sense, we require that the LICQ condition [14, Chap. 12] from classical optimization is satisfied at all points on our manifold $\mathcal{M}_1$, which generally keeps the solution domain of our optimization problems well-behaved. In order for the rank of a mapping $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ to be $d_2$ at any point $p \in \mathcal{M}_1$, we necessarily need $d_1 \geq d_2$.

When constructing new manifolds, it is common to view them as embedded in a higher dimensional manifold. For example, the elements of the 2-sphere manifold are the points in $\mathbb{R}^3$ with norm one. To be more precise about this notion, we introduce the concept of an *embedded submanifold*. Given two manifolds $(\mathcal{M}_1, \mathcal{A}_1^+)$, $(\mathcal{M}_2, \mathcal{A}_2^+)$ with $\mathcal{M}_1 \subset \mathcal{M}_2$ we say that $\mathcal{M}_1$ is an *embedded submanifold* of $\mathcal{M}_2$ if the manifold topology of $(\mathcal{M}_1, \mathcal{A}_1^+)$ coincides with the topology it inherits as a subspace of $\mathcal{M}_2$ [23, p. 25].

The concept of a submersion is highly useful when one is trying to characterize whether a set admits a manifold structure, as we can use the following theorem to characterize closed embedded submanifolds as the pre-image of a regular value under a constant rank differentiable mapping between two manifolds.

**Theorem 4.1** (Submersion Theorem [23, Prop. 3.3.3]). *Let $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ be a smooth mapping between manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ of dimensions $d_1$ and $d_2$ respectively with $d_1 > d_2$. Then if $q \in \mathcal{M}_2$ is a regular value of $F$, the pre-image $\mathcal{M} = F^{-1}(q)$ is a closed embedded submanifold of $\mathcal{M}_1$. Furthermore, $\dim(\mathcal{M}) = d_1 - d_2$.*

We note that a mapping $F\colon \mathcal{M}_1 \to \mathcal{M}_2$ is a submersion iff. $\mathrm{D}F(p)\colon T_p\mathcal{M}_1 \to T_{F(p)}\mathcal{M}_2$ is a surjection for all $p \in \mathcal{M}_1$ [23, p. 38].

## 4.3 Tangent Vectors

Given a curve $\gamma\colon \mathbb{R} \to \mathcal{M}$, the regular notion of a derivative of that curve at a time $t$ as defined by the limit

$$\gamma'(t) = \lim_{\tau \to 0} \frac{\gamma(t+\tau) - \gamma(t)}{\tau}, \tag{4.5}$$

is not applicable on manifolds in general, as they lack a vector space structure with addition and subtraction operations between two points $p, q \in \mathcal{M}$, as well as lacking scalar multiplication. This means that we need to introduce another manner of measuring derivatives of curves on manifolds, and we do this by instead computing derivatives of functions composed with the smooth curve in question. Given a manifold $\mathcal{M}$, a smooth function $\gamma\colon \mathbb{R} \to \mathcal{M}$, $t \mapsto \gamma(t)$ is called a *curve*. Composing such a curve $\gamma$ with a $C^\infty$ smooth real valued function $f\colon \mathcal{M} \to \mathbb{R}$ we get a well-defined function $f \circ \gamma\colon \mathbb{R} \to \mathbb{R}$ for which the classical derivative is well-defined [23, p. 33].

For $p \in \mathcal{M}$ let $\mathfrak{F}_p(\mathcal{M})$ denote the set of smooth functions on a neighborhood of $p$, and let $\mathfrak{F}(\mathcal{M})$ denote the set of smooth real valued functions on $\mathcal{M}$. Then given a curve $\gamma\colon \mathbb{R} \to \mathcal{M}$ with $\gamma(0) = p$ we define the derivative operator $\dot\gamma(0)\colon \mathfrak{F}_p(\mathcal{M}) \to \mathbb{R}$ as

$$\dot\gamma(0)f = \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))\bigg|_{t=0}, \quad f \in \mathfrak{F}_p(\mathcal{M}). \tag{4.6}$$

In anticipation of the fact that the space of derivative operators given by all curves passing through a point $p \in \mathcal{M}$ forms a vector space, we follow [23, p. 34] in defining tangent vectors at $p$.

**Definition 4.5** (Tangent Vectors [23, Def. 3.5.1])**.** *Let $p$ be a point on a manifold $\mathcal{M}$. Then a tangent vector $\xi_p$ is an operator $\xi_p\colon \mathfrak{F}_p(\mathcal{M}) \to \mathbb{R}$ for which there exists a curve $\gamma\colon \mathbb{R} \to \mathcal{M}$ with $\gamma(0) = p$ such that*

$$\xi_p f = \dot\gamma(0)\, f = \frac{d}{dt} f(\gamma(t))\bigg|_{t=0} \quad \forall\, f \in \mathfrak{F}_p(\mathcal{M}), \tag{4.7}$$

*in which case we say that $\gamma$ realizes the tangent vector $\xi_p$.*

Furthermore, the point $p$ is called the *anchor* of $\xi_p$, but the subscript $p$ is usually omitted if it's clear from the context what point is the anchor of a particular tangent vector. The collection of all tangent vectors at $p$ is denoted $T_p\mathcal{M}$. It can be shown that the space of tangent vectors $T_p\mathcal{M}$ at a point $p$ on a $d$-dimensional manifold $\mathcal{M}$ form a $d$-dimensional vector space, where addition and scaling of tangent vectors is defined as follows [23, p. 34]. Given $\alpha, \beta \in \mathbb{R}$ and $\gamma_1, \gamma_2$ smooth curves on a manifold $\mathcal{M}$ such that $\gamma_1(0) = \gamma_2(0) = p$, we define the result of scaling and adding two tangent vectors into another tangent vector $\dot\gamma(0) = \alpha\dot\gamma_1(0) + \beta\dot\gamma_2(0)$ by

$$[\alpha\dot\gamma_1(0) + \beta\dot\gamma_2(0)]f = \alpha(\dot\gamma_1(0)f) + \beta(\dot\gamma_2(0)f). \tag{4.8}$$

To show that there exists a curve $\gamma$ which realizes the tangent vector $\dot{\gamma}(0)$ at $p$ let $(\mathcal{U}, \varphi)$ be a chart about $p$ centered such that $\varphi(p) = 0$. Then define the local curve

$$\gamma(t) = \varphi^{-1}(\alpha\varphi(\gamma_1(t)) + \beta\varphi(\gamma_2(t))).$$

We see immediately that $\gamma(0) = \varphi^{-1}(0) = p$ as required, and using the chain rule on $f \circ \varphi^{-1} \colon \mathbb{R}^d \to \mathbb{R}$ we find that the curve $\gamma$ realizes the scaled and added tangent vectors $\gamma_1$ and $\gamma_2$ as required by the definition in (4.8).

We can also find a basis for the tangent space at $p$ using the chart $\varphi$. By defining the set of curves $\{\gamma_i\}_{i=1}^d, \gamma_i(t) = \varphi^{-1}(\varphi(p) + te_i)$ moving along each of the canonical coordinates $e_i \in \mathbb{R}^d$, the set of tangent vectors $\{\dot{\gamma}_i(0)\}_{i=1}^d$ form a basis for $T_p\mathcal{M}$ [23, p. 35]. To see this we read $f \colon \mathcal{M} \to \mathbb{R}$ through the chart as $\hat{f} = f \circ \varphi^{-1} \colon \mathbb{R}^d \to \mathbb{R}$ and note that for any $\xi \in T_p\mathcal{M}$ realized by the smooth curve $\gamma$,

$$\begin{aligned} \xi f &= \frac{\mathrm{d}}{\mathrm{d}t}\,(f \circ \varphi^{-1})(\varphi \circ \gamma(t))\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}\,(\hat{f}\,)(x_1(t), \ldots, x_d(t))\Big|_{t=0}, \\ &= \sum_{i=1}^{d} \dot{x}_i(0) \frac{\partial}{\partial x_i} \hat{f}(\varphi(\gamma_i)) = \sum_{i=1}^{d} \dot{x}_i(0)(\dot{\gamma}_i(0) f) = \left( \sum_{i=1}^{d} \dot{x}_i(0)\dot{\gamma}_i(0) \right) f. \end{aligned} \tag{4.9}$$

In the second-to-last equality above we've used that $\frac{\partial}{\partial x_i}(f \circ \varphi^{-1})(\varphi(\gamma_i)) = \dot{\gamma}_i f$. With the basis constructed above it can be shown that $T_p\mathcal{M}$ is a $d$-dimensional vector space [4]. For ease of notation we can denote the canonical basis $\{\dot{\gamma}_i(0)\}_{i=1}^d$ at $p$ as $\{(\partial_i)_p\}_{i=1}^d$, where we suppress the reliance on the curves $\{\gamma_i\}_{i=1}^d$ from above.

Collecting all the tangent vector spaces of a $d$-dimensional manifold $\mathcal{M}$ we define the *tangent bundle of* $\mathcal{M}$ [23, p. 36] as

$$T\mathcal{M} = \bigsqcup_{p \in \mathcal{M}} T_p\mathcal{M}. \tag{4.10}$$

As each $\xi \in T\mathcal{M}$ is an element in a single tangent vector space, the manifold $\mathcal{M}$ is a quotient of $T\mathcal{M}$ with the projection $\pi \colon T\mathcal{M} \to \mathcal{M},\ \xi_p \in T_p\mathcal{M} \mapsto p$. For each chart $(\mathcal{U}, \varphi)$ over $\mathcal{M}$ the mapping

$$\xi_p \in T_p\mathcal{M} \mapsto [\varphi_1(p), \ldots, \varphi_d(p), \xi_p\varphi_1, \ldots, \xi_p\varphi_d]^T \in \mathbb{R}^{2d}, \tag{4.11}$$

with the domain $\pi^{-1}(\mathcal{U})$ is a chart over $T\mathcal{M}$. Collecting these charts into an atlas one can show that it generates a manifold structure on $T\mathcal{M}$ [23, p. 36].

Another concept we can generalize to manifolds is the idea of *vector fields*, defining it as a smooth function from $\mathcal{M}$ to $T\mathcal{M}$ which assigns a tangent vector $X_p \in T_p\mathcal{M}$ to each point $p \in \mathcal{M}$ [23, p. 37]. We denote the set of all smooth vector fields over a manifold $\mathcal{M}$ by $\mathfrak{X}(\mathcal{M})$, where the evaluation of vector field $\xi$ at $p \in \mathcal{M}$ is denoted $\xi_p \in T_p\mathcal{M}$. A vector field can be applied to a smooth function $f \in \mathfrak{F}(\mathcal{M})$, returning a scalar function $\xi f \in \mathfrak{F}(\mathcal{M}),\ p \mapsto \xi_p(f) \in \mathbb{R}$. Multiplication of vector fields with smooth scalar valued functions is also defined, returning another vector field $(f\xi)_p = f(p)\xi_p$. Addition of two vector fields $\xi, \eta \in \mathfrak{X}(\mathcal{M})$ simply evaluates pointwise as $(\xi + \eta)_p = \xi_p + \eta_p \in T_p\mathcal{M}$.

We can provide a local basis for vector fields on a chart $(\mathcal{U}, \varphi)$ by defining the vector fields $\{E_i\}_{i=1}^d,\ E_i \in \mathfrak{X}(\mathcal{U})$ by

$$(E_i f)(p) := \frac{\partial}{\partial x_i}(f \circ \varphi^{-1})(\varphi(p)) = \mathrm{D}(f \circ \varphi^{-1})(\varphi(p))[e_i], \tag{4.12}$$

called the *$i$'th coordinate vector field* of $(\mathcal{U}, \varphi)$. The coordinate vector fields are smooth, and every vector field $\xi \in \mathfrak{X}(\mathcal{U})$ can be written as a linear combination of the coordinate vector fields as

$$\xi = \sum_{i=1}^{d} (\xi\varphi_i)E_i, \tag{4.13}$$

where $(\xi\varphi_i) \in \mathfrak{F}(\mathcal{U})$ [23, p. 37]. An important example of a vector field is the *velocity vector field* of a smooth curve $\gamma \colon \mathbb{R} \to \mathcal{M}$, denoted $\dot{\gamma} \colon \gamma(\mathbb{R}) \subset \mathcal{M} \to TM$. The composition $\dot{\gamma} \circ \gamma \colon \mathbb{R} \to TM,\ t \mapsto \dot{\gamma}_{\gamma(t)} \in T_{\gamma(t)}\mathcal{M}$ is denoted $\dot{\gamma}(t) \in T_{\gamma(t)}\mathcal{M}$ for ease of notation.

Using the concept of tangent vectors and tangent vector spaces we define an intrinsic notion of the differential of a mapping between manifolds $\mathcal{M}$ and $\mathcal{N}$.

**Definition 4.6** (Differential of a manifold mapping [23, p. 38]). *Given a mapping $F \colon \mathcal{M} \to \mathcal{N}$, we denote the differential of the mapping by $\mathrm{D}F \colon \mathcal{M} \to \mathcal{L}(T_p\mathcal{M}, T_{F(p)}\mathcal{N})$, $p \in \mathcal{M}$, the pointwise linear mapping*

$$\mathrm{D}F(p) \colon T_p\mathcal{M} \to T_{F(p)}\mathcal{N}, \quad \xi_p \mapsto \mathrm{D}F(p)[\xi_p], \tag{4.14}$$

*where for $f \in \mathfrak{F}_{F(p)}(\mathcal{N})$, the resulting tangent vector $\mathrm{D}F(p)[\xi_p] \in T_{F(p)}\mathcal{N}$ is applied to $f$ as $(\mathrm{D}F(p)[\xi_p])(f) = \xi_p(f \circ F)$. And for any curve $\gamma \colon \mathbb{R} \to \mathcal{M}$ which realizes $\xi_p \in T_p\mathcal{M}$, the tangent vector $\mathrm{D}F(p)[\xi] \in T_{F(p)}\mathcal{N}$ is realized by the curve $F \circ \gamma \colon \mathbb{R} \to \mathcal{N}$.*

Connecting back to Theorem 4.1, we note from Absil, Mahony, et al. [23, p. 38] that a mapping $F$ is a submersion iff. $\mathrm{D}F(p) \colon T_p\mathcal{M} \to T_{F(p)}\mathcal{N}$ is a surjection for all $p \in \mathcal{M}$. Furthermore, whenever a manifold $\mathcal{M}$ is constructed as the level-set of a submersion $F \colon \mathcal{M} \to \mathcal{N}$, i.e. $\mathcal{M} = F^{-1}(q),\ q \in \mathcal{N}$, we have for all curves $\gamma$ on $\mathcal{M}$ that $F \circ \gamma(t) = q \,\forall t \in \mathbb{R}$. This allows a characterization of the tangent space at $p \in \mathcal{M}$, as $\mathrm{D}F(p)[\dot{\gamma}] = \frac{\mathrm{d}}{\mathrm{d}t} F(\gamma(t))|_{t=0} = 0$, where $\gamma(0) = p$ realizes a tangent vector $\xi_p \in T_p\mathcal{M}$. This means that $T_p\mathcal{M} \subset \ker(\mathrm{D}F(p))$. As both $T_p\mathcal{M}$ and $\ker(\mathrm{D}F(p))$ are vector spaces and it can be shown that these spaces are of equal dimension [23, p. 40], we conclude that

$$T_p\mathcal{M} \cong \ker(\mathrm{D}F(p)), \tag{4.15}$$

for embedded submanifolds $\mathcal{M}$ defined as in Theorem 4.1. For linear manifolds $\mathcal{E}$, the notion of a classical derivative for a curve $\gamma \colon \mathbb{R} \to \mathcal{E}$ is well-defined in the embedding space, and is denoted $\gamma'(0) = \lim_{t\to 0} \frac{1}{t}(\gamma(t) - p) \in \mathcal{E}$ with $\gamma(0) = p$. For the tangent vector $\xi_p$ realized by $\gamma$ we have a one-to-one correspondence

$$\xi_p f = \mathrm{D}f(p)[\gamma'(0)], \tag{4.16}$$

for $f \in \mathfrak{F}_p(\mathcal{E})$, independent of which curve $\gamma$ realizes $\xi_p$, meaning that $T_p\mathcal{E} \cong \mathcal{E}$ [23, p. 35]. And if $\mathcal{M}$ is an open submanifold of $\mathcal{E}$ the property extends to

$$T_p\mathcal{M} = T_p\mathcal{E} \cong \mathcal{E}, \tag{4.17}$$

as stated by Boumal [25, Thrm. 3.15].

## 4.4 The Riemannian Metric

The tangent vector space about a point $p \in \mathcal{M}$ can be interpreted as a vector space approximation of our manifold about $p$, but without a norm or inner product $T_p\mathcal{M}$ we lack a lot of useful structure which Hilbert spaces enjoy. However, we can introduce such a structure by endowing the manifold $\mathcal{M}$ with a *Riemannian metric*, denoted $g$. A Riemannian metric $g$ defines an inner product, i.e. a bilinear, symmetric, positive definite form

$$g_p \colon T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R} \tag{4.18}$$

for all $p \in \mathcal{M}$, smooth w.r.t. $p$ [4, Def. 2.1][23, Chap. 3.6]. By the Riemannian metric being smooth w.r.t. $p$ we mean that for any open set $\mathcal{U} \subset \mathcal{M}$ and smooth vector fields $\xi, \eta \in \mathfrak{X}(\mathcal{U})$, the function

$$g(\xi, \eta) \colon \mathcal{M} \to \mathbb{R},\ p \mapsto g_p(\xi_p, \eta_p) \tag{4.19}$$

is smooth, i.e. $g(\xi, \eta) \in \mathfrak{F}(\mathcal{M})$ [4, p. 38]. Any manifold $(\mathcal{M}, \mathcal{A}^+)$ admits a Riemannian structure [23, p. 45], and combining a manifold $(\mathcal{M}, \mathcal{A}^+)$ with the additional structure of a Riemannian metric $g$ we get what is called a *Riemannian manifold* $(\mathcal{M}, \mathcal{A}^+, g)$. In this thesis we'll only be concerned with this class of manifolds, and when the metric $g$ is implied, we'll simply say the *Riemannian manifold* $\mathcal{M}$ for $(\mathcal{M}, \mathcal{A}^+, g)$

With the Riemannian metric defining an inner product $g_p$ at each $p \in \mathcal{M}$, it induces a norm at each tangent space, i.e. $||\xi_p||_g = \sqrt{g_p(\xi_p, \xi_p)}$. The action of the inner product on $\xi, \zeta \in T_p\mathcal{M}$ is denoted $g_p(\xi, \zeta)$ or $\langle \xi, \zeta \rangle_p$. In a chart $(\mathcal{U}, \varphi)$ we can express the Riemannian metric component wise by the functions

$$g_{i,j} = g(E_i, E_j) \in \mathfrak{F}(\mathcal{U}),\ i, j = 1, \dots, d, \tag{4.20}$$

where $\{E_i\}_{i=1}^d$ are the chart coordinate vector fields from Eq. (4.12). For vector fields $\xi = \sum_{i=1}^d \xi_i E_i$, $\eta = \sum_{j=1}^d \eta_j E_j$ the application of the metric $g$ is then expressed as

$$g(\xi, \eta) = \sum_{i,j=1}^{d} \xi_i g(E_i, E_j)\eta_j = \hat{\xi}^T \, \mathbf{G} \, \hat{\eta}, \tag{4.21}$$

where $\mathbf{G} \colon \mathcal{U} \to \mathbb{R}^{d\times d}$ is the smooth matrix valued function with $(\mathbf{G})_{i,j} = g(E_i, E_j)$ and $\hat{\xi} = [\xi_1, \ldots, \xi_d]$, $\hat{\eta} = [\eta_1, \ldots, \eta_d]$. If vector space manifolds $\mathcal{E}$ with an inner product $\langle \cdot, \cdot \rangle$ are imbued with the flat Riemannian metric $\bar{g}$ we call the space *Euclidean*, and $\bar{\mathbf{G}} = I_d$. The action of the Riemannian metric in Euclidean spaces further reduces to $\bar{g}_p(\xi, \zeta) = \xi^T \bar{\mathbf{G}} \zeta = \langle \xi, \zeta \rangle$ for $\xi, \zeta \in T_p\mathcal{E} \cong \mathcal{E}$.

The structure of a Riemannian metric allows for a natural definition of the *Riemannian gradient* of a smooth scalar function $f \colon \mathcal{M} \to \mathbb{R}$. For each $p \in \mathcal{M}$ the differential $\mathrm{D}f(p)$ is a linear functional on the Hilbert space $(T_p\mathcal{M}, g_p)$, and the gradient $\mathrm{grad} f(p)$ is defined as the unique element in $T_p\mathcal{M}$ s.t.

$$\langle \mathrm{grad} f(p), \xi \rangle_p = \mathrm{D}f(p)[\xi], \; \forall \, \xi \in T_p\mathcal{M}, \tag{4.22}$$

the existence and uniqueness of which follows from Riesz' representation theorem [23, p. 46]. The Riemannian gradient enjoys some of the same properties as the regular Euclidean gradient. It is the direction of the steepest ascent for $f(p)$ in the sense that

$$\frac{\mathrm{grad} f(p)}{||\mathrm{grad} f(p)||_p} = \underset{\xi \in T_p\mathcal{M}, ||\xi||_p = 1}{\arg\max} \; \mathrm{D}f(p)[\xi], \tag{4.23}$$

and its norm $||\mathrm{grad} f(p)||_p$ gives the magnitude of the steepest slope of $f$ at $p$.

## 4.5 Geodesics

If an iterative process finds itself at a point $p \in \mathcal{M}$ on a Riemannian metric $\mathcal{M}$, we would like to be able to move along a direction given by a certain tangent vector $\xi_p \in T_p\mathcal{M}$, to e.g. reduce an objective function. For vector spaces we define a straight line from $p \in \mathbb{R}^d$ along $\xi \in T_p\mathbb{R}^d \cong \mathbb{R}^d$ by the curve $\gamma(t) = p + t\xi$, and for manifolds the analogue to moving along a straight line is the *geodesic*. Before presenting the definition of a geodesic we'll briefly present the ideas of an *affine connection* and the *covariant derivative along a curve* on $\mathcal{M}$.

First we state what an affine connection is, following do Carmo [4]:

**Definition 4.7** (Affine Connection [4, Def. 2.1]). *An affine connection* $\nabla$ *on a smooth manifold* $\mathcal{M}$ *is a mapping*

$$\nabla \colon \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M}), \tag{4.24}$$

*denoted* $(\xi, \zeta) \mapsto \nabla_\xi \zeta$*, and which satisfies the following properties:*

1. $\nabla_{f\xi+g\zeta}\eta = f\nabla_\xi \eta + g\nabla_\zeta \eta$,
2. $\nabla_\xi(\zeta+\eta) = \nabla_\xi \zeta + \nabla_\xi \eta$,
3. $\nabla_\xi(f\zeta) = f\nabla_\xi \zeta + \xi(f)\zeta$,

*in which* $\xi, \zeta, \eta \in \mathfrak{X}(\mathcal{M})$ *and* $f \in \mathfrak{F}(\mathcal{M})$.

Intuitively, the affine connection generalizes derivatives of a vector field $\zeta$ along the vector field $\xi$, as the idea used for defining tangent vectors by composing scalar functions with smooth curves $\gamma\colon \mathbb{R} \to \mathcal{M}$ does not work for vector fields $\xi \in \mathfrak{X}(\mathcal{M})$. We can see this by "computing"

$$\left.\frac{\mathrm{d}\xi_{\gamma(t)}}{\mathrm{d}t}\right|_{t=0} = \lim_{t\to 0} \frac{\xi_{\gamma(t)} - \xi_{\gamma(0)}}{t}, \tag{4.25}$$

which is not defined in general as $\xi_{\gamma(t)} \in T_{\gamma(t)}$ and $\xi_{\gamma(0)} \in T_{\gamma(0)}$ are elements of different vector spaces [23, p. 93]. Any affine connection also induces a covariant derivative of vector fields along smooth curves.

**Definition 4.8** (Covariant derivative [4, Prop. 2.2]). *Let* $\mathcal{M}$ *be a smooth manifold with an affine connection* $\nabla$*. Then there exists a unique correspondence which associates to a vector field* $\xi\colon I \to TM$ *along the differentiable curve* $\gamma\colon I \subset \mathbb{R} \to \mathcal{M}$ *(i.e.* $\xi(t) \in T_{\gamma(t)}\mathcal{M}$*) another vector field* $\frac{\mathrm{D}\xi}{\mathrm{d}t}$ *along* $\gamma$*, called the* covariant derivative of $\xi$ along $\gamma$*, such that:*

1. $\frac{\mathrm{D}}{\mathrm{d}t}(\xi+\zeta) = \frac{\mathrm{D}\xi}{\mathrm{d}t} + \frac{\mathrm{D}\zeta}{\mathrm{d}t}$,
2. $\frac{\mathrm{D}}{\mathrm{d}t}(f\xi) = \frac{\mathrm{d}f}{\mathrm{d}t}\xi + f\frac{\mathrm{D}\xi}{\mathrm{d}t}$,
3. *If* $\xi$ *is induced by a vector field* $\eta \in \mathfrak{X}(\mathcal{M})$*, i.e.* $\xi(t) = \eta(\gamma(t))$*, then* $\frac{\mathrm{D}\xi}{\mathrm{d}t} = \nabla_{\dot\gamma}\eta$,

*for* $\xi, \zeta\colon I \to TM$ *and* $f\colon I \to \mathbb{R}$ *smooth.*

Consider a chart $(\mathcal{U}, \varphi)$ of $\mathcal{M}$ for which $\gamma(I) \cap \mathcal{U} \neq \emptyset$, $\varphi(\gamma(t)) = [x_1(t), \dots, x_d(t)]^T$ is a local coordinate representation of the smooth curve $\gamma\colon I \subset \mathbb{R} \to \mathcal{M}$. Any vector field $\xi$ along $\gamma$ can then be expanded as

$$\xi(t) = \sum_{i=1}^{d} \xi_i(t) E_i(t), \tag{4.26}$$

for $\xi_i \in C^\infty(I)$ and $\{E_i(t) = (E_i)_{\gamma(t)}\}_{i=1}^d$ the coordinate vector fields of $(\mathcal{U}, \varphi)$. In this chart the velocity vector field of $\gamma$ can be expressed as $\dot\gamma(t) = \sum_{i=1}^d \frac{\mathrm{d}x_i(t)}{\mathrm{d}t} E_i(t)$, and the covariant derivate of $E_i(t)$ can be computed using the third property in the definition of the covariant derivative (4.8),

$$\frac{\mathrm{D}E_i(t)}{\mathrm{d}t} = \nabla_{\dot\gamma} E_i(t) = \nabla_{\Sigma\frac{\mathrm{d}x_j(t)}{\mathrm{d}t}E_i(t)} E_i(t) = \sum_{j=1}^{d} \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} \nabla_{E_j(t)} E_i(t). \tag{4.27}$$

If we expand the $d^2$ smooth vector fields $\nabla_{E_j} E_i, i, j = 1, \ldots, d$ in the same basis $\{E_i\}_{i=1}^d$ independent of $t$ as

$$\nabla_{E_j} E_i = \sum_{k=1}^{d} \Gamma_{j,i}^k E_k, \tag{4.28}$$

the chart-dependent coefficient functions $\Gamma_{j,i}^k \in \mathfrak{F}(\mathcal{U})$ are called the *Christoffel symbols* [4, p. 52], and they fully define the connection $\nabla$ in each chart [23, p. 95].

Continuing, $\frac{\mathrm{D}\xi}{\mathrm{d}t}$ can be computed in the chart using properties of the affine connection and the covariant derivative [4, p. 51]:

$$\begin{aligned}
\frac{\mathrm{D}\xi}{\mathrm{d}t} &= \sum_{i=1}^{d} \frac{\mathrm{d}\xi_i(t)}{\mathrm{d}t} E_i(t) + \sum_{i=1}^{d} \xi_i(t) \frac{\mathrm{D}E_i(t)}{\mathrm{d}t}, \\
&= \sum_{i=1}^{d} \frac{\mathrm{d}\xi_i(t)}{\mathrm{d}t} E_i(t) + \sum_{i,j=1}^{d} \xi_i(t) \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} \nabla_{E_j(t)} E_j(t), \\
&= \sum_{k=1} \left( \frac{\mathrm{d}\xi_k(t)}{\mathrm{d}t} + \sum_{i,j=1}^{d} \xi_i(t) \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} \Gamma_{j,i}^k(\gamma(t)) \right) E_k(t),
\end{aligned} \tag{4.29}$$

where we've renamed the $i$ index with $k$ in the first sum on the last line. Intuitively, the covariant derivative takes into account the fact that the basis vectors $\{E_i(t)\}_{i=1}^d$ change from point to point along the curve. On a flat vector space manifold $\mathcal{E}$, the Christoffel symbols all vanish, as the tangent vector space at all points $p \in \mathcal{E}$ can be identified with $\mathcal{E}$, and is spanned by $\{e_i\}_{i=1}^d$ independent of the time $t$. Then, the covariant derivative reduces to the regular second time derivative of a curve in a vector space $\mathcal{E}$.

The choice of an affine connection $\nabla$ on a Riemannian manifold $\mathcal{M}$ is in principle free, but there exists a unique preferred connection with certain properties called the *Riemannian* connection or the *Levi-Civita* connection.

**Definition 4.9** (Riemannian (Levi-Civita) Connection [4, Thrm. 3.6])**.** *Given a Riemannian manifold $(\mathcal{M}, \mathcal{A}^+, g)$, there exists a unique affine connection $\nabla$ on $\mathcal{M}$ satisfying the conditions:*

1. *$\nabla$ is* symmetric*, meaning that its associated Christoffel symbols are symmetric in the lower indices, $\Gamma_{i,j}^k = \Gamma_{j,i}^k \;\forall\, i, j, k,$ in all charts $(\mathcal{U}, \varphi) \in \mathcal{A}^+$.*
2. *$\nabla$ is* compatible *with the Riemannian metric $g$, meaning that for any two vector fields $\xi, \zeta$ along a smooth curve $\gamma \colon I \to \mathcal{M}$ we have*

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle \xi, \zeta \rangle_{\gamma(t)} = \left\langle \frac{\mathrm{D}\xi}{\mathrm{d}t}, \zeta \right\rangle_{\gamma(t)} + \left\langle \xi, \frac{\mathrm{D}\zeta}{\mathrm{d}t} \right\rangle_{\gamma(t)}. \tag{4.30}$$

In this work we'll always be using the Levi-Civita connection on the Riemannian manifolds we work with. And finally we are ready to state the definition of a geodesic as a curve with a vanishing *acceleration vector field* $\frac{\mathrm{D}^2}{\mathrm{d}t^2}\gamma = \frac{\mathrm{D}}{\mathrm{d}t}\dot{\gamma}$ [23, p. 102].

**Definition 4.10** (Geodesic [4, Def. 2.1]). *Let $\gamma\colon [t_0, t_1] \to \mathcal{M}$ be a smooth curve on the manifold $\mathcal{M}$. Then, $\gamma$ is a* geodesic at $t' \in [t_0, t_1]$ *if*

$$\frac{D}{\mathrm{d}t}\dot{\gamma}\bigg|_{t=t'} = \nabla_{\dot{\gamma}}\dot{\gamma} = 0, \tag{4.31}$$

*If the curve $\gamma$ is geodesic at all points $t' \in [t_0, t_1]$, we say that $\gamma$ is a* geodesic *connecting $\gamma(t_0)$ and $\gamma(t_1)$.*

Concerning the existence and uniqueness of such geodesics given an initial point and tangent vector, we refer to do Carmo [4, Proposition 2.5].

**Theorem 4.2** (Local Geodesic Existence and Uniqueness do Carmo [4, Prop. 2.5]). *Given $p \in \mathcal{M}$, there exists a neighborhood $\mathcal{V} \subset \mathcal{M}$ of $p$, $\delta > 0$, $\epsilon > 0$ and a $C^\infty$ mapping*

$$\gamma\colon (-\delta, \delta) \times \mathcal{U}_\epsilon \to \mathcal{M},\ \mathcal{U}_\epsilon = \{(q, \zeta) \mid q \in V, \zeta \in T_q\mathcal{M},\ ||\zeta||_q < \epsilon\} \tag{4.32}$$

*such that the curve $t \mapsto \gamma(t, q, \zeta),\ t \in (-\delta, \delta)$ defines the unique geodesic passing through $q$ at $t = 0$ with tangent vector $\zeta$, for each $(q, \zeta) \in \mathcal{U}_\epsilon$.*

The proof of the above theorem relies on the existence and uniqueness of solutions to systems of ODE's. If we write out the requirement for a curve $\gamma\colon I \subset \mathbb{R} \to \mathcal{M}$ to be a geodesic in a chart $(\mathcal{U}, \varphi)$ as was done in Eq. (4.29) with $\varphi(\gamma(t)) = \boldsymbol{x}(t) = [x_1(t), \dots, x_d(t)]^T \in \mathbb{R}^d$, $\gamma$ is a geodesic iff.

$$0 = \frac{\mathrm{D}}{\mathrm{d}t}(\dot{\gamma}) = \sum_{k=1}\left(\frac{\mathrm{d}^2 x_k(t)}{\mathrm{d}t^2} + \sum_{i,j=1}^{d} \frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\frac{\mathrm{d}x_j(t)}{\mathrm{d}t}\hat{\Gamma}^k_{j,i}(\boldsymbol{x}(t))\right) E_k(t), \tag{4.33}$$

where $\hat{\Gamma}^k_{j,i} = \Gamma^k_{j,i} \circ \varphi^{-1}$. To construct a geodesic $\gamma$ then, do Carmo [4, pp. 61-63] demonstrates that we only need to prove the existence and uniqueness of a solution to the coupled second order system of ODE's

$$\frac{\mathrm{d}^2 x_k(t)}{\mathrm{d}t^2} + \sum_{i,j=1}^{d} \frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\frac{\mathrm{d}x_j(t)}{\mathrm{d}t}\hat{\Gamma}^k_{j,i}(\boldsymbol{x}(t)) = 0, \quad k = 1, \dots, d, \tag{4.34}$$

for $t \in (-\delta, \delta)$, with initial conditions $\varphi(p) = \boldsymbol{x}(0)$, $\dot{\gamma}(0) = \sum_{i=1}^{d} \frac{\mathrm{d}x_i(t)}{\mathrm{d}t} E_i(0)$.

## 4.6 Exponential and Logarithmic Mapping

Theorem 4.2 guarantees the existence and uniqueness of geodesics locally about a point $p$, with a trade-off between the magnitude of the tangent vector $\zeta$ at $t = 0$ and the time domain of the geodesic. This trade-off can be formalized into the *homogeneity principle* as formulated by do Carmo [4, p. 64] which states that if a geodesic $\gamma(t, p, \xi)$ is defined on an interval $(-\delta, \delta)$, then the geodesic $\gamma(t, p, a\xi),\ a > 0$ is defined on the interval $(-\delta/a, \delta/a)$ and

$$\gamma(at, p, \xi) = \gamma(t, p, a\xi). \tag{4.35}$$

The above property motivates the definition of a map returning the result of moving for a time $t = 1$ along the geodesic given by an initial point $p \in \mathcal{M}$ and an initial tangent vector $\xi \in T_p\mathcal{M}$. This map is called the *exponential map*, defined by

$$\operatorname{Exp}\colon \mathcal{U}_\epsilon \to \mathcal{M},\ \operatorname{Exp}(p, \zeta) = \gamma(1, p, \zeta) = \gamma(||\zeta||_p, p, \zeta/||\zeta||_p),\ \epsilon = \delta/2, \tag{4.36}$$

where $\mathcal{U}_\epsilon$ is as defined in Proposition 4.2, limiting the tangent vector norms to $||\zeta|| < \delta/2$ to ensure that the geodesic curve is defined for times $|t| \leq 1$ [4, p. 65]. Furthermore, we may restrict the exponential map to a point $p \in \mathcal{M}$ and view the exponential map as a mapping between $T_p\mathcal{M}$ and $\mathcal{M}$,

$$\operatorname{Exp}_p\colon B_{\epsilon_1}(0_p) \subset T_p\mathcal{M} \to \mathcal{M},\ \operatorname{Exp}_p(\zeta) = \operatorname{Exp}(p, \zeta), \tag{4.37}$$

where $B_r(\zeta)$ is the open ball of radius $r$ about $\zeta \in T_p\mathcal{M}$. Other immediate properties of the exponential map are that $\operatorname{Exp}_p(0_p) = p$, and by its definition through the $C^\infty$ mapping $\gamma\colon (-\delta, \delta) \times \mathcal{U}_\epsilon$ the exponential map is smooth. In the same manner as do Carmo [4, p. 65], we can also compute the differential of the exponential map $\operatorname{Exp}_p$ at the origin $0_p \in T_p\mathcal{M}$ using the canonical identification $T_{0_p}(T_p\mathcal{M}) \cong T_p\mathcal{M}$, giving

$$\begin{aligned} \mathrm{D}(\operatorname{Exp}_p)(0_p)[\zeta] &= \frac{\mathrm{d}}{\mathrm{d}t}(\operatorname{Exp}_p(t\zeta))\bigg|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}(\gamma(1, p, t\zeta))\bigg|_{t=0}, \\ &= \frac{\mathrm{d}}{\mathrm{d}t}(\gamma(t, p, \zeta))\bigg|_{t=0} = \zeta, \end{aligned} \tag{4.38}$$

where the last equality follows from the definition of the mapping $\gamma(t, q, \zeta)$. In conclusion, the differential of the exponential map at the origin is the identity map, $\mathrm{D}(\operatorname{Exp}_p)_{0_p} = \mathrm{Id}_{T_p\mathcal{M}}$. By the inverse function theorem this allows us to conclude that $\operatorname{Exp}_p$ is a local diffeomorphism of $B_{\epsilon_1}(0_p)$ onto an open subset $\mathcal{V} \subset \mathcal{M}$ containing $p$ [4, Prop. 2.9]. The image $\operatorname{Exp}_p(B_\epsilon(0_p)) \subset \mathcal{M}$ of such a ball $B_\epsilon(0_p) \subset T_p\mathcal{M}$, given an $\epsilon > 0$ for which $\operatorname{Exp}_p\colon B_\epsilon(0_p) \to \mathcal{M}$ is a diffeomorphism onto its image, is called a *geodesic ball* in $\mathcal{M}$ [26, p. 158].

The inverse of the exponential mapping is called the *logarithmic mapping*, denoted

$$\operatorname{Log}_p\colon \mathcal{V}_p \to T_p\mathcal{M}. \tag{4.39}$$

Its domain $\mathcal{V}_p$ is the image of the largest subset $\tilde{\mathcal{C}} \subset T_p\mathcal{M}$ containing $0_p$ for which $\operatorname{Exp}_p\colon \tilde{\mathcal{C}} \to \mathcal{M}$ is a diffeomorphism onto its image. Points $q \in \mathcal{V}$ have a unique representation $\operatorname{Log}_p(q) = \tilde{q} \in T_p\mathcal{M}$ in the tangent space of $p$, so that $\exp_p(\tilde{q}) = q$.

## 4.7 Riemannian Distance

Given a Riemannian manifold $(\mathcal{M}, g)$ we measure the length of a smooth curve $\gamma\colon I \subset \mathbb{R} \to \mathcal{M}$ by integrating the norm of its velocity vector field $\dot{\gamma}(t)$ over its domain [26, p. 34],

$$L_g(\gamma) = \int_{t_0}^{t_1} ||\dot{\gamma}(t)||_g \,\mathrm{d}t, = \int_{t_0}^{t_1} \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}} \,\mathrm{d}t, \tag{4.40}$$

analogous to the definition of the length of a parametrized curve in $\mathbb{R}^d$. We can use the minimum length of all curves connecting two points on a manifold as a "measure tape" to define the distance between two points, but in order to be able to define a metric space structure on a Riemannian manifold $\mathcal{M}$ we require that it be *path-connected*.

A manifold $\mathcal{M}$ is path-connected if for any $p, q \in \mathcal{M}$, there exists continuous curves

$$\gamma_{p\to q}\colon [t_0, t_1] \to \mathcal{M}, \text{ s.t. } \gamma_{p\to q}(t_0) = p, \; \gamma_{p\to q}(t_1) = p, \tag{4.41}$$

connecting the two points. A curve $\gamma\colon [a, b] \to \mathcal{M}$ connecting two points $p, q \in \mathcal{M}$, in the sense that $\gamma(a) = p$ and $\gamma(b) = q$, is called *admissible* if there exists a partition $a = a_0 < a_1 < \cdots < a_k = b$, $k \in \mathbb{N}$ s.t. $\gamma|_{[a_{i-1}, a_i]}\colon [a_{i-1}, a_i] \to \mathcal{M}, i = 1, \ldots, k$ is smooth with nonvanishing velocity [26, pp. 33-34]. If $\mathcal{M}$ is a path-connected manifold, then any two points $p, q \in \mathcal{M}$ can be connected by an admissible curve [26, Prop. 2.50]. Restricting the "measuring tapes" to admissible connecting curves, we define the *Riemannian distance* between two points:

**Definition 4.11** (Riemannian distance [26, p. 36]). *Let $(\mathcal{M}, g)$ be a path-connected Riemannian manifold. The* Riemannian distance $\mathrm{dist}\colon \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ *is defined as*

$$\mathrm{dist}(p, q) = \inf_{\gamma_{p\mapsto q}} L_g(\gamma_{p\mapsto q}), \tag{4.42}$$

*where $\gamma_{p\mapsto q}$ is any admissible curve connecting $p$ and $q$.*

With the metric function $\mathrm{dist}\colon \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ defined above, the path-connected Riemannian manifold $\mathcal{M}$ is a metric space whose metric topology is the same as the atlas topology [26, Thrm. 2.55].

Any curve $\gamma\colon [t_0, t_1] \to \mathcal{M}$ connecting $p$ and $q$ in $\mathcal{M}$ and for which $\mathrm{dist}(p, q) = L_g(\gamma)$ is called a *minimizing curve connecting $p$ and $q$*. If $q \in \mathcal{M}$ is contained in a geodesic ball about $p \in \mathcal{M}$, then the geodesic curve $t \mapsto \exp_p(t\xi), \; t \in [0, 1]$ with $\xi = \log_p(q)$ is the unique minimizing curve connecting $p$ to $q$, up to a reparametrization in $t$ due to the homogeneity principle [26, Prop. 6.11].

We say that a Riemannian manifold $\mathcal{M}$ is *metrically complete* if every Cauchy sequence on $\mathcal{M}$ converges. It turns out that this notion of completeness is equivalent to the concept of a manifold being *geodesically complete*, in the sense that the exponential mapping is defined on the entire tangent bundle $T\mathcal{M}$ [26, Thrm. 6.19]. As these notions of completeness are equivalent, we say that a Riemannian manifold $\mathcal{M}$ is *complete* if either of the above notions of completeness hold. The Riemannian distance function is particularly well-behaved on complete Riemannian manifolds, as if $\mathcal{M}$ is a complete, path-connected Riemannian manifold, any two points $p, q \in \mathcal{M}$ can be connected by a minimizing geodesic [26, Cor. 6.21].

For a geodesic $\gamma$, the norm of the velocity field tangent vectors of $\dot{\gamma}$ are constant along the path, as

$$\frac{\mathrm{d}}{\mathrm{d}t}||\dot{\gamma}(t)||_g^2 = \frac{\mathrm{d}}{\mathrm{d}t}\langle\dot{\gamma}(t), \dot{\gamma}(t)\rangle = 2\left\langle \frac{\mathrm{D}}{\mathrm{d}t}\dot{\gamma}(t), \dot{\gamma}(t)\right\rangle = 0. \tag{4.43}$$

This means that for a geodesic $\gamma\colon [0,1] \to \mathcal{M}$ starting at $\gamma(0) = p$ with $\dot{\gamma}(0) = \xi_p$, the length of the curve is given by

$$L(\gamma) = \int_0^1 \sqrt{||\dot{\gamma}(t)||_g^2}\,\mathrm{d}t = \int_0^1 \sqrt{||\dot{\gamma}(0)||_g^2}\,\mathrm{d}t = ||\xi_p||. \tag{4.44}$$

This fact lends itself to another interpretation of $\mathrm{Exp}_p(\xi)$ as moving a distance of $||\xi||_p$ away from $p$ along the geodesic curve passing through $p$ with tangent vector $\xi$, as long as $||\xi||_p$ is small enough for $\mathrm{Exp}_p(\xi)$ to be well-defined. And the Riemannian distance $p$ and $q$ contained in geodesic ball about $p$ can be computed as

$$\mathrm{dist}(p, q) = ||\log_p(q)||_p. \tag{4.45}$$

For a point $p$ on the Riemannian manifold $\mathcal{M}$, let $\tilde{\mathcal{C}}_p \subset T_p\mathcal{M}$ be the largest subset of $T_p\mathcal{M}$ containing the origin $0_p$ for which $\mathrm{Exp}_p\colon \tilde{\mathcal{C}}_p \to \mathrm{Exp}_p(\tilde{\mathcal{C}}_p)$ is a diffeomorphism. Then image of the boundary $\partial\tilde{\mathcal{C}}$ under the exponential mapping is called the *cut locus of* $p$, denoted $\mathcal{C}_p = \mathrm{Exp}_p(\partial\tilde{\mathcal{C}}_p) \subset \mathcal{M}$ [27, Sec. 2.1.3]. The Riemannian distance between a point $p \in \mathcal{M}$ and its cut locus,

$$\mathrm{dist}(p, \mathcal{C}_p) = \inf_{q \in \mathcal{C}_p} \mathrm{dist}(p, q), \tag{4.46}$$

is called the *injectivity radius of* $p$, denoted $\mathrm{inj}(p)$. Furthermore, we define the injectivity radius of a manifold $\mathcal{M}$ as the infimum over the injectivity radii of all its points, i.e. $\mathrm{inj}(\mathcal{M}) = \inf_{p\in\mathcal{M}} \mathrm{inj}(p)$ [27, p. 4].

For the class of *Hadamard* manifolds an even stronger result holds regarding the existence of minimizing geodesics. One way of defining a Hadamard manifold is through its distance function:

**Definition 4.12** (Hadamard manifold [28, Thrm. 1.3.2]). *Let $\mathcal{M}$ be a complete, path-connected Riemannian manifold. Then $\mathcal{M}$ is* Hadamard *if for every pair $p, q \in \mathcal{M}$ there exists a point $m \in \mathcal{M}$ such that*

$$\mathrm{dist}^2(m, z) + \frac{1}{4}\mathrm{dist}^2(p, q) \le \frac{1}{2}\mathrm{dist}^2(p, z) + \frac{1}{2}\mathrm{dist}^2(z, q),\ \forall\ z \in \mathcal{M}. \tag{4.47}$$

*In which case $m$ is the midpoint between $p$ and $q$, with* $\mathrm{dist}(p, m) = \mathrm{dist}(m, q) = \mathrm{dist}(p, q)/2$.

And for Hadamard manifolds $\mathcal{M}$, any two points $p, q \in \mathcal{M}$ are connected by a unique minimizing geodesic, meaning the logarithm mapping between them is well-defined, and $\mathrm{inj}(\mathcal{M})$ is infinite [28, p. 2].

In practice, we often work with the squared distance function due to its differentiability properties. The squared distance function with one argument fixed

$$\mathrm{dist}^2(p, \cdot)\colon \mathcal{M} \to \mathbb{R}^+, \ q \mapsto \mathrm{dist}^2(p, q), \tag{4.48}$$

is $C^\infty$ smooth on $\mathcal{M} \setminus \mathcal{C}_p$ [27, pp. 4-6], with Riemannian gradient

$$\mathrm{grad}_q \, \mathrm{dist}(p, q)^2 = -2 \log_p(q), \tag{4.49}$$

as long as $q$ is within the injectivity radius of $p$.

## 4.8 Riemannian Center of Mass

In vector spaces, the mean of a set of points, e.g. $\{x_i\}_{i=1}^N \subset \mathbb{R}^d$ is defined as $\bar{x} = (1/N) \sum_{i=1}^N x_i$, but this summation operation followed by multiplication with $N^{-1}$ is in general not possible for manifold valued points. The mean, or *Riemannian center of mass*, of a set of points $\{q_i\}_{i=1}^N \subset \mathcal{M}$ can instead be defined as the solution(s) to the following minimization problem [27, Def. 2.5]:

$$\bar{\mu} = \operatorname*{argmin}_{p \in \mathcal{M}} \mu(p) := \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \mathrm{dist}(p, q_i)^2. \tag{4.50}$$

In Euclidean spaces, the unique minimizer of Eq. (4.50) is the aforementioned $\bar{x}$, whilst on general Riemannian manifolds there may be several local minimizers of the objective function $\mu(p)$. However, the Riemannian center of mass is unique if all the points $\{q_i\}_{i=1}^N$ in question are contained in a metric ball

$$B(o, \rho) = \{p \in \mathcal{M} \mid \mathrm{dist}(o, p) < \rho\}, \tag{4.51}$$

about $o \in \mathcal{M}$ and with a radius $\rho > 0$ depending on the manifold [27, Thrm. 2.6].

## 4.9 Line Search Optimization

With the structure of a Riemannian metric $g$ on a smooth manifold $\mathcal{M}$, as well as the concept of geodesics along tangent vectors we can generalize the idea of line search methods for minimizing smooth scalar objective functions $f\colon \mathcal{M} \to \mathbb{R}$. Intuitively, we can find the direction of the steepest ascent of $f$ at $p \in \mathcal{M}$ through its gradient $\mathrm{grad} f(p)$, and subsequently take steps along geodesics in the steepest descent direction $-\operatorname{grad} f(p)$ by means of the exponential mapping $\exp_p(-t \cdot \operatorname{grad} f(p)),\ t > 0$.

Even though it is desirable to step along geodesics by means of the exponential mapping, it can be computationally expensive to compute, or theoretically intractable to solve for the exponential mapping in a closed form expression given the Riemannian metric $g$. To generalize the action of stepping from a point along a certain tangent vector we introduce the concept of a *retraction*.

**Definition 4.13** (Retraction [23, Def. 4.1.1])**.** *Given a manifold $\mathcal{M}$, a retraction is a smooth mapping $\mathcal{R}\colon T\mathcal{M} \to \mathcal{M}$ where we let $\mathcal{R}_p = \mathcal{R}(p, \cdot)\colon T_p\mathcal{M} \to \mathcal{M}$ denote the restriction onto $p \in \mathcal{M}$ with the following two properties:*

1. *$\mathcal{R}_p(0_p) = p$, with $0_p$ the origin of $T_p\mathcal{M}$.*
2. *$\mathrm{D}\mathcal{R}_p(0_p)\colon T_p\mathcal{M} \to T_p\mathcal{M} = \mathrm{Id}_{T_p\mathcal{M}}$, i.e. $\mathrm{D}\mathcal{R}_p(0_p)[\xi] = \xi,\ \forall\, \xi \in T_p\mathcal{M}$.*

We can interpret a retraction $\mathcal{R}$ as a first order approximation to the exponential map for a Riemannian manifold, even though the curves $t \mapsto \mathcal{R}_p(t\xi_p)$ are in general not geodesics.

The idea of a line search method from classical vector space optimization as presented by Nocedal and Wright [14, Chap. 3] can be adopted to work on Riemannian manifolds. Indeed, if we compose an objective function $f\colon \mathcal{M} \to \mathbb{R}$ with the retraction operator at a point $p \in \mathcal{M}$, we have a local model of $f$ over the vector space $T_p\mathcal{M} \cong \mathbb{R}^d$ where $d = \dim(\mathcal{M})$. Denoted by

$$\hat{f}_p = f \circ \mathcal{R}_p\colon T_p\mathcal{M} \cong \mathbb{R}^d \to \mathbb{R}, \tag{4.52}$$

the function $\hat{f}_p$ is called the *pullback* of $f$ through $\mathcal{R}_p$ [23, Eq. 4.3]. Due to the properties of retractions from Def. 4.13, the action of the differential of the pullback at $p$ is the same as for the objective function itself:

$$\mathrm{D}\hat{f}_p(0_p)[\xi] = \mathrm{D}(f \circ \mathcal{R}_p(0_p))[\xi] = \mathrm{D}f(\mathcal{R}_p(0_p))\,[\mathrm{D}\mathcal{R}_p(0_p)[\xi]] = \mathrm{D}f(p)[\xi],\ \xi \in T_p\mathcal{M}. \tag{4.53}$$

Thus, the differentials $\mathrm{D}\hat{f}(0_p)$ and $\mathrm{D}f(p)$ are equal under the canonical identification $T_{0_p}(T_p\mathcal{M}) \cong T_p\mathcal{M}$. Consequently, the gradient of the pullback and original objective function at $0_p$ and $p$ are also the same up to the canonical isomorphism, i.e. $\operatorname{grad}\hat{f}(0_p) = \operatorname{grad} f(p)$ [23, Eq. 4.4].

The above considerations justify the use of retractions to take steps along descent directions using the pullback function, and we expect the behavior of the pullback function to match the objective function composed with the exponential mapping to first order. In close analogy with line-search methods on Euclidean spaces we define iterates

$$\{p_i\} \subset \mathcal{M}, \quad p_{i+i} = \mathcal{R}_{p_i}(t_i \eta_i), \tag{4.54}$$

where $t_i \in \mathbb{R}^+$ is the step size taken in the search direction $\eta_i \in T_{p_i}\mathcal{M}$. We seek to iterate until we reach critical points $p^*$ of the objective function $f\colon \mathcal{M} \to \mathbb{R}$, characterized by $\mathrm{grad} f(p^*) = 0$. In the same manner as for regular Euclidean optimization this means that the differential $\mathrm{D}f(p^*)[\xi] = 0 \; \forall \; \xi \in T_{p^*}\mathcal{M}$, and we've found a stationary point of our objective function.

In order to prove the convergence of line-search methods to critical points of the objective function, we have to constrain the sequence of search directions. To this end we introduce the definition of a *gradient-related sequence*, a necessary condition on our search directions for our optimization algorithm to converge to a critical point of our objective function.

**Definition 4.14** (Gradient-related Sequence [23, def. 4.2.1]).
*Given a cost function $f\colon \mathcal{M} \to \mathbb{R}$ on a Riemannian manifold $\mathcal{M}$, a sequence of line-search directions $\{\eta_i\}, \eta_i \in T_{p_i}\mathcal{M}$ is* gradient-related *if for any subsequence $\{p_i\}_{i\in\mathcal{K}}$ of the iterates $\{p_i\}$ that converges to a non-critical point of $f$, the corresponding sub-sequence is bounded and satisfies*

$$\limsup_{i\to\infty,\, i\in\mathcal{K}} \; g_{p_i}(\mathrm{grad} f(p_i), \eta_i) < 0. \tag{4.55}$$

In this work we employ the steepest descent method where the search directions are given by $\eta_i = -\mathrm{grad} f(p_i)$, and as such our sequence of search directions are gradient-related.

Additionally, restrictions must be put on the step size $t_i$ taken along search directions $\eta_i$ to ensure convergence to a critical point. The *Armijo condition* is a standard condition on the step size $t_i$, stated for vector spaces by Nocedal and Wright [14, Alg. 3.1], and adapted to work on manifolds by Absil, Mahony, et al. [23, Def. 4.2.2].

$$f(\mathcal{R}_{p_i}(t_i\, \eta_i)) \leq f(p_i) + c_1 \; g_{p_i}(\mathrm{grad} f(p_i), \eta_i) = f(p_i) - c_1 ||\mathrm{grad} f(p_i)||^2_{p_i}, \tag{4.56}$$

The parameter $c_1 \in (0,1)$ sets the required descent proportional to the magnitude of the gradient and the step length. When computing an *Armijo step size* $t_k^A$ in practice satisfying the Armijo conditions, we choose an initial step size $t_i > 0$ and a contraction parameter $c_2 \in (0,1)$. Then we backtrack to find a $k \in \mathbb{N}$ s.t. $t_i^A = c_2^k\, t_i$ satisfies the Armijo condition in Eq. (4.56).

Adapting the Accelerated Line Search algorithm of Absil, Mahony, et al. [23] to our purposes, the general structure of our gradient based line-search method for minimizing objective functions is presented in Algorithm 1.

In this work we use the default values $c_1 = 10^{-4}$ and $c_2 = 0.2$ whenever we use Algorithm 1. Before stating a theorem regarding the convergence of the line search

**Algorithm 1** Riemannian Gradient Descent [23, p. 63]

```
Require: Riemannian manifold $\mathcal{M}$, continuously differentiable scalar valued $f\colon \mathcal{M} \to \mathbb{R}$,
    retraction $\mathcal{R}\colon T\mathcal{M} \to \mathcal{M}$, $c_1, c_2 \in (0,1)$, initial point $p_0 \in \mathcal{M}$, initial step size $t_0 > 0$.
for $i = 0, 1, \ldots$ do
    Compute $\mathrm{grad} f(p_i)$.
    Find $k \in \mathbb{N}$ s.t.
        $f(\mathcal{R}_{p_i}(\beta^k t_i\, \eta_i)) \leq f(p_i) - c_1 ||\mathrm{grad} f(p_i)||^2_{p_i},$    (4.57)
    where $t_{i+1} = c_2^k t_i$ is the backtracking Armijo step size.
    $p_{i+1} = \mathcal{R}_{p_i}(-t_{i+1}\, \mathrm{grad} f(p_i))$.
end for
```

method in Algorithm 1, we describe how the concept of convergence of a sequence can be handled on a manifold. An infinite sequence $\{p_n\}_{n=1}^{\infty}$ on the manifold $\mathcal{M}$ is said to be *convergent* if there exists a chart $(\mathcal{U}, \varphi)$ over $\mathcal{M}$, a point $p^* \in \mathcal{M}$, and an $N > 0$ s.t. for all $n > N$, $p_n \in \mathcal{U}$, and the sequence $\{\varphi(p_n)\}_{n=N+1}^{\infty}$ converges to $\varphi(p^*)$ [23, Sec. 4.3.1]. The point $p^* = \varphi^{-1}(\lim_{n\to\infty} \varphi(p_n))$ is the *limit* of such a convergent sequence $\{p_n\}_{n=1}^{\infty}$, which is unique in a Hausdorff topology [23, p. 63]. Now we can state a convergence result regarding the iterates of Algorithm 1.

**Theorem 4.3** (Accumulation to critical points [23, Thm. 4.3.1])**.** *Let $\{p_n\}$ be an infinite sequence of iterates generated by Algorithm 1. Then every accumulation point of $\{p_n\}$ is a critical point of the cost function $f\colon \mathcal{M} \to \mathbb{R}$.*

## 4.10 The Sphere and SPD Matrices

Examples of sets which admit a smooth Riemannian manifold structure include the $n$-spheres and the sets of $n \times n$ symmetric positive definite (SPD) matrices, $n \in \mathbb{N}$. We can imbue the $n$-sphere

$$\mathbb{S}(n) = \{p \in \mathbb{R}^{n+1} \mid ||p||_2 = 1\} \subset \mathbb{R}^{n+1} \tag{4.58}$$

with a smooth Riemannian manifold structure as an embedded submanifold of $\mathbb{R}^{n+1}$. To see this, consider the mapping

$$F_n\colon \mathbb{R}^{n+1} \to \mathbb{R}, \quad p \mapsto ||p||_2^2, \tag{4.59}$$

which is a submersion on $\mathbb{R}^{n+1} \setminus \{0\}$. Specifically, $1 \in \mathbb{R}$ is a regular value of $F_n$ as we can see by showing that $\mathrm{D}F_n(x)$ is a surjection for all $x \in F_n^{-1}(1) = \mathbb{S}(n)$. For any $\alpha \in \mathbb{R}$, let $\hat{x} = \frac{\alpha}{2}x$, s.t.

$$\mathrm{D}F_n(x)[\hat{x}] = \langle 2x, \hat{x}\rangle = \alpha \langle x, x\rangle = \alpha. \tag{4.60}$$

By Theorem 4.1, $\mathbb{S}(n)$ is a closed embedded submanifold of $\mathbb{R}^{n+1}$ with dimension $n$. The tangent space $T_p\mathbb{S}(n)$ can be associated with the kernel of $\mathrm{D}F_n(x)$,

$$T_p\mathbb{S}(n) = \ker(\mathrm{D}F_n(x)) = \{\xi \in \mathbb{R}^{n+1} \mid \langle \xi, p \rangle = 0\}. \tag{4.61}$$

As $T_p\mathbb{S}(n)$ can be seen as a subspace of the embedding space $\mathbb{R}^{n+1}$ the flat Riemannian metric $\bar{g}$ induces a Riemannian metric $g$ on $\mathbb{S}(n)$ by the action

$$g_p(\xi, \zeta) = \bar{g}_p(\xi, \zeta) = \langle \xi, \zeta \rangle, \ \xi, \zeta \in T_p\mathbb{S}(n). \tag{4.62}$$

The geodesics of on $\mathbb{S}(n)$ with the flat Riemannian metric inherited from the embedding space are great circles

$$\gamma(t) = \cos(||\xi||_p \, t)p + \frac{\xi_p}{||\xi||_p} \sin(||\xi||_p \, t), \quad t \in \mathbb{R} \tag{4.63}$$

with $\gamma(0) = p$, $\dot{\gamma}(0) = \xi$ [23, p. 103]. The exponential mapping at $p$ becomes

$$\exp_p \colon T_p\mathbb{S}(n) \to \mathbb{S}(n), \quad \exp_p(\xi) = \cos(||\xi||_p)p + \frac{\xi_p}{||\xi||_p} \sin(||\xi||_p). \tag{4.64}$$

The logarithmic mapping at $p$ becomes

$$\log_p \colon \mathbb{S}(n) \to T_p\mathbb{S}(n), \ \log_p(q) = \arccos(\langle p, q \rangle) \frac{q - p\langle p, q \rangle}{||q - p\langle p, q \rangle||_2}, \tag{4.65}$$

which is well-defined on $\mathbb{S}(n) \setminus \{-p\}$, as the cut locus of $p$ is the antipodal point. Finally, the Riemannian distance between two points on $\mathbb{S}(n)$ becomes

$$\mathrm{dist}_{\mathbb{S}} \colon \mathbb{S}(n) \times \mathbb{S}(n) \to \mathbb{R}^+, \quad \mathrm{dist}_{\mathbb{S}}(p, q) = \arccos(\langle p, q \rangle). \tag{4.66}$$

The set of SPD matrices in $\mathbb{R}^{n \times n}$, $n \in \mathbb{N}$

$$\mathcal{P}(n) = \{p \in \mathbb{R}^{n \times n} \mid p^T = p \text{ and } x^T p x > 0 \text{ for all } a \in \mathbb{R}^n, \ a \neq 0\} \tag{4.67}$$

is an open subset of $\mathcal{S}(n) = \{p \in \mathbb{R}^{n \times n} \mid p^T = p\}$ [29, Prop. 2.7], the vector space of $n \times n$ symmetric matrices. Thus, $\mathcal{P}(n)$ can be imbued with a smooth manifold structure as an open submanifold of $\mathcal{S}(n)$ with dimension $n(n+1)/2$. The tangent space at $p \in \mathcal{P}(n)$ can further be identified with the embedding space, i.e.

$$T_p\mathcal{P}(n) \cong \mathcal{S}(n), \quad p \in \mathcal{P}(n). \tag{4.68}$$

For the choice of Riemannian metric on $\mathcal{P}(n)$, we first present the Linear-Affine (LA) metric introduced by Pennec, Fillard, et al. [30] and defined pointwise for $p \in \mathcal{P}(n)$ as

$$g_p^{\mathrm{LA}}(\xi, \zeta) = \mathrm{tr}(p^{-1}\xi p^{-1}\zeta), \quad \xi, \zeta \in \mathcal{S}(n). \tag{4.69}$$

Highly useful for any Riemannian manifold structure on SPD matrices is the fact that the matrix exponential $\mathrm{Exp}\colon \mathcal{S}(n) \to \mathcal{P}(n)$ is a diffeomorphism from $\mathcal{S}(n)$ onto $\mathcal{P}(n)$ [29, Thrm. 2.8]. Its inverse $\mathrm{Log}\colon \mathcal{P}(n) \to \mathcal{S}(n)$ is in particular well-defined for all SPD matrices.

Writing $\xi = U\Xi U^T \in \mathcal{S}(n)$ in its spectral decomposition, with $U$ orthonormal and $\Xi = \mathrm{diag}(\{\xi_i\}_{i=1}^n)$, $\mathrm{Exp}(\xi) = U\,\mathrm{diag}(\{\exp(\xi_i)\}_{i=1}^n)U^T$. The matrix logarithm applied to an SPD matrix $p = V\varLambda V^T$ acts similarly as $\mathrm{Log}(p) = V\,\mathrm{diag}(\{\log(\lambda_i)\}_{i=1}^n)V^T$, with $V$ orthogonal and $\varLambda = \mathrm{diag}(\{\lambda_i\}_{i=1}^n), \lambda_i > 0, i = 1, \ldots, n$ [30, pp. 45-46]. Additionally, raising an SPD matrix to any power is a smooth mapping [29, Cor. 2.9]:

$$(\cdot)^\alpha\colon \mathcal{P}(n) \to \mathcal{P}(n), \quad p^\alpha = \mathrm{Exp}\left(\alpha\,\mathrm{Log}(p)\right). \tag{4.70}$$

With the LA metric, the formula for the exponential mapping reads

$$\exp_p\colon \mathcal{S}(n) \to \mathcal{P}(n), \quad \exp_p(\xi) = p^{-1/2}\,\mathrm{Exp}\left(p^{1/2}\xi p^{1/2}\right)p^{-1/2}, \tag{4.71}$$

whilst the logarithmic mapping becomes

$$\log_p\colon \mathcal{P}(n) \to \mathcal{S}(n), \quad \log_p(q) = p^{1/2}\,\mathrm{Log}\left(p^{-1/2}\xi p^{-1/2}\right)p^{1/2}, \tag{4.72}$$

and is well-defined for all $q \in \mathcal{P}(n)$ [30, p. 48]. This means that $\mathrm{inj}(\mathcal{P}(n)) = \infty$. The Riemannian distance under the Linear-Affine metric is

$$\mathrm{dist}_{\mathcal{P},\mathrm{LA}}\colon \mathcal{P}(n) \times \mathcal{P}(n) \to \mathbb{R}^+, \quad \mathrm{dist}_{\mathcal{P},\mathrm{LA}}(p,q) = ||\,\mathrm{Log}\left(p^{-1/2}qp^{-1/2}\right)||_{\mathrm{Fr}}, \tag{4.73}$$

where $||\mathbf{A}||_{\mathrm{Fr}} = \sqrt{\mathrm{tr}(\mathbf{A}^T\mathbf{A})}$ is the Frobenius norm. And with the Linear-Affine metric, $(\mathcal{P}(n), g^{\mathrm{LA}})$ is Hadamard [28, p. 10].

A second choice of Riemannian metric for the SPD manifold is the Log-Euclidean (LE) metric, as introduced by Arsigny, Fillard, et al. [29]. The authors exploit the matrix exponential diffeomorphism between $\mathcal{S}(n)$ and $\mathcal{P}(n)$ to construct a Lie group structure on the manifold of SPD matrices by the logarithmic product

$$p \odot q := \mathrm{Exp}\left(\mathrm{Log}(p) + \mathrm{Log}(q)\right), \tag{4.74}$$

which preserves the identity element and regular matrix inverse [29, Thrm. 3.3]. Choosing a Riemannian metric compatible with this Lie group structure leads to computationally expensive expressions for the exponential and logarithmic mapping, but the expression for the Riemannian distance is particularly simple and reads

$$\mathrm{dist}_{\mathcal{P},\mathrm{LE}}\colon \mathcal{P}(n) \times \mathcal{P}(n) \to \mathbb{R}^+, \quad \mathrm{dist}_{\mathcal{P},\mathrm{LE}}(p,q) = ||\,\mathrm{Log}(p) - \mathrm{Log}(q)||_{\mathrm{Fr}}. \tag{4.75}$$

Under the LE metric the space of SPD matrices is a flat Riemannian manifold, isometric to $\mathcal{S}(n)$ endowed with the Frobenius norm [29, Cor. 3.10].

# Existing Models 5

In this chapter we present the existing models for classification on Riemannian manifolds, as mentioned in Section 1.1. We assume we're working on a complete Riemannian manifold $(\mathcal{M}, \mathcal{A}^+, g)$, and denote the training data $\mathcal{X} = \{(p_i, y_i)\}_{i=1}^N \subset \mathcal{M} \times \{1, -1\}$.

## 5.1 Tangent Vector Space SVM

As mentioned in Section 1.1, one way of generalizing the classical SVM model to Riemannian manifolds is to exploit the fact that each point $q \in \mathcal{M}$ has a tangent vector space $T_q\mathcal{M}$ with an inner-product $g_q \colon T_q\mathcal{M} \times T_q\mathcal{M} \to \mathbb{R}$. If all our training points are within the injectivity radius of a certain reference point $p_{\text{ref}} \in \mathcal{M}$, so that $\log_{p_{\text{ref}}}(p_i) \in T_{p_{\text{ref}}}\mathcal{M}$ is well-defined for $i = 1, \dots, N$, we can map all the training points into $T_{p_{\text{ref}}}\mathcal{M}$ with the mapping $p_i \mapsto \log_{p_{\text{ref}}}(p_i) =: \zeta_i \in T_{p_{\text{ref}}}\mathcal{M}$. We can then use the classical SVM model from Chapter 3 directly on the tangent vector space representation of our training data

$$\log_{p_{\text{ref}}}(\mathcal{X}) = \{(\zeta_i, y_i)\}_{i=1}^N \subset T_{p_{\text{ref}}}\mathcal{M} \times \{1, -1\}. \tag{5.1}$$

In this work we will call this type of model which relies on mapping the training points to a specific tangent vector space a *Tangent Space Support Vector Machine*, or TS-SVM. In order to classify new points, they would be mapped to $T_{p_{\text{ref}}}\mathcal{M}$, and evaluated by the linear separator there. Denoting the optimal Lagrange multipliers for the trained linear separator on $T_{p_{\text{ref}}}\mathcal{M}$ by $\lambda \in \mathbb{R}^N$, the TS-SVM classifier becomes

$$\begin{gathered} f_{\text{TS-SVM}} \colon \mathcal{M} \times \mathcal{M} \times \mathbb{R}^N \times \mathbb{R}, \\ f_{\text{TS-SVM}}(q \mid p_{\text{ref}}, \lambda_i, \beta_0) = \beta_0 + \sum_{i=1}^N \lambda_i y_i \langle \log_{p_{\text{ref}}}(q), \zeta_i \rangle_{p_{\text{ref}}}. \end{gathered} \tag{5.2}$$

The TV-SVM model has been proposed and tested by Barachant, Bonnet, et al. [9] to classify human brain activity in the domain of Brain Computer Interfaces (BCIs). They work on with points on the manifold of $n \times n$ SPD matrices $\mathcal{P}(n)$ with the Linear-Affine metric, which has the advantage that $\operatorname{inj}(\mathcal{P}(n)) = \infty$, as $(\mathcal{P}(n), g^{\text{LA}})$ is Hadamard as per Definition 4.12.

With regard to the choice of reference point $p_{\text{ref}}$, we would in principle like to find the reference point with the tangent space which most closely captures the geometry locally about our training data. However, it's not obvious which objective to use in order determine such an optimal reference point.

Tuzel, Porikli, et al. [10, Eq. (26)], propose to use the discrepancy between the Riemannian distance between training points and the distance between their tangent vector space representations as a measure of how well a specific tangent space captures the local geometry. That is, they suggest that a point $p_{\text{ref}}$ which minimizes the objective

$$\epsilon(q) = \sum_{i,j=1}^{N} \Big(\operatorname{dist}(p_i, p_j) - || \log_q(p_i) - \log_q(p_i)||_q\Big)^2 \tag{5.3}$$

would be a good candidate for a reference point in which tangent space to perform the linear separation of our training data, as the distances computed in that tangent vector space would then most closely resemble the Riemannian distances between the training points.

The above objective $\epsilon(q)$ is not trivial to minimize though, and as a heuristic Tuzel, Porikli, et al. [10] suggest using the Riemannian center of mass as defined in Section 4.8. They argue that even though they don't have a theoretical proof as to why the Riemannian center of mass should be the best choice for representing the training data in a tangent space, they've generated many data sets on $\mathcal{P}(n)$ for differing $n$'s and found empirically that the discrepancy $\epsilon(q)$ at the Riemannian center of mass of the points in question was significantly lower than for any of the training points. And in the special case where all the training data lie on a geodesic, the approximation error in Eq. (5.3) vanishes at points $q$ on the geodesic, including at the Riemannian center of mass of the training data which then lies on that geodesic.

On the other hand, Barachant, Bonnet, et al. [9], compare three different natural choices of a reference point on $\mathcal{P}(n)$. The identity matrix $I \in \mathbb{R}^{n\times n}$, the arithmetic mean of their training data, and the Riemannian center of mass of their training data. The identity matrix is clearly SPD, and the arithmetic mean of SPD matrices in the embedding $\mathbb{R}^{n\times n}$ is also SPD as

$$x^T \left( \sum_{i=1}^{N} \frac{1}{N} p_i \right) x = \frac{1}{N} \sum_{i=1}^{N} x^T p_i x \geq 0, \ \{p_i\}_{i=1}^{N} \subset \mathcal{P}(n). \tag{5.4}$$

Based on the classification accuracy on their test data they conclude that the choice of the Riemannian center of mass yields the best result. They test their model by classifying four classes of human brain activity as measured through 22 electrodes. The dataset they use was part of the BCI-IV competition [39], specifically dataset 2a gathered by Naeem, Brunner, et al. [31]. We will later present and use the same dataset as an example for comparing the performance of different manifold SVM models in Section 7.3.

## 5.2 Manifold Radial Basis Function SVM

Jayasumana, Hartley, et al. [11], propose a way of generalizing the classical SVM model to Riemannian manifolds by utilizing the kernel trick presented in Section 3.2. Specifically, they construct a valid kernel function on the Riemannian manifold, and take advantage of the fact that the classical SVM model only requires a valid kernel function to implicitly generate linear separators in a Hilbert space $\mathcal{H}$.

The authors of [11] construct the manifold kernel based on the Euclidean RBF kernel introduced in Eq. (3.28), in combination with Theorem 3.2. They show that if the Riemannian distance function can be expressed as

$$\operatorname{dist}_g(p,q) = ||\Phi(p) - \Phi(q)||_{\mathcal{H}}, \quad \Phi\colon \mathcal{M} \to \mathcal{H}, \tag{5.5}$$

where $\mathcal{H}$ is a Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, then

$$K_{\text{MRBF}}\colon \mathcal{M}\times\mathcal{M} \to \mathbb{R}^+, \quad K_{\text{MRBF}}(p,q \mid \sigma^2) = \exp\left(-\frac{\operatorname{dist}_g^2(p,q)}{2\sigma^2}\right) \tag{5.6}$$

is positive definite for all $\sigma^2 > 0$ and thus a valid kernel function [11, Thrm. 6.1]. To prove the above statement they define the kernel

$$\phi_g\colon \mathcal{M}\times\mathcal{M} \to \mathbb{R}, \quad \phi_g(p,q) = \operatorname{dist}_g^2(p,q) = ||\Phi(p) - \Phi(q)||_{\mathcal{H}}^2, \tag{5.7}$$

and show that it is conditionally negative definite in the sense of Definition 3.1. To see this, consider any set of points $\{q_i\}_{i=1}^M$, $q_i \in \mathcal{M}$, and any set $\{c_i\}_{i=1}^M$, $c_i \in \mathbb{R}$ s.t. $\sum_{i=1}^M c_i = 0$, for an arbitrary $M \in \mathbb{N}$. Then,

$$\begin{aligned}
\sum_{i,j=1}^M c_i c_j \phi_g(q_i,q_j) &= \sum_{i,j=1}^M c_i c_j ||\Phi(q_i) - \Phi(q_j)||_{\mathcal{H}}^2 \\
&= \sum_{i,j=1}^M c_i c_j \langle \Phi(q_i) - \Phi(q_j), \Phi(q_i) - \Phi(q_j)\rangle_{\mathcal{H}} \\
&= \sum_{j=1}^M c_j \sum_{i=1}^M c_i \langle \Phi(q_i), \Phi(q_i)\rangle_{\mathcal{H}} \\
&\quad - 2\sum_{i,j=1}^M c_i c_j \langle \Phi(q_i), \Phi(q_j)\rangle_{\mathcal{H}} \\
&\quad + \sum_{i=1}^M c_i \sum_{j=1}^M c_j \langle \Phi(q_j), \Phi(q_j)\rangle_{\mathcal{H}}, \\
&= -2\sum_{i,j=1}^M c_i c_j \langle \Phi(q_i), \Phi(q_j)\rangle_{\mathcal{H}} = -2\left|\left|\sum_{i=1}^M c_i \Phi(q_i)\right|\right|_{\mathcal{H}}^2 \leq 0.
\end{aligned} \tag{5.8}$$

We will call this type of model which extends the SVM kernel trick to Riemannian manifolds through the RBF kernel *Manifold RBF SVM*, or MRBF for short. A desirable property of MRBF models as compared to the TS-SVM model is the fact that we do not need to choose a reference point in whose tangent space we construct the classical SVM model. However, the only Riemannian manifolds for which MRBF generates a positive definite kernel are exactly the ones for which the distance function satisfies Eq. (5.5) for some mapping $\Phi\colon \mathcal{M} \to \mathcal{H}$, meaning that $\mathcal{M}$ can be isometrically imbedded in a Hilbert space [11, Thrm. 6.1], [21, p. 81].

The Riemannian manifold of SPD matrices with the Log-Euclidean metric, $(\mathcal{P}(n), g^{\mathrm{LE}})$, is an example of a manifold which can be isometrically imbedded in $\mathcal{S}(n)$, the set of symmetric $n \times n$ matrices. Letting $\Phi_{g^{\mathrm{LE}}}(p) = \mathrm{Log}(p) \in \mathcal{S}(n)$, the Riemannian distance function on $(\mathcal{P}(n), g^{\mathrm{LE}})$ from Eq. (4.75) can be expressed as

$$\mathrm{dist}_{\mathrm{P},\, g^{\mathrm{LE}}}(p, q) = ||\Phi_{g^{\mathrm{LE}}}(p) - \Phi_{g^{\mathrm{LE}}}(q)||_{\mathcal{S}(n)}. \tag{5.9}$$

Several manifolds can not be isometrically imbedded in a Hilbert space, like the n-sphere $\mathbb{S}(n)$ [11, p. 2468]. However, for practical purposes one can still generate a ”pseudo-kernel” matrix $\widetilde{\mathbf{K}}_{\mathrm{MRBF}}$ using the kernel mapping in Eq. (5.6). And as long as $\widetilde{\mathbf{K}}_{\mathrm{MRBF}}$ is positive semi-definite one can still solve the Wolfe dual in Prob. (3.18), although the notion that we're constructing an implicit linear separator in an infinite dimensional Hilbert space is no longer valid in this case.

In practice when using the MRBF model, we need to choose an appropriate ”variance” or length scale for the kernel mapping $K_{\mathrm{MRBF}}(p, q \mid \sigma^2)$, as applied to the training data $\mathcal{X}$. In this work we choose $\sigma$ as a fraction of the average *between-class distance* (BCD). If there are $N_1$ positive class points and $N_2$ negative class points in the training data with $N = N_1 + N_2$ points, this quantity is calculated as

$$\overline{\mathrm{BCD}}(\mathcal{X}) = \frac{1}{N_1 N_2} \sum_{\substack{(y_i, p_i) \in \mathcal{X}, \\ y_i = 1}} \; \sum_{\substack{(y_j, p_j) \in \mathcal{X}, \\ y_j = -1}} \mathrm{dist}(p_i, p_j). \tag{5.10}$$

Similarly, we quantify the average *positive class distance* (PCD) as

$$\overline{\mathrm{PCD}} = \frac{1}{N_1(N_1 - 1)/2} \sum_{i=1}^{N_1} \sum_{j=i+1}^{N_1} \mathrm{dist}(p_i, p_j), \quad y_i, y_j = 1, \tag{5.11}$$

and average *negative class distance* as

$$\overline{\mathrm{NCD}} = \frac{1}{N_2(N_2 - 1)/2} \sum_{i=1}^{N_2} \sum_{j=i+1}^{N_2} \mathrm{dist}(p_i, p_j), \quad y_i, y_j = -1, \tag{5.12}$$

To set $\sigma^2$ then, we choose a fractional scale $\sigma_s > 0$, and set

$$\sigma = \sigma_s \cdot \overline{\mathrm{BCD}}(\mathcal{X}). \tag{5.13}$$

## 5.3 Manifold Control Point SVM

In contrast to the previous manifold SVM models which work to adapt the classical SVM model to work on Riemannian manifolds, the model introduced by Sen, Foskey, et al. [13] takes a more manifold centric approach. They introduce two control points $c_+$ and $c_-$ on $\mathcal{M}$ as representatives for the positive and negative class, respectively, and classify points $q \in \mathcal{M}$ according to which control point they are closest to.

Specifically, Sen, Foskey, et al. [13] introduce the control point (CP-SVM) classifier

$$f_{\mathrm{CP}}(q \mid c_+, c_-) = \operatorname{dist}^2(q, c_-) - \operatorname{dist}^2(q, c_+). \tag{5.14}$$

The alternating choice of signs means that $f_{\mathrm{CP}}(q \mid c_+, c_-) > 0$ when $q$ is farther away from $c_-$ than $c_+$, and vice versa. The separating surface on $\mathcal{M}$ for a choice of control points consists of all the points $q \in \mathcal{M}$ for which $f_{\mathrm{CP}}(q \mid c_+, c_-) = 0$, and is denoted

$$H(c_+, c_-) = \{q \in \mathcal{M} \mid \operatorname{dist}^2(q, c_-) = \operatorname{dist}^2(q, c_+)\}. \tag{5.15}$$

If we imagine that we're working on a Euclidean space $\mathcal{E}$, the distance between any point $x \in \mathcal{E}$ and $H(c_+, c_-)$ can be expressed as

$$\operatorname{dist}(x, H(c_+, c_-)) = \frac{|f_{\mathrm{CP}}(x \mid c_+, c_-)|}{2 \operatorname{dist}(c_+, c_-)}, \tag{5.16}$$

which Sen, Foskey, et al. [13, Sec. 2.3.1] say motivates the idea of minimizing $\operatorname{dist}^2(c_+, c_-)$ to maximize the "margin" between training points and the separating surface $H(c_+, c_-)$. To solve for the best set of control points they propose the unconstrained optimization problem

$$\min_{c_+, c_- \in \mathcal{M} \times \mathcal{M}} \quad F_{\mathrm{CP}} := \operatorname{dist}^2(c_+, c_-) + \frac{C}{N} \sum_{i=1}^{N} h_+(k - y_i f_{\mathrm{CP}}(p_i \mid c_+, c_-)), \tag{5.17}$$

where $h_+(x) = \max\{x, 0\}$ is the hinge loss function. The parameter $k > 0$ functions as an insensitivity threshold, of $k > 0$, returning zero if a point $p_i$ is correctly classified by an amount $k$ or greater. The parameter $C > 0$ functions as a penalty weight, balancing the desire to classify points correctly by an amount $k$, with minimizing the squared distance between the control points.

The hinge loss function is however not differentiable at zero, which makes the objective in Prob. (5.17) not differentiable. Sen, Foskey, et al. [13] do not discuss which optimization procedure they applied to minimize the objective in Prob. (5.17), but in order to be able to apply the gradient descent method in Algorithm 1 we replace $h(x)_+$ with the Huber hinge loss $\mathrm{Hu}_+$ as presented in [32, p. 4]:

$$\mathrm{Hu}_+ \colon \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}, \quad \mathrm{Hu}_+(z \mid \varepsilon) = \begin{cases} 0 & \text{if } z < 0, \\ \frac{z^2}{2\varepsilon} & \text{if } 0 \leq z < \varepsilon, \\ z - \frac{\varepsilon}{2} & \text{if } \varepsilon \leq z. \end{cases} \tag{5.18}$$

The Huber hinge loss is continuously differentiable as it transitions from quadratic over $[0, \epsilon)$ to linear over $[\varepsilon, \infty)$, with a vanishing derivative at $z = 0$. In this work we've chosen the default value $\varepsilon = 10^{-4}$ to retain a close resemblance to the hinge loss.

Similar to how we choose the $\sigma^2$ hyperparameter for the MRBF model, we choose the scale dependent margin $k > 0$ for the CP-SVM model as a fraction of the average between class distance. Specifically, we choose $k_s \in (0, 1)$ and set

$$k = k_s \cdot \overline{\mathrm{BCD}}(\mathcal{X}). \tag{5.19}$$

With the Huberized hinge loss the objective in Prob. (5.17) is differentiable, and we apply Algorithm 1 to minimize it. Without any global convergence results to an optimum we can only expect to find a local minimum of the CP-SVM objective. And as a stopping criterion we use whether $||\mathrm{grad} F_{\mathrm{CP}}||_{p^i} < 10^{-6}$, or $|F_{\mathrm{CP}}(p^{i+1}) - F_{\mathrm{CP}}(p^i)| < 10^{-6}$ from one iteration step to the next.

# 6 Distance SVM

In this thesis we present a new model for classifying points on a Riemannian manifold $\mathcal{M}$ called *Distance SVM*. The classifier works by weighing the squared distances to a set of support points $\mathcal{X}_\mathrm{S} = \{y_i, p_i\}_{i=1}^{M} \subset \{-1, 1\} \times \mathcal{M}$,

$$f_\mathrm{D} \colon \mathcal{M} \times \mathbb{R}^M \times \mathbb{R} \to \mathbb{R}, \quad f_\mathrm{D}(q \mid \beta, \beta_0) = \beta_0 - \sum_{i=i}^{M} \beta_i y_i \operatorname{dist}^2(q, p_i). \tag{6.1}$$

The set of support points can in general be a subset of the full set of training points $\mathcal{X}_T = \{y_i, p_i\}_{i=1}^{N}$, i.e. $\mathcal{X}_S \subseteq \mathcal{X}_T$, but for the introduction of the model we'll assume that the entire training set is used as support points unless otherwise stated. For notational purposes it is useful to define the *negated squared distance* mapping, which takes as input a point $q \in \mathcal{M}$ and computes the negated squared distances to all the support points,

$$\begin{gathered} \Upsilon \colon \mathcal{M} \times \mathcal{M}^N \to \mathbb{R}^N, \\ \Upsilon(q \mid \mathcal{X}_S) = [-\operatorname{dist}^2(q, p_1),\ \ldots,\ -\operatorname{dist}^2(q, p_N)]^\mathrm{T}. \end{gathered} \tag{6.2}$$

The dependence of $\Upsilon$ on the support points is suppressed for brevity when it is obvious which set of support points we are considering. The negated squared distance mapping allows us to rewrite the Distance SVM classifier in Eq. (6.1) as

$$f_\mathrm{D}(q \mid \beta, \beta_0) = \beta_0 + \Upsilon(q)^T \mathbf{Y} \beta = \beta_0 + (\mathbf{Y}\Upsilon(q))^T \beta, \tag{6.3}$$

where $\mathbf{Y} = \operatorname{diag}(\{y_i\}_{i=1}^{N})$. This expression for the classifier suggests the interpretation that the Distance SVM model constructs a linear separator in the feature space $\mathbb{R}^N$ after mapping $q \mapsto \mathbf{Y}\Upsilon(q) \in \mathbb{R}^N$. This reformulation also allows us to represent the gradient of the separator function w.r.t. $\beta$ as

$$\nabla_\beta f_\mathrm{D}(q \mid \beta, \beta_0) = \mathbf{Y}\Upsilon(q). \tag{6.4}$$

The Distance SVM classifier is inspired by the control point SVM model of Sen, Foskey, et al. [13]. However, whereas their classifier introduces two new control points and assigns a class label according to which control point is closer, the Distance SVM classifier weighs signed distances to existing training points. Additionally, the Distance SVM classifier includes a bias $\beta_0 \in \mathbb{R}$ to allow for greater flexibility in handling skewed support point sets, if for example the number of points of each class is not balanced.

In the same manner as for the Euclidean SVM model, our training data is correctly classified with a *margin* of $\mathrm{M} > 0$ if

$$y f_{\mathrm{D}}(p \mid \beta, \beta_0) \geq \mathrm{M}, \ \forall \ (y, p) \in \mathcal{X}_T. \tag{6.5}$$

Even though the quantity $\mathrm{M}$ is analogous to the margin in the Euclidean SVM case, equal to the minimum distance from any training point to the separating hyperplane defined in Eq. (3.1), this geometric interpretation does not hold for the Distance SVM classifier. Denoting the separating hypersurface on our manifold $\mathcal{M}$ by

$$H_{\mathrm{D}}(\beta, \beta_0) = \{q \in \mathcal{M} \mid f_{\mathrm{D}}(q \mid \beta, \beta_0) = 0\}, \tag{6.6}$$

the absolute value of the classification function $|f_{\mathrm{D}}(q \mid \beta, \beta_0)|$ is not proportional to the distance between $q$ and $\mathrm{H}_{\mathrm{D}}(\beta, \beta_0)$, i.e.

$$\left| \frac{1}{||\beta||_2} f_{\mathrm{D}}(q \mid \beta, \beta_0) \right| \neq \operatorname{dist}(q, H_{\mathrm{D}}(\beta, \beta_0)) = \inf_{p \in H(\beta, \beta_0)} \operatorname{dist}(q, p). \tag{6.7}$$

This is due to the lack of a Hilbert space structure on the Riemannian manifold $\mathcal{M}$, which means we cannot construct a linear separator on the manifold directly, instead constructing a linear separator for $\mathbf{Y}\Upsilon(\mathcal{M}) \subset \mathbb{R}^N$.

To continue the analogy with Euclidean SVM's we would ideally like to end up with a sparse $\beta$ vector in the final classification function, so that when we wish to classify a new point $q \in \mathcal{M}$, we only need to compute the distance between the $q$ and a subset of our training points. The nonzero $\beta_i$ would then correspond to points $p_i$ which we call *support points* of our model on the manifold.

## 6.1 The Optimization Problem

Analogous to the optimization problem for the Euclidean SVM model we seek to maximize the quantity $\mathrm{M}$ from Eq. (6.5) to find the maximum margin Distance SVM classifier. This leads directly to the optimization problem

$$\begin{aligned} \max_{\beta \in \mathbb{R}^N, \beta_0 \in \mathbb{R}} & \quad \mathrm{M}, \\ \text{s.t.} & \\ & y f_{\mathrm{D}}(p|\beta, \beta_0) \geq \mathrm{M}, \ \forall \ (y, p) \in \mathcal{X}_T. \end{aligned} \tag{6.8}$$

We can transform Prob. (6.8) to an equivalent problem which is more amenable to being solved in its dual form by noting that

$$|f_{\mathrm{D}}(q \mid \beta, \beta_0)| \propto ||\beta||_2. \tag{6.9}$$

Then similar to what we did when constructing the Euclidean SVM optimization problem in Chap. 3 we rescale the classification inequalities of Eq. (6.5) by setting $\mathrm{M} = 1/||\beta||_2$ and requiring

$$\frac{1}{||\beta||_2} y f_{\mathrm{D}}(p|\beta, \beta_0) \geq \mathrm{M} \ \Rightarrow \ y f_{\mathrm{D}}(p|\beta, \beta_0) \geq ||\beta||_2 \mathrm{M} = 1, \ \forall \, (y, p) \in \mathcal{X}_T. \tag{6.10}$$

To find the maximum margin Distance SVM classifier then, we instead minimize $\frac{1}{2}||\beta||_2^2$ leading to the equivalent optimization problem

$$\begin{aligned} \min_{\beta \in \mathbb{R}^M, \beta_0 \in \mathbb{R}} \quad & \frac{1}{2}||\beta||^2, \\ \text{s.t.} \quad & \\ y f_{\mathrm{D}}(p \mid \beta, \beta_0) = y(\beta_0 + \beta^T \kappa(p)) \geq 1, & \\ & \forall \, (y, p) \in \mathcal{X}_T. \end{aligned} \tag{6.11}$$

We say that the training data set $\mathcal{X}_T$ is *separable* if there exists $\beta, \beta_0$ such that the condition in Eq. (6.5) is fulfilled for an $\mathrm{M} > 0$, or equivalently if the feasible set of Prob. (6.11) is non-empty. Not all training sets are separable however, and we would like to be able to be able to construct a classifier trained on non-separable data as well. Additionally, by requiring that we classify a training data set exactly, we run the risk of overfitting our classifier to the training data. To alleviate both of the above-mentioned problems we introduce non-negative slack variables $\xi \in \mathbb{R}^N, \xi \geq 0$ to the rescaled classification inequalities in Eq. (6.10), requiring instead

$$y_i f_{\mathrm{D}}(p_i|\beta, \beta_0) \geq 1 - \xi_i, \ i = 1, \ldots, N. \tag{6.12}$$

This way of achieving leniency in the classification of each training point is entirely similar as to what was done for the Euclidean SVM model in Eq. (3.9). The introduction of the slack variable $\xi_i$ allows for under-classification of the training point $p_i$ by an amount proportional to $1/||\beta||$, leading to misclassification if $\xi > 1$. In order to control the amount of misclassification we add the mean of the slack variables to $\xi$ to Prob. (6.11) to arrive at the following optimization problem for the Distance SVM classifier:

$$\min_{\beta \in \mathbb{R}^N, \beta_0 \in \mathbb{R}, \xi \in \mathbb{R}^N} \quad \frac{1}{2}||\beta||^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i, \tag{6.13a}$$

$$\text{s.t.} \quad y_i \, f_{\mathrm{D}}(p_i \mid \beta, \beta_0) \geq 1 - \xi_i, \ i = 1, \ldots, N, \tag{6.13b}$$

$$\xi \geq 0. \tag{6.13c}$$

The constant $C > 0$ is a hyperparameter penalizing the average misclassification, and balancing the desire to maximize the separating margin with the average misclassification of our training data, which is bounded above by $\frac{1}{N}\sum_{i=1}^{N} \xi_i$. Letting $C \to \infty$ we recover the separable formulation, but this problem might be infeasible depending on the training data as mentioned.

## 6.2 Distance SVM on Hilbert Spaces

Although the Distance SVM model is intended for use on manifolds where we take advantage of the Riemannian distance, it is beneficial to investigate properties of the model when used on a Hilbert space $V$ with an inner product $\langle \cdot, \cdot \rangle \colon V \times V \to \mathbb{R}$. By investigating how the Distance SVM model looks in Hilbert spaces we can glean intuition about how the Distance SVM model compares to the Euclidean SVM model which we seek to generalize. Crucially we can express the squared distance function between two points $p, q \in V$ as $\operatorname{dist}^2(p,q) = \langle p - q, p - q \rangle$, which allows us to separate the Distance SVM classifier into several terms.

Considering training data $\mathcal{X}_{\mathrm{T}} = \{(y_i, p_i)\}_{i=1}^N \subset \{-1, 1\} \times V$ the Distance SVM classifier can be expressed as

$$\begin{aligned} f_{\mathrm{D}}(q \mid \beta, \beta_0) &= \beta_0 - \sum_{i=1}^N \beta_i y_i \operatorname{dist}^2(q, p_i) = \beta_0 - \sum_{i=1}^N \beta_i y_i \langle q - p_i, q - p_i \rangle, \\ &= \beta_0 - \sum_{i=1}^N \beta_i y_i (||q||^2 - 2\langle q, p_i \rangle + ||p_i||^2), \\ &= \beta_0 - \sum_{i=1}^N \beta_i y_i ||p_i||^2 + \sum_{i=1}^N 2\beta_i y_i \langle q, p_i \rangle - ||q||^2 \sum_{i=1}^N \beta_i y_i, \\ &= \left( \beta_0 - \sum_{i=1}^N \beta_i y_i ||p_i||^2 \right) + \left\langle q, \sum_{i=1}^N 2\beta_i y_i p_i \right\rangle - y^T \beta ||q||^2, \\ &= \tilde{\beta}_0 + \langle q, \boldsymbol{n}_{\mathrm{D}} \rangle - y^T \beta ||q||^2. \end{aligned} \tag{6.14}$$

In the last equality above we've introduced the parameters

$$\tilde{\beta}_0 = \beta_0 - \sum_{i=1}^N \beta_i y_i ||p_i||^2, \quad \boldsymbol{n}_{\mathrm{D}} = \sum_{i=1}^N 2\beta_i y_i p_i, \tag{6.15}$$

which are the effective bias and $\nabla_q f_{\mathrm{D}}(q \mid \beta, \beta_0)|_{q=0}$, respectively. From the last line of Eq. (6.14) we see that the Distance SVM classifier is similar in form to the Euclidean SVM classifier, except for the quadratic dependence on $q$ through the term $y^T \beta ||q||^2$.

Consider the Euclidean SVM classifier with normal vector $\beta_{\mathrm{SVM}} = \sum_{i=1}^N \alpha_i y_i p_i$ in the expression required by the KKT conditions in Eq. (3.14), allowing us to write

$$f_{\mathrm{SVM}}(q \mid \beta_{\mathrm{SVM}}, \beta_0) = \beta_0 + \sum_{i=1}^N \alpha_i y_i \langle q, p_i \rangle = \beta_0 + \left\langle q, \sum_{i=1}^N \alpha_i y_i p_i \right\rangle, \tag{6.16}$$

with bias $\beta_0 \in \mathbb{R}$ and Lagrange multipliers $\alpha_i \geq 0,\ i = 1, \ldots, N$. Then we can clearly see the similarities with the expression for the Distance SVM classifier in Eq. (6.14). Comparing the Distance SVM and Euclidean SVM separators further, we have from the KKT condition corresponding to the Euclidean SVM classifier bias $\beta_0$ that

$$\sum_{i=1}^{N} \alpha_i y_i = 0. \tag{6.17}$$

If we imagine imposing the same constraint on the analogous Distance SVM support point weights $\beta$ we find that $f_{\mathrm{D}}$ reduces to a linear separator on $V$ as

$$y^T \beta ||q||^2 = ||q||^2 \sum_{i=1}^{N} \beta_i y_i = 0, \tag{6.18}$$

eliminating the dependence on $||q||^2$ in $f_{\mathrm{D}}(q \mid \beta, \beta_0)$. The constraint in Eq. (6.18) can be intuitively understood as requiring that we weigh the contributions of the two classes equally in the Distance SVM separator. Using the Euclidean SVM separator as inspiration then, we can constrain the Distance SVM model to linear separators in Hilbert spaces by requiring $y^T \beta = 0$ as we solve Prob. (6.13). We call this extra constraint the *zero curvature* (ZC) constraint, as it ensures that the separating surface $H_{\mathrm{D}}(\beta, \beta_0)$ becomes a zero curvature hyperplane on Hilbert spaces.

Imposing this constraint on the weights $\beta$ leads to the *Zero Curvature Distance SVM* (ZCDSVM) model, and to find it we solve the optimization problem

$$\begin{aligned}
\min_{\beta \in \mathbb{R}^N, \beta_0 \in \mathbb{R}, \xi \in \mathbb{R}^N} \quad & \frac{1}{2}||\beta||^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i, && \text{(6.19a)}\\
\text{s.t.} \quad & y_i\, f_{\mathrm{D}}(p_i \mid \beta, \beta_0) \geq 1 - \xi_i,\ i = 1, \ldots, N, && \text{(6.19b)}\\
& \xi \geq 0, && \text{(6.19c)}\\
& \beta^T y = 0. \quad (\textit{ZC const.}) && \text{(6.19d)}
\end{aligned}$$

Having constrained the Distance SVM model to linear separators in $V$ through the zero curvature constraint one might expect the Distance SVM separator to recreate the Euclidean SVM maximum margin linear separator, at least in the separable case. However, that is not generally the case, and the reason as we'll see below comes down to the difference in objective functions for the Euclidean SVM and Distance SVM model optimization problems.

| | Distance SVM | Zero Curvature DSVM | Euclidean SVM |
|---|---|---|---|
| Classifier: | $\tilde{\beta}_0 + \langle q, \boldsymbol{n}_{\mathrm{D}} \rangle - y^T\beta \|\|q\|\|^2$ | $\tilde{\beta}_0 + \langle q, \boldsymbol{n}_{\mathrm{D}} \rangle$ | $\beta_0 + \langle q, \beta_{\mathrm{SVM}} \rangle$ |
| Objective: | $\frac{1}{2}\|\|\beta\|\|_2^2$ | $\frac{1}{2}\|\|\beta\|\|_2^2$ | $\frac{1}{2}\|\|\beta_{\mathrm{SVM}}\|\|_2^2$ |
| Constraints: | $y_i\, f_{\mathrm{D}}(p_i \mid \beta, \beta_0) \geq 1$ | $y_i\, f_{\mathrm{D}}(p_i \mid \beta, \beta_0) \geq 1,$<br>$\beta^T y = 0$ | $y_i\, f_{\mathrm{SVM}}(p_i \mid \beta_{\mathrm{SVM}}, \beta_0) \geq 1$ |

Tab. 6.1.: Comparison of DSVM, ZC-DSVM and Euclidean SVM classifiers and optimization problems for separable training data $\mathcal{X}_T = (y_i, p_i)_{i=1}^N$ on a Hilbert space $V$. Parameters $\tilde{\beta}_0$ and $\boldsymbol{n}_{\mathrm{D}}$ of DSVM models as given in Eq. (6.15).

For simplicity, assume that the training data $\mathcal{X}$ is separable, so that we can disregard the slack variables $\xi$ for both models when comparing them. Stated one more time, the Zero Curvature Distance SVM and Euclidean SVM classifiers take the forms

$$f_{\mathrm{D}}(q \mid \beta, \beta_0) = \tilde{\beta}_0 + \langle q, \boldsymbol{n}_{\mathrm{D}} \rangle \quad = \tilde{\beta}_0 + \sum_{i=1}^N 2\beta_i y_i \langle q, p_i \rangle, \tag{6.20}$$

and

$$f_{\mathrm{SVM}}(q \mid \beta_{\mathrm{SVM}}, \beta_0) = \beta_0 + \langle q, \beta_{\mathrm{SVM}} \rangle = \beta_0 + \sum_{i=1}^N \alpha_i y_i \langle q, p_i \rangle, \tag{6.21}$$

respectively. The normal vector to the Euclidean SVM classifier is $\beta_{\mathrm{SVM}}$, expressed as the linear combination of support vectors $\beta_{\mathrm{SVM}} = \sum_{i=1}^N \alpha_i y_i p_i$, whilst the normal vector to the zero curvature DSVM hyperplane is $\boldsymbol{n}_{\mathrm{D}} = \sum_{i=1}^N 2\beta_i y_i p_i$.

Comparing the Zero Curvature Distance SVM optimization problem in Table 6.1 we see that it is very similar to the Euclidean SVM optimization, but with the crucial difference that we're not directly minimizing the norm of the normal vector to the resulting ZCDSVM separating hyperplane, as is the case with the Euclidean SVM separator. Instead, we're minimizing the norm of the weight vector $||\beta||_2$, although the two are related in the sense that

$$\begin{aligned} ||\boldsymbol{n}_{\mathrm{D}}||_2 = \left|\left| \sum_{i=1}^N 2\beta_i y_i p_i \right|\right| &\leq 2 \sum_{i=1}^N |\beta_i| \cdot ||y_i p_i||_2, \\ &\leq 2 \left( \sum_{i=1}^N ||p_i||_2^2 \right)^{1/2} \left( \sum_{i=1}^N |\beta_i|^2 \right)^{1/2}, \\ &= 2 \left( \sum_{i=1}^N ||p_i||_2^2 \right)^{1/2} ||\beta||_2. \end{aligned} \tag{6.22}$$

The reason we can interpret $f_{\mathrm{SVM}}(q \mid \beta_{\mathrm{SVM}}, \beta_0)$ as the maximum margin linear separator is because $\left| \frac{1}{||\beta_{\mathrm{SVM}}||_2} f_{\mathrm{SVM}}(q \mid \beta_{\mathrm{SVM}}, \beta_0) \right|$ equals the Euclidean distance from $q$ to the separating hyperplane

$$H(\beta_{\mathrm{SVM}}, \beta_0) = \{q \in V \mid \beta_0 + \langle q, \beta_{\mathrm{SVM}} \rangle = 0\}. \tag{6.23}$$

By minimizing $||\beta_{\mathrm{SVM}}||_2$, while ensuring that all the training points are correctly classified, we maximize the margin $\mathrm{M} = 1/||\beta_{\mathrm{SVM}}||$. The Distance SVM model on the other hand does not minimize the norm of the resulting normal vector $\boldsymbol{n}_{\mathrm{D}}$, as the Distance SVM model is designed to work on Riemannian manifolds without a normed vector space structure. Therefore, the Zero Curvature Distance SVM linear separator in general is not the same maximum margin linear separator as for the Euclidean SVM model.

## 6.3 Optimizing the Distance SVM Models

When solving for the Distance SVM classifiers, with or without the zero curvature condition, we do not make the assumption that the classifier uses the entire training set as support points. Instead, we denote the training set and support set on the Riemannian manifold $\mathcal{M}$ as

$$\mathcal{X}_{\mathrm{T}} = \left\{(y_j^{\mathrm{t}},\, p_j^{\mathrm{t}})\right\}_{i=1}^{N} \subset \{-1, 1\} \times \mathcal{M}, \quad \mathcal{X}_{\mathrm{S}} = \left\{(y_j^{\mathrm{s}},\, p_j^{\mathrm{s}})\right\}_{j=1}^{M} \subseteq \mathcal{X}_{\mathrm{T}}. \tag{6.24}$$

We also define the class designation vectors

$$y_{\mathrm{T}} = [y_1^{\mathrm{t}}, \ldots, y_N^{\mathrm{t}}] \in \mathbb{R}^N \quad \text{and} \quad y_{\mathrm{S}} = [y_1^{\mathrm{s}}, \ldots, y_M^{\mathrm{s}}] \in \mathbb{R}^M, \tag{6.25}$$

which encapsulate the class labels of the training points and support points, respectively. Their matrix representations are denoted $\mathbf{Y}_{\mathrm{T}} = \mathrm{diag}(y_{\mathrm{T}})$ and $\mathbf{Y}_{\mathrm{S}} = \mathrm{diag}(y_{\mathrm{S}})$. With this notation the Distance SVM classifier with support points $\mathcal{X}_{\mathrm{S}}$ can be expressed as

$$f_{\mathrm{D}} \colon \mathcal{M} \times \mathbb{R}^M \times \mathbb{R} \to \mathbb{R} \quad f_{\mathrm{D}}(q \mid \beta, \beta_0) = \beta_0 + (\mathbf{Y}_{\mathrm{S}} \Upsilon_{\mathrm{S}}(q))^T \beta, \tag{6.26}$$

where the signed squared distance mapping is restricted to the set of support points:

$$\Upsilon_{\mathrm{S}} \colon \mathcal{M} \to \mathbb{R}^M, \quad q \mapsto \Upsilon(q \mid \mathcal{X}_{\mathrm{S}}) = [-\operatorname{dist}^2(q, p_1^{\mathrm{s}}), \ldots, \operatorname{dist}^2(q, p_M^{\mathrm{s}})]^T. \tag{6.27}$$

Let $\lambda \in \mathbb{R}^N$ be the Lagrange multipliers for the classification constraints in Eq. (6.13b) and $\mu \in \mathbb{R}^N$ be the Lagrange multipliers for the non-negativity constraint on $\xi$ in Eq. (6.13c). Then the Lagrangian for the DSVM optimization problem in Prob. (6.13) is

$$\begin{aligned}
\mathcal{L}_{\mathrm{D}} &:= \frac{1}{2}||\beta||^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i(y_i f_{\mathrm{D}}(p_i|\beta,\beta_0) - 1 + \xi_i) - \sum_{i=1}^{N}\mu_i\xi_i \\
&= \frac{1}{2}||\beta||^2 + \sum_{i=1}^{N}\left(\frac{C}{N} - \mu_i\right)\xi_i - \sum_{i=1}^{N}\lambda_i\left(y_i(\beta_0 + (\mathbf{Y}_{\mathrm{S}}\Upsilon(p_i))^T\beta) - 1 + \xi_i\right), \\
&= \frac{1}{2}||\beta||^2 + \sum_{i=1}^{N}\left(\frac{C}{N} - \lambda_i - \mu_i\right)\xi_i - \sum_{i=1}^{N}\lambda_i y_i\left(\mathbf{Y}_{\mathrm{S}}\Upsilon(p_i)\right)^T\beta - \beta_0\lambda^T y_{\mathrm{T}} + \lambda^T\mathbb{1}, \\
&= \frac{1}{2}||\beta||^2 + \sum_{i=1}^{N}\left(\frac{C}{N} - \lambda_i - \mu_i\right)\xi_i - (\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda)^T\beta - \beta_0\lambda^T y_{\mathrm{T}} + \lambda^T\mathbb{1}.
\end{aligned} \tag{6.28}$$

Here we've introduced the *squared distance kernel matrix* $\mathbf{K}_{\mathrm{D}}$ defined column wise as

$$\mathbf{K}_{\mathrm{D}} = \left[\Upsilon_{\mathrm{S}}(p_1) \mid \ldots \mid \Upsilon_{\mathrm{S}}(p_N)\right] \in \mathbb{R}^{M\times N}. \tag{6.29}$$

In the special case where we use all our training points as support points, $\mathbf{K}_{\mathrm{D}} \in \mathbb{R}^{N\times N}$ becomes symmetric, and can be expressed element wise as

$$(\mathbf{K}_{\mathrm{D}})_{i,j} = -\operatorname{dist}^2(p_i, p_j),\ i,j = 1,\ldots,N. \tag{6.30}$$

Now we're prepared to state the theorem stating how we can solve for the Distance SVM model on complete Riemannian manifolds.

**Theorem 6.1** (Solving the DSVM Optimization Problem)**.** *Let $(\mathcal{M}, \mathcal{A}^+, g)$ be a complete, path-connected Riemannian manifold. Denote a set of $N$ training points $\mathcal{X}_{\mathrm{T}}$ and $M \leq N$ support points $\mathcal{X}_{\mathrm{S}} \subseteq \mathcal{X}_{\mathrm{T}}$ as in Eq.* (6.24)*. Given $C > 0$, the optimal Distance SVM classifier is found by solving the convex quadratic minimization problem*

$$\begin{array}{lcl}
\hat{\lambda} = & \underset{0\leq\lambda\leq C/N,\,\lambda\in\mathbb{R}^N}{\arg\min} & \frac{1}{2}\lambda^T\mathbf{D}\lambda - \lambda^T\mathbb{1}, \\
& \text{s.t.} & \beta^T y_{\mathrm{T}} = 0,
\end{array} \tag{6.31}$$

*with $\mathbf{D} = (\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}})^T(\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}})$ positive semi definite. The optimal weight vector $\hat{\beta}$ is then*

$$\hat{\beta} = \mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\hat{\lambda}, \tag{6.32}$$

*and the optimal bias term $\hat{\beta}_0$ can be computed from any training points $p_i,\ i \in [1, N]$ with corresponding $0 < \lambda_i < \frac{C}{N}$ as*

$$\hat{\beta}_0 = y_i - (\mathbf{Y}_{\mathrm{S}}\Upsilon_{\mathrm{S}}(p_i))^T\hat{\beta}. \tag{6.33}$$

*Proof.* The optimization problem for the Distance SVM model in Prob. (6.13) is convex as per Definition 2.3. Additionally, all the inequality constraints are affine, meaning Slater's Condition reduces to finding a feasible point for Prob. (6.13). A feasible point $(\tilde{\beta}, \tilde{\beta}_0, \tilde{\xi})$ always exists for arbitrary $\tilde{\beta}$ and $\tilde{\beta}_0$ by choosing

$$\tilde{\xi}_i = \max\{y_i\, f_{\mathrm{D}}(p_i \mid \tilde{\beta}, \tilde{\beta}_0) - 1,\ 0\},\ i = 1, \dots, N. \tag{6.34}$$

Thus, strong duality holds for Prob. (6.13) and the KKT conditions are necessary and sufficient for primal-dual optimality of the variables $(\beta, \beta_0, \xi, \lambda, \mu)$. To solve Prob. (6.13) we transform it to its Wolfe dual from Section 2.4. The KKT conditions from Eq. (2.4) applied to Prob. (6.13) are

$$\begin{aligned}
\nabla_\beta \mathcal{L}_{\mathrm{D}} &= \beta - \mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda = 0, && \text{(6.35a)}\\
\partial_{\beta_0} \mathcal{L}_{\mathrm{D}} &= -\lambda^T y_{\mathrm{T}} = 0, && \text{(6.35b)}\\
\partial_{\xi_i} \mathcal{L}_{\mathrm{D}} &= C/N - \lambda_i - \mu_i = 0,\ i = 1,\, \dots,\, N, && \text{(6.35c)}\\
y_i\, f_{\mathrm{D}}(p_i \mid \beta, \beta_0) - 1 + \xi_i &\geq 0,\ i = 1,\, \dots,\, N, && \text{(6.35d)}\\
\lambda_i(y_i\, f_{\mathrm{D}}(p_i \mid \beta, \beta_0) - 1 + \xi_i) &= 0,\ i = 1,\, \dots,\, N, && \text{(6.35e)}\\
\mu_i \xi_i &= 0,\ i = 1,\, \dots,\, N, && \text{(6.35f)}\\
\xi, \lambda, \mu &\geq 0. && \text{(6.35g)}
\end{aligned}$$

The vanishing gradient KKT condition in Eq. (6.35a) means that $\beta = \mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda$ for any primal-dual optimal $(\beta, \beta_0, \xi, \lambda)$. The condition in Eq. (6.35c) along with the non-negativity constraints on $\xi, \lambda$, and $\mu$ allows us to eliminate $\mu$ by constraining $\lambda$ to $0 \leq \lambda \leq C/N$ and implicitly setting $\mu_i = C - \lambda_i$.

Inserting the vanishing gradient condition on $\beta$ along with rest of the vanishing gradient KKT conditions from Eq. (6.35c) and Eq. (6.35c), which make the last two terms of the Lagrangian in Eq. (6.28) vanish, the expression for the Lagrangian conditioned on its gradient w.r.t the primal variables vanishing is

$$\begin{aligned}
\mathcal{L}_{\mathrm{D}} &= \frac{1}{2}(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda)^T(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda) - \sum_{i=1}^{N}\lambda_i \left(y_i(\mathbf{Y}_{\mathrm{S}}\Upsilon_{\mathrm{S}}(p_i))^T\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda - 1\right),\\
&= \frac{1}{2}\lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{Y}_{\mathrm{S}}^2\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda - \sum_{i=1}^{N}\lambda_i y_i \Upsilon_{\mathrm{S}}(p_i)^T\mathbf{Y}_{\mathrm{S}}^2\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \sum_{i=1}^{N}\lambda_i,\\
&= \frac{1}{2}\lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda - \left(\sum_{i=1}^{N}\lambda_i y_i \Upsilon_{\mathrm{S}}(p_i)^T\right)\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \lambda^T\mathbb{1},\\
&= \frac{1}{2}\lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda - \lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \lambda^T\mathbb{1},\\
&= -\frac{1}{2}\lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \lambda^T\mathbb{1},
\end{aligned} \tag{6.36}$$

where we've used that $Y_S^2 = I_M$, and whilst requiring $\lambda^T y_T = 0$. Defining the matrix

$$\mathbf{D} = (\mathbf{K}_D \mathbf{Y}_T)^T (\mathbf{K}_D \mathbf{Y}_T), \tag{6.37}$$

the Wolfe dual problem to Prob. (6.13) is

$$\begin{aligned} \max_{0 \le \lambda \le \frac{C}{N}, \lambda \in \mathbb{R}^N} \quad & -\frac{1}{2}\lambda^T \mathbf{D} \lambda + \lambda^T \mathbb{1}, \\ \text{s.t.} \quad & \lambda^T y_T = 0. \end{aligned} \tag{6.38}$$

This is a concave quadratic maximization problem, and by minimizing the objective with the opposite sign we get the optimization problem in Prob. (6.31). Its solution gives the dual-optimal Lagrange multipliers $\hat{\lambda}$.

We recover the primal optimal variables $\hat{\beta}$ from the vanishing gradient KKT condition in Eq. (6.35a). Then we recover $\hat{\beta}_0$ by solving the complementarity conditions in Eq. (6.35e) for $\beta_0$ for any margin points with $0 < \hat{\lambda}_i < C/N$, and for numerical stability we can the mean of all the values found for each margin point. Lastly, the primal-dual optimal values for $\xi$ and $\mu$ are determined by the primal-dual optimal $(\hat{\beta}, \hat{\beta}_0, \hat{\lambda})$. □

Due to the nature of the constraints for the problem in Prob. (6.13), we only get non-zero $\hat{\lambda}_i$ for those indices $i \in [1, N]$ which correspond to points which where not correctly classified by margin greater than $M = 1/||\hat{\beta}||$, i.e. $y_i f_D(p_i|\hat{\beta}, \hat{\beta}_0) - 1 \le 0$. This means that we can expect some degree of sparsity in the dual problem optimum $\hat{\lambda}$, but that is not the case for the resulting optimum weighting vector $\hat{\beta}$ which is a linear combination of the dense columns of $\mathbf{Y}_S \mathbf{K}_D \mathbf{Y}_T$ picked out by $\hat{\lambda}$.

We can also regard the above considerations as having a sparse representation for $\hat{\beta}$ among the set of vectors $\mathbf{Y}_S \Upsilon(p)$, $p \in \mathcal{X}_S$. However, to evaluate the classifier

$$f_D(q \mid \beta, \beta_0) = \beta_0 + (\mathbf{Y}_S \Upsilon(q))^T \beta \tag{6.39}$$

on a point $q \in \mathcal{M}$, we first map it to $\mathbb{R}^M$ as $q \mapsto \mathbf{Y}_S \Upsilon(q)$, an operation which is "dense" w.r.t. the set of support points, as it requires computing the distance to all the support points in $\mathcal{X}_S$.

The Lagrangian for the Zero Curvature Distance SVM optimization problem in Prob. (6.19) is closely related to the Lagrangian in Eq. (6.28), except that we introduce another Lagrangian multiplier $\nu \in \mathbb{R}$.

$$\begin{aligned}\mathcal{L}_{\mathrm{DZC}} &:= \mathcal{L}_{\mathrm{D}} - \nu y_{\mathrm{S}}^T \beta, \\ &= \frac{1}{2}||\beta||^2 + \sum_{i=1}^{N} \left( \frac{C}{N} - \lambda_i - \mu_i \right) \xi_i, \\ &- (\mathbf{Y}_{\mathrm{S}} \mathbf{K}_{\mathrm{D}} \mathbf{Y}_{\mathrm{T}} \lambda)^T \beta, \\ &- \beta_0 \lambda^T y_{\mathrm{T}} + \lambda^T \mathbb{1} - \nu y_{\mathrm{S}}^T \beta.\end{aligned} \tag{6.40}$$

With the added term in the Lagrangian we can again state a result concerning how to solve for the Zero Curvature Distance SVMmodel.

**Theorem 6.2** (Solving the Zero Curvature DSVM Optimization Problem)**.** *Let $(\mathcal{M}, \mathcal{A}^+, g)$ be a complete, path-connected Riemannian manifold. Denote a set of $N$ training points $\mathcal{X}_T$ and $2 \leq M \leq N$ support points $\mathcal{X}_S \subseteq \mathcal{X}_{\mathrm{T}}$ as in Eq. (6.24) with at least one training point of each class included in the set of support points. Given $C > 0$, the optimal Distance SVM classifier is found by solving the convex quadratic minimization problem*

$$\begin{aligned}\hat{\lambda},\ \hat{\nu} = \underset{\lambda \in \mathbb{R}^N, 0 \leq \lambda \leq C/N,\ \nu \in \mathbb{R}}{\arg\min} & \quad \frac{1}{2}\lambda^T \mathbf{D} \lambda - \lambda^T \mathbb{1} - \frac{M}{2}\nu^2 \\ \text{s.t.} & \\ \lambda^T y_{\mathrm{T}} &= 0, \\ y_{\mathrm{S}}^T \mathbf{Y}_{\mathrm{S}} \mathbf{K}_{\mathrm{D}} \mathbf{Y}_{\mathrm{T}} \lambda + \nu M &= 0,\end{aligned} \tag{6.41}$$

*where again $\mathbf{D} = (\mathbf{K}_{\mathrm{D}} \mathbf{Y}_{\mathrm{T}})^T (\mathbf{K}_{\mathrm{D}} \mathbf{Y}_{\mathrm{T}})$. The optimal weight vector $\hat{\beta}$ is then*

$$\hat{\beta} = \mathbf{Y}_{\mathrm{S}} \mathbf{K}_{\mathrm{D}} \mathbf{Y}_{\mathrm{T}} \hat{\lambda} + \hat{\nu} y_{\mathrm{S}}, \tag{6.42}$$

*and the optimal bias term $\hat{\beta}_0$ is computed from any training points $p_i,\ i \in [1, N]$ with corresponding $0 < \lambda_i < \frac{C}{N}$ as*

$$\hat{\beta}_0 = y_i - (\mathbf{Y}_{\mathrm{S}} \Upsilon_{\mathrm{S}}(p_i))^T \hat{\beta}. \tag{6.43}$$

*Proof.* The optimization problem for the Zero Curvature Distance SVM model in Prob. (6.19) is still convex as per Definition 2.3, with the addition of a single linear equality constraint as compared to Prob. (6.13). As all the inequality constraints are affine Slater's Condition reduces to finding a feasible point for Prob. (6.19). A feasible point $(\tilde{\beta}, \tilde{\beta}_0, \tilde{\xi})$ always exists for arbitrary $\tilde{\beta}_0$ by choosing any $\tilde{\beta}$ s.t. $\tilde{\beta}^T y_{\mathrm{S}} = 0$, which is possible for non-trivial $\tilde{\beta}$ as there is at least a single support point of each class. Lastly we can choose

$$\tilde{\xi}_i = \max\{y_i f_{\mathrm{D}}(p_i \mid \tilde{\beta}, \tilde{\beta}_0) - 1,\ 0\},\ i = 1, \ldots, N. \tag{6.44}$$

Thus, strong duality holds for Prob. (6.19), and the KKT conditions are necessary for primal-dual optimality of the variables $(\beta, \beta_0, \xi, \lambda, \mu, \nu)$. To solve Prob. (6.19) we transform it to its Wolfe dual from Section 2.4. Although the Wolfe dual formulation does not explicitly include linear equality constraints we can rewrite the zero curvature equality constraint in manner inspired by [33, p. 221] as two inequality constraints with their respective non-negative Lagrange multipliers $\nu_+$ and $\nu_-$. If we then set $\nu = \nu_+ - \nu_-$ we've effectively introduced the equality constraint to the Wolfe dual problem formulation, with the modification that the corresponding Lagrange multiplier is not constrained to be non-negative.

The KKT conditions from Eq. (2.4) applied to Prob. (6.19) are

$$\nabla_\beta \mathcal{L}_\mathrm{D} = \beta - \mathbf{Y}_\mathrm{S} \mathbf{K}_\mathrm{D} \mathbf{Y}_\mathrm{T} \lambda - \nu y_\mathrm{S} = 0, \tag{6.45a}$$

$$\partial_{\beta_0} \mathcal{L}_\mathrm{D} = -\lambda^T y_\mathrm{T} = 0, \tag{6.45b}$$

$$\partial_{\xi_i} \mathcal{L}_\mathrm{D} = C/N - \lambda_i - \mu_i = 0,\ i = 1, \dots, N, \tag{6.45c}$$

$$y_\mathrm{S}^T \beta = 0, \tag{6.45d}$$

$$y_i\, f_\mathrm{D}(p_i \mid \beta, \beta_0) - 1 + \xi_i \geq 0,\ i = 1, \dots, N, \tag{6.45e}$$

$$\lambda_i (y_i\, f_\mathrm{D}(p_i \mid \beta, \beta_0) - 1 + \xi_i) = 0,\ i = 1, \dots, N, \tag{6.45f}$$

$$\mu_i \xi_i = 0,\ i = 1, \dots, N, \tag{6.45g}$$

$$\xi, \lambda, \mu \geq 0. \tag{6.45h}$$

The vanishing gradient KKT condition in Eq. (6.45a) means that $\beta = \mathbf{Y}_\mathrm{S} \mathbf{K}_\mathrm{D} \mathbf{Y}_\mathrm{T} \lambda + \nu y_\mathrm{S}$ for any primal-dual optimal $(\beta, \beta_0, \xi, \lambda, \nu)$. Together with the zero curvature condition in Eq. (6.45d) we get the relation

$$y_\mathrm{S}^T (\mathbf{Y}_\mathrm{S} \mathbf{K}_\mathrm{D} \mathbf{Y}_\mathrm{T} \lambda + \nu y_\mathrm{S}) = y_\mathrm{S}^T \mathbf{Y}_\mathrm{S} \mathbf{K}_\mathrm{D} \mathbf{Y}_\mathrm{T} \lambda + \nu y_\mathrm{S}^T y_\mathrm{S} = 0. \tag{6.46}$$

As all the elements of $y_\mathrm{S}$ equal $\pm 1$, $y_\mathrm{S}^T y_\mathrm{S} = ||y_\mathrm{S}||_2^2 = M$. As for the DSVM model the condition in Eq. (6.35c) along with the non-negativity constraints on $\xi$, $\lambda$, and $\mu$ allows us to eliminate $\mu$ by constraining $\lambda$ to $0 \leq \lambda \leq C/N$ and implicitly setting $\mu_i = C - \lambda_i$.

Inserting the vanishing gradient conditions from the KKT conditions the expression for the Lagrangian conditioned on its gradient w.r.t the primal variables vanishing is

$$\mathcal{L}_\mathrm{ZCD} = \frac{1}{2} ||\mathbf{Y}_\mathrm{S} \mathbf{K}_\mathrm{D} \mathbf{Y}_\mathrm{T} \lambda + \nu y_\mathrm{S}||_2^2 - \sum_{i=1}^{N} \lambda_i y_i (\mathbf{Y}_\mathrm{S} \Upsilon_\mathrm{S}(p_i))^T (\mathbf{Y}_\mathrm{S} \mathbf{K}_\mathrm{D} \mathbf{Y}_\mathrm{T} \lambda + \nu y_\mathrm{S}) + \lambda^T \mathbb{1}. \tag{6.47}$$

The first term of $\mathcal{L}_{\mathrm{ZCD}}$ can be expanded as

$$\begin{aligned}
||\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu y_{\mathrm{S}}||_2^2 &= (\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu y_{\mathrm{S}})^T(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu y_{\mathrm{S}}), \\
&= \lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda)^T y_{\mathrm{S}}, \\
&\quad + \nu y_{\mathrm{S}}^T\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu^2 y_{\mathrm{S}}^T y_{\mathrm{S}}, \\
&= \lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + 2\nu y_{\mathrm{S}}^T\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu^2 y_{\mathrm{S}}^T y_{\mathrm{S}}.
\end{aligned} \tag{6.48}$$

The second term can likewise be expanded as

$$\begin{aligned}
&\sum_{i=1}^{N}\lambda_i y_i(\mathbf{Y}_{\mathrm{S}}\Upsilon_{\mathrm{S}}(p_i))^T(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu y_{\mathrm{S}}) \\
&\qquad = \sum_{i=1}^{N}\lambda_i y_i(\mathbf{Y}_{\mathrm{S}}\Upsilon_{\mathrm{S}}(p_i))^T\,(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda) \\
&\qquad\quad + \nu\sum_{i=1}^{N}\lambda_i y_i(\mathbf{Y}_{\mathrm{S}}\Upsilon_{\mathrm{S}}(p_i))^T y_{\mathrm{S}}, \\
&\qquad = \lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu(\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda)^T y_{\mathrm{S}}.
\end{aligned} \tag{6.49}$$

Combining the first two terms and using that $y_{\mathrm{S}}^T\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda = (\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda)^T y_{\mathrm{S}} \in \mathbb{R}$ we get the final expression for the ZC-DSVM Lagrangian conditioned on the primal gradients vanishing as

$$\mathcal{L}_{\mathrm{ZCD}} = -\frac{1}{2}\lambda^T\mathbf{Y}_{\mathrm{T}}\mathbf{K}_{\mathrm{D}}^T\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \lambda^T\mathbb{1} + \nu^2 y_{\mathrm{S}}^T y_{\mathrm{S}}, \tag{6.50}$$

whilst requiring that $\lambda^T y_{\mathrm{T}} = 0$ and $y_{\mathrm{S}}^T\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu M = 0$. The Wolfe dual to the ZC-DSVM optimization problem is thus

$$\begin{aligned}
\max_{\lambda \in \mathbb{R}^N, 0 \le \lambda \le C/N,\ \nu \in \mathbb{R}} \quad & -\frac{1}{2}\lambda^T\mathbf{D}\lambda + \lambda^T\mathbb{1} + \frac{M}{2}\nu^2 \\
\text{s.t.} \quad & \lambda^T y_{\mathrm{T}} = 0, \\
& y_{\mathrm{S}}^T\mathbf{Y}_{\mathrm{S}}\mathbf{K}_{\mathrm{D}}\mathbf{Y}_{\mathrm{T}}\lambda + \nu M = 0.
\end{aligned} \tag{6.51}$$

The problem in Eq. (6.51) is concave quadratic with a convex feasible set, so again if we switch the sign of the objective and minimize we get the convex quadratic optimization problem in Eq. (6.41). Solving it returns the dual-optimal Lagrange multipliers $\hat{\lambda}$ and $\hat{\nu}$. From which we can recover the primal-optimal $\hat{\beta}$ using the vanishing gradient condition in Eq. (6.45a). And exactly as for the DSVM optimization we recover the primal optimal $\hat{\beta}_0$ by solving the complementarity conditions in Eq. (6.45f) for $\beta_0$ for any margin points with $0 < \hat{\lambda}_i < C/N$. And finally the primal-dual optimal values for $\xi$ and $\mu$ are determined by $\hat{\beta}$, $\hat{\beta}_0$ and $\hat{\lambda}$. □

## 6.4 Heuristic for Sparse Distance SVM Models

In principle, we could optimize over which subset of the available training points to use as the support points for the DSVM and ZC-DSVM model, in order to find a classifier with a sparse set of support points. However, this is a difficult combinatorial problem, and as a heuristic we propose the following two-step method. First, we optimize the DSVM models with the full set of training points as support points. Then we perform a second optimization where the support points are chosen as the points $p_i$ for which the corresponding $\hat{\lambda}_i$ was non-zero in the full model.

Phrased differently, we first train the full model, and then choose the subset of training points $p_i$ which ended up at the margin, or were under-/misclassified by the full model as the set of support points for our sparse classifier, and perform a second optimization. We call the models by applying this two-step sparsity heuristic to the DSVM and ZC-DSVM models *Sparse Distance SVM* (Sp-DSVM) and *Sparse Zero Curvature Distance SVM* (Sp-ZCDSVM), respectively.

This procedure is more expensive than simply solving for the DSVM and ZCDSVM models using all the training points as support points, but by fitting the full classifier the hope is that we're able to extract the training points most relevant to separate the training data well. And by retraining the DSVM and ZCDSVM models with this reduced set of support points we have in a sense reduced the flexibility of the classifiers, which will hopefully reduce the risk of overfitting to the training data.

# 7 Numerical Experiments

To compare the Distance SVM models with the Euclidean SVM model and see how they compare with the existing manifold SVM models, we implement and test the models numerically. First we compare the DSVM models with the classical SVM model on artificial datasets on $\mathbb{R}^2$, then compare the DSVM models with the existing manifold SVM models on artificial datasets on $\mathbb{S}(2)$, before concluding with a real world case comparing the test accuracy of all the manifold SVM models on the BCI-IV 2a dataset [39].

This implementation of all the different models is done with the `Julia` programming language [34], version 1.7. The functions needed to work on the manifolds $\mathbb{S}(n)$, $(\mathcal{P}(n), g^{\mathrm{LA}})$ and $(\mathcal{P}(n), g^{\mathrm{LE}})$, i.e. the exponential and logarithmic mappings, the Riemannian metrics, and the Riemannian distance functions, are all implemented in the `Manifolds.jl, v0.75` [35] package. Furthermore, methods and algorithms to perform gradient descent with Armijo line search on arbitrary Riemannian manifolds are implemented in the `Manopt.jl, v0.3.27` [36] package. To solve euclidean constrained optimization problems we've employed the optimization modeling package `JuMP.jl v1.1.0`, in conjunction with the `Ipopt.jl, v1.0.2` solver [19]. All experiments are performed on a laptop running Ubuntu 20.04 with an eight core Intel(R) Core(TM) i7-1065G7 @ 1.30GHz CPU and 16 GB of RAM.

To illustrate the separating curves and margin curves of a differentiable classifier $f\colon \mathcal{M} \to \mathbb{R}$ on a two-dimensional Riemannian manifold $(\mathcal{M}, g)$, e.g. $\mathbb{R}^2$ and $\mathbb{S}(2)$, we first find an initial point $p_0 \in \mathcal{M}$ for which $f(p_0) = \delta$. On all our figures $\delta \in \{0, \pm 1\}$, except for the CP-SVM classifiers where $\delta \in \{0, \pm k\}$, with $k$ the desired margin hyperparameter of the CP-SVM model from 5.3. In order to construct an ordinary differential equation whose solution lies along level sets of $f$, consider the tangent vector

$$N_f(p) \in T_p\mathcal{M}, \tag{7.1}$$

constructed by rotating the gradient of the classifier $\operatorname{grad} f(p)$ by $\pi/2$ clockwise in $T_p\mathcal{M}$. That means

$$g_p(N_f(p), \operatorname{grad} f(p)) = 0, \, \forall \, p \in \mathcal{M}$$

and $||N_f(p)||_p = ||\operatorname{grad} f(p)||_p$. Then we numerically solve the ordinary differential equation

$$\dot{\gamma}(t) = N_f(p), \quad \gamma(0) = p_0, \tag{7.2}$$

for the curve $\gamma\colon [-T, T] \to \mathcal{M}$, $T > 0$, using `ManifoldDiffEq.jl, v0.1.2` [41]. This package implements several numerical ODE methods tailed to differential equations on manifolds, from among others [37]. The time derivative of $f(\gamma(t))$ is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t)) &= \mathrm{D}f(\gamma(t))[\dot{\gamma}(t)] = g_p\left(\operatorname{grad} f(\gamma(t)), \dot{\gamma}(t)\right) \\ &= g_p\left(\operatorname{grad} f(\gamma(t)), N_f(\gamma(t))\right) = 0, \end{aligned} \tag{7.3}$$

meaning the curve $\gamma$ is a level set curve for $f$ with $f(\gamma(t)) = \delta,\ t \in [-T, T]$.

## 7.1 DSVM vs. Euclidean SVM on the Plane

To better understand the differences between the Distance SVM models as compared to the classical SVM (Eucl. SVM) model we present the different models applied to three different datasets on the plane $\mathbb{R}^2$ in Figs. 7.1 to 7.3. As we're working on the Hilbert space $\mathbb{R}^2$ the considerations in Section 6.2 apply, and the zero curvature DSVM models are guaranteed to result in linear classifiers $f_{\mathrm{ZCD}}(x \mid \beta, \beta_0)$.

We train the Euclidean SVM model on each dataset and compare the Distance SVM model with and without the zero curvature condition and heuristic sparsity method from Section 6.4 applied, and we repeat here that we denote them as ”DSVM”, ”Sp-DSVM”, ”ZCDSVM”, and ”Sp-ZCDSVM”, respectively. Additionally, we train a zero curvature DSVM model where the support points $\mathcal{X}_S$ for the classifier are chosen as the support vectors of the Euclidean SVM model, and we call this model ”Eucl. Supp. ZCDSVM”. For consistency all models were trained with a misclassification penalty parameter of $C = 10^3$.

The dataset in Fig. 7.1 is linearly separable, with good class separation. Looking at the top two frames of Fig. 7.1 we see that the ”Eucl. Supp. ZCDSVM” separator is close to the same as the Euclidean SVM separator, but differs slightly in the direction of the hyperplane normal vector and margin points, even when the models use the same support points. In the middle two frames showing the DSVM and Sp-DSVM models we can also see the effect of the quadratic dependence on $x$ from Eq. (6.14) manifesting itself as parabolic level sets of the classifier. This curvature is also more pronounced for the sparse Sp-DSVM model.

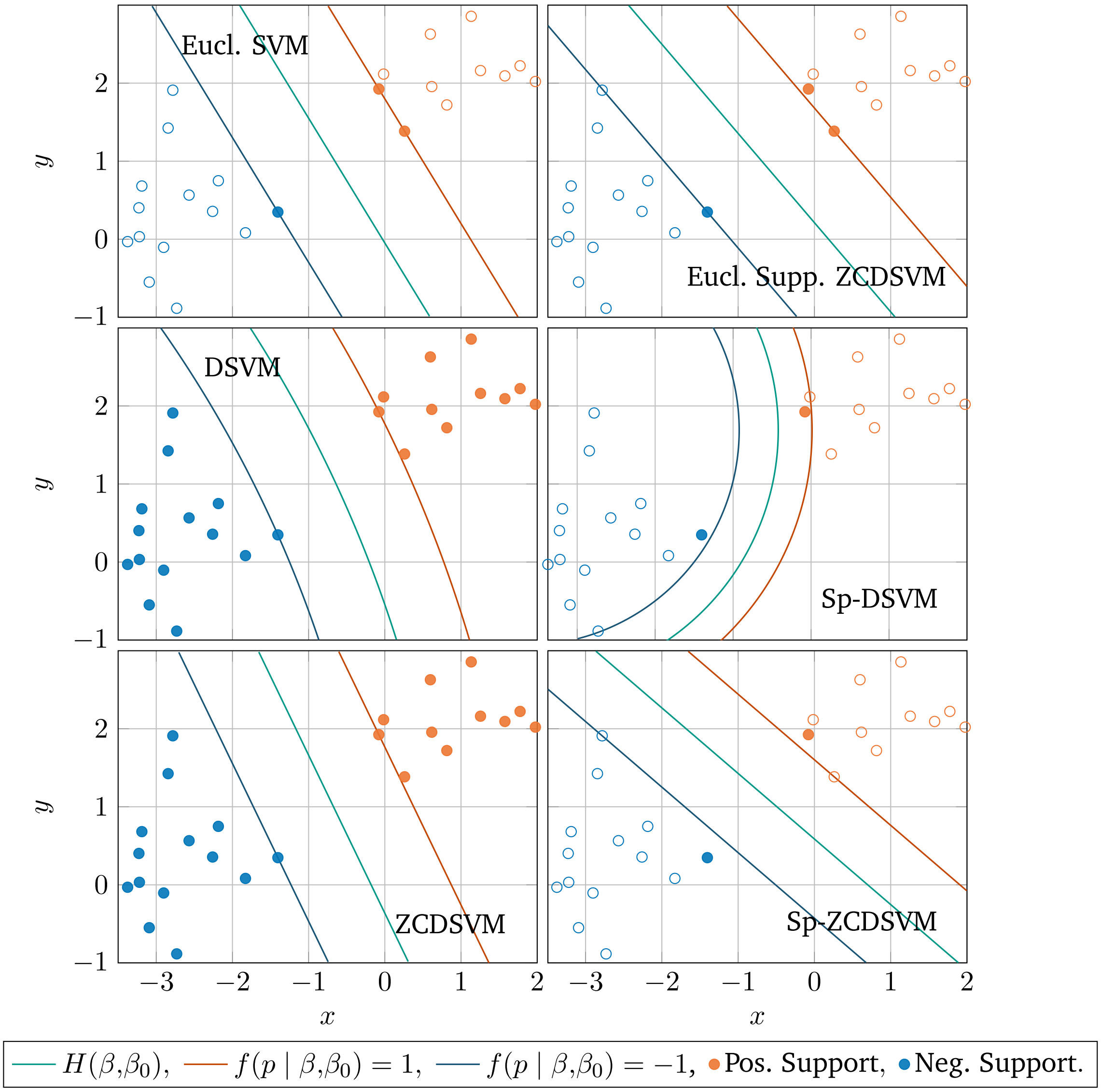

**Fig. 7.1.:** Easily separable dataset. Comparing the DSVM models with the Euclidean SVM model on $\mathbb{R}^2$. All models optimized with misclassification penalty $C = 10^3$.

In Fig. 7.2 the dataset is linearly separable, but the margin of the maximum margin linear separator is a lot smaller than in Fig. 7.1. And with the misclassification penalty $C = 10^3$ we see that the Euclidean SVM separator misclassifies a single negative point. However, on this dataset the "Eucl. Supp. ZCDSVM" linear separator recreates the Euclidean SVM separator up to a difference in reconstructed parameter norms of $10^{-6}$. That is,

$$||\beta_{\text{SVM}} - \boldsymbol{n}_{\text{D}}||_2 + |\beta_0 - \tilde{\beta}_0|| < 10^{-6}, \tag{7.4}$$

using the notation from Eq. (6.15) for the reconstructed normal vector and bias of a zero curvature DSVM model. Furthermore, the ZCDSVM separator in Fig. 7.2 is actually the maximum margin separator of the dataset, which we confirmed by training the classical

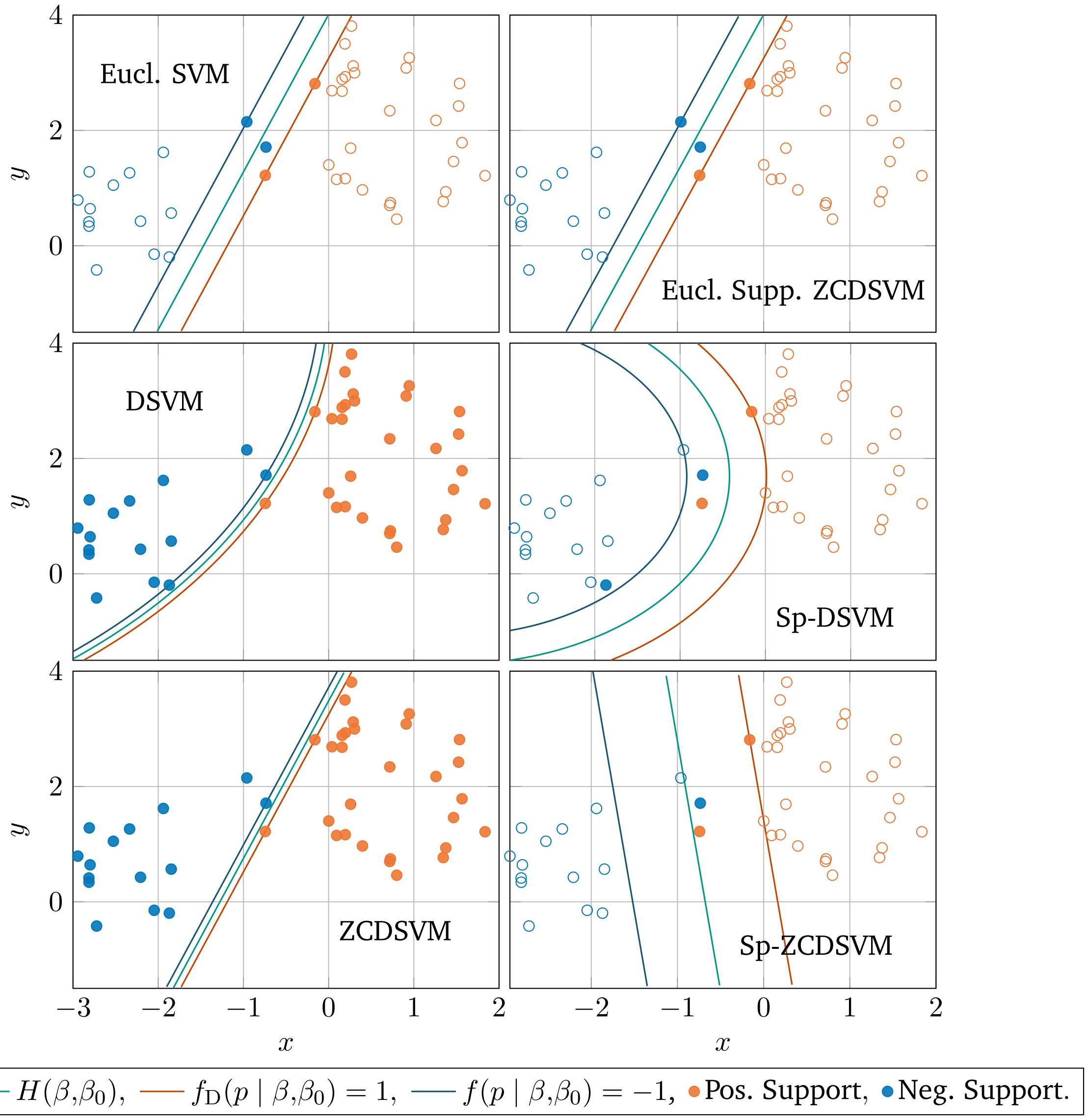

**Fig. 7.2.:** Barely linearly separable dataset. Comparing the DSVM models with the Euclidean SVM model on $\mathbb{R}^2$. All models optimized with misclassification penalty $C = 10^3$.

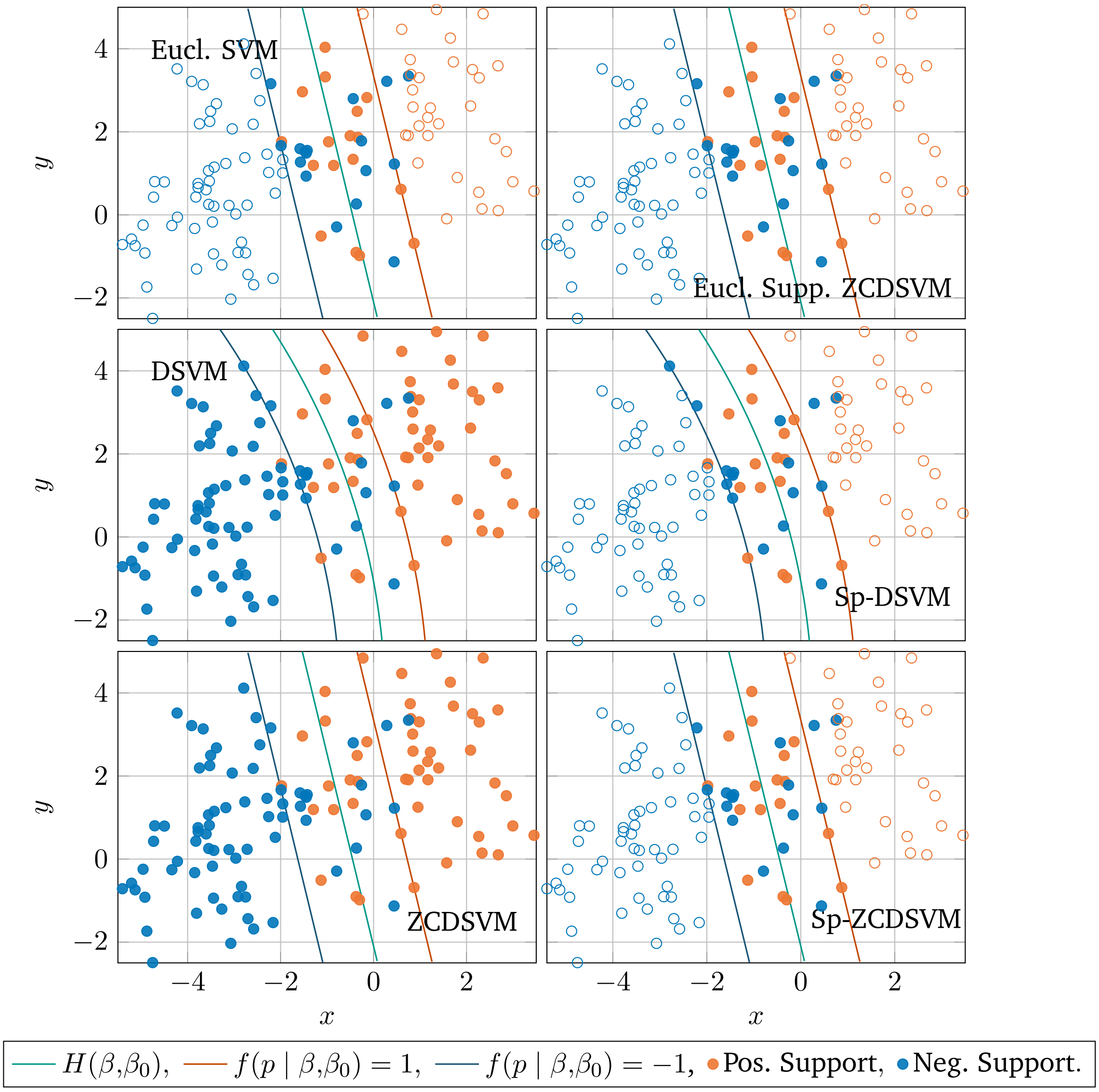

**Fig. 7.3.:** Non-linearly separable dataset. Comparing the DSVM models with the Euclidean SVM model on $\mathbb{R}^2$. All models optimized with misclassification penalty $C = 10^3$.

SVM on the same dataset with the misclassification penalty set to $C = 10^6$, in which case we get the same linear separator as shown in the lowermost left frame of Fig. 7.2.

The last dataset on $\mathbb{R}^2$ in Fig. 7.3 is not linearly separable, in contrast to the previous two datasets. And again we see the same behavior as in the previous figure, where the Eucl. Supp. ZCDSVM classifier matches the Euclidean SVM classifier exactly. Interestingly, we see that the sparse versions of both the DSVM and ZCDSVM models recreate exactly the same classifiers, even though they are based on a restricted subset of support points. This seems to indicate that the re-optimization sparsity heuristic works well, and the training points which are on the margin or under classified by the full model are the ones most relevant for the final classifier as well. And comparing with the Euclidean SVM classifier again, we note that the linear classifiers of both zero curvature DSVM models are exactly the same as the Euclidean SVM classifier.

## 7.2 Manifold SVM Models on the Two-Sphere

In order to illustrate the existing manifold SVM models and the Distance SVM models, we generated two data sets on $\mathbb{S}(2)$, one easily separable and one with points of each class overlapping. Then we trained the models on each dataset, and demonstrate the effect of varying the parameter values for existing manifold SVM models in Figs. 7.4 to 7.17.

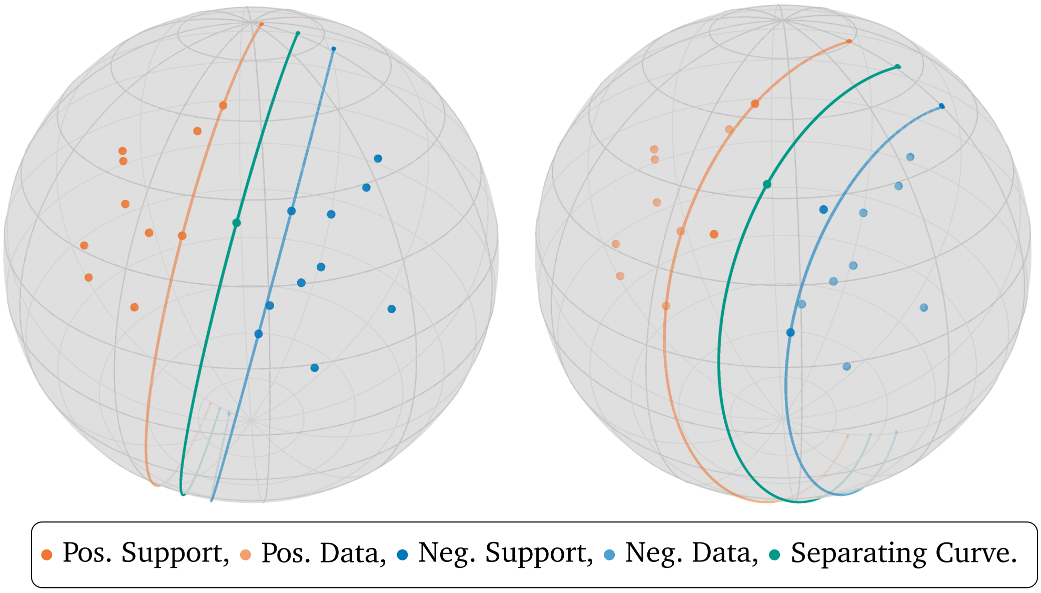

**Fig. 7.4.:** Distance SVM classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: DSVM model, right: Sp-DSVM model. Misclassification penalty $C_{\text{DSVM}} = 10^3$.

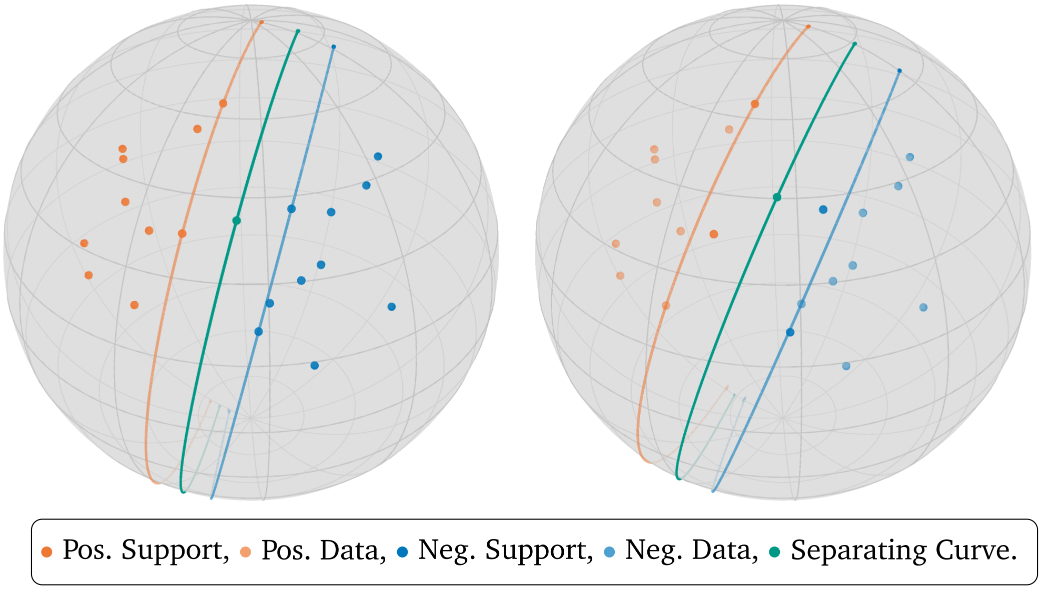

**Fig. 7.5.:** Zero curvature DSVM classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: ZCDSVM model, right: Sp-ZCDSVM model. Misclassification penalty $C_{\text{DSVM}} = 10^3$.

In figures Figs. 7.4 and 7.5 we show the DSVM models and ZCDSVM models trained on the easily separable dataset, with the full models on the left, and the sparse models on the right, all with $C_{\mathrm{DSVM}} = 10^3$. The full DSVM and ZCDSVM models turn out to be visually identical in this case.

Looking at the sparse DSVM classifiers, the zero curvature classifier appears noticeably ”straighter” than its counterpart. It is unclear how to quantify this ”straightness”, but visually the zero curvature separating curve $H(\beta, \beta_0)$ looks close to a great circle on the sphere, in which case it could be realized by a geodesic. However, the margin curves where $f_{\mathrm{D}}(p \mid \beta, \beta_0) = \pm 1$ are clearly not great circles based on visual inspection. We also note that the sparse DSVM models recover the full classifiers if we increase $C_{\mathrm{DSVM}}$ to $10^4$.

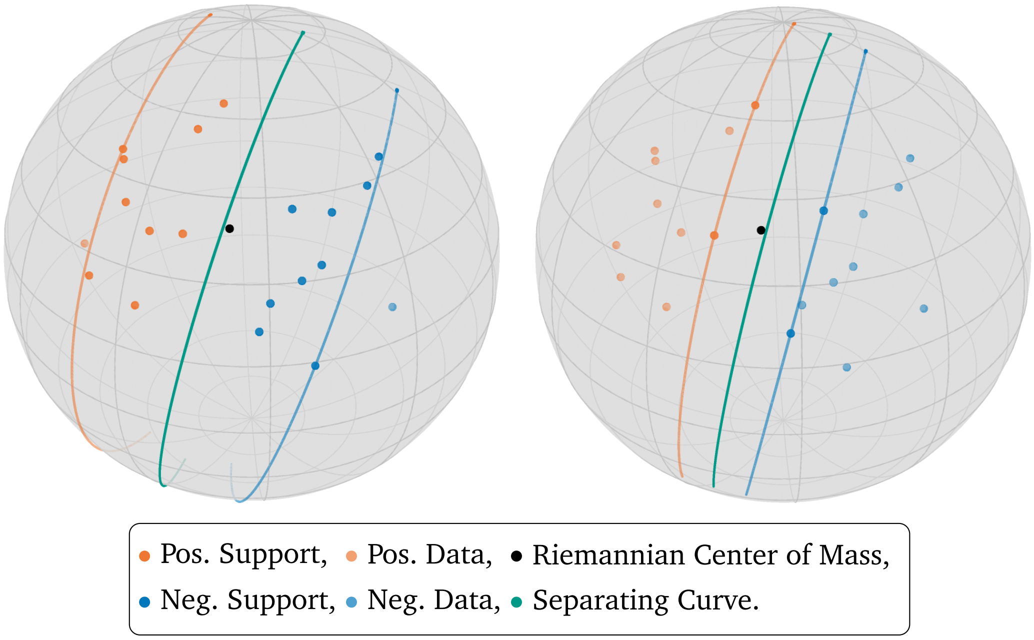

**Fig. 7.6.:** TS-SVM classifier on $\mathbb{S}(2)$, trained on a separable dataset. Left: $C_{\mathrm{TS-SVM}} = 10^1$, right: $C_{\mathrm{TS-SVM}} = 10^3$.

The TS-SVM model trained on the same easily separable dataset is shown in Fig. 7.6, with a lower misclassification penalty of $C_{\mathrm{TS-SVM}} = 10^1$ on the left and $C_{\mathrm{TS-SVM}} = 10^3$ on the right. The point in which tangent space we’re applying the classical SVM model is marked as a black point on the figures. As expected the margins are greater in the left frame of Fig. 7.6 as we penalize under classification less (i.e. $y_i f_{\mathrm{TS-SVM}}(p_i \mid p_{\mathrm{ref}}, \lambda_i, \beta_0) < 1$). However, when $C_{\mathrm{TS-SVM}} = 10^3$ the TS-SVM classifier is visually very similar to the full DSVM and ZCDSVM classifiers from the left frames of Figs. 7.4 and 7.5.

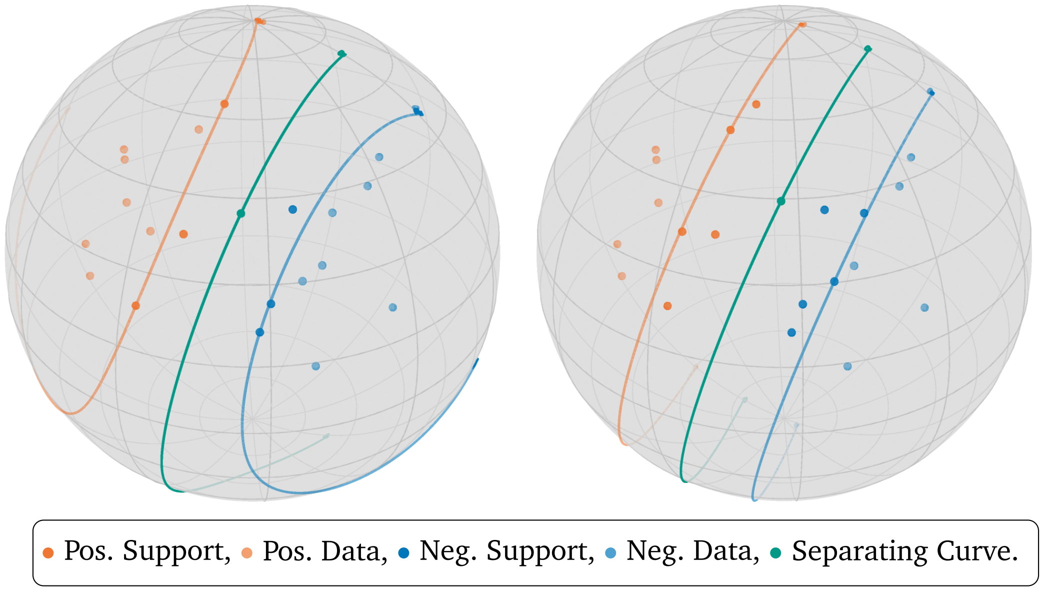

**Fig. 7.7.:** Manifold RBF classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $\sigma^2 = 1/2$, right: $\sigma^2 = 2$. Misclassification penalty set to $C_{\mathrm{MRBF}} = 10^2$ for both.

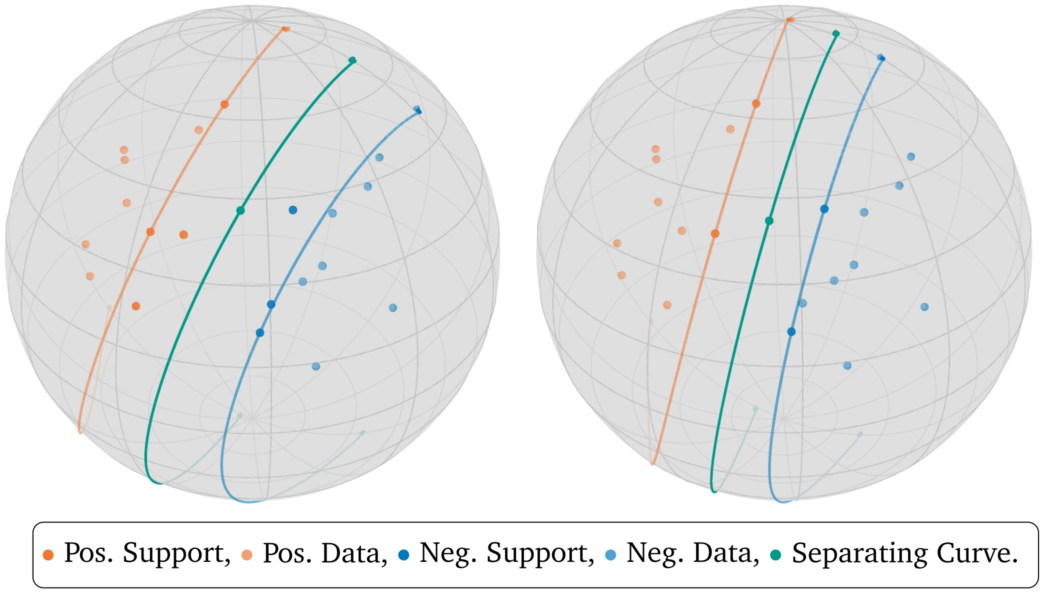

**Fig. 7.8.:** Manifold RBF classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $C_{\mathrm{MRBF}} = 10^2$, right: $C_{\mathrm{MRBF}} = 10^4$. Variance set to $\sigma^2 = 1$ for both.

To illustrate the parameter dependence of the MRBF model on the easily separable dataset on $\mathbb{S}(2)$ we've varied the variance $\sigma^2$ in Fig. 7.7 with fixed $C_{\mathrm{MRBF}} = 10^2$, and varied the misclassification penalty whilst keeping the variance fixed at $\sigma^2 = 1$ in Fig. 7.8. Even though the guarantee of a positive definite kernel from 5.2 do not apply on $\mathbb{S}(2)$ as it is not isometrically embeddable in a Hilbert space, we can still optimize the classical SVM model as long as the kernel matrix for our specific dataset is positive semi definite.

From Fig. 7.7 it is clear that the variance has quite a big impact on the shape of the separating curves. We see pronounced curvature in the margin curves for the MRBF classifier with $\sigma^2 = 1/2$, whilst the separating lines for the classifier with $\sigma^2 = 2$ are straighter. This behavior is consistent with the characteristics of the classical SVM model using the RBF kernel, which is capable of generating non-linear separating lines when they're projected down to the original feature space from the infinite dimensional RKHS of the RBF kernel. Changing the misclassification penalty $C_{\text{MRBF}}$ in Fig. 7.7 also produces the expected result of tighter margins, and in the right frame the MRBF classifier is able to separate the training data.

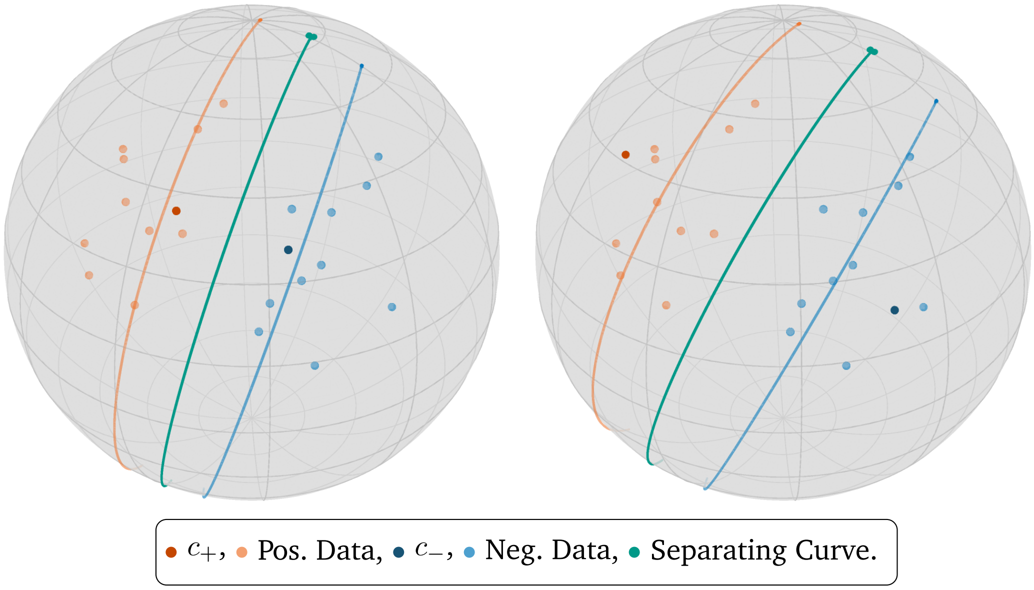

**Fig. 7.9.:** Control Point SVM classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $k = \pi/20$, right; $k = \pi/5$. Misclassification penalty set to $C_{\text{CP}-\text{SVM}} = 10$ for both. Margin lines drawn at $f_{\text{CP}}(q \mid c_+, c_-) = \pm k$.

The CP-SVM classifiers trained on the easily separable dataset are illustrated in Figs. 7.9 and 7.10, where we vary the desired margin $k$ and the misclassification penalty $C$ respectively. We observe in Fig. 7.9 that when we increase the desired classification margin $k$ the control points move farther away from one another. And in Fig. 7.10 the increased misclassification penalty $C_{\text{CP}-\text{SVM}}$ also moves the control points farther away from one another, and thereby decreases the Riemannian distance between the separating curve and the margin curves. This illustrates the complex interplay between the hyperparameters $k$ and $C_{\text{CP}-\text{SVM}}$.

A quirk of the CP-SVM model on $\mathbb{S}(2)$ is that when $k$ is set to a value greater than $\pi/5$ the control points have a tendency of ending up on the opposite side of the sphere from the training points. There they can achieve very small distance between the control points, and still classify training points on the other side of the sphere, but in a sense they're no longer representative of the distribution of training points as intended. This behavior can be understood as a consequence of the underlying manifold being curved, and would not be possible on flat euclidean vector spaces.

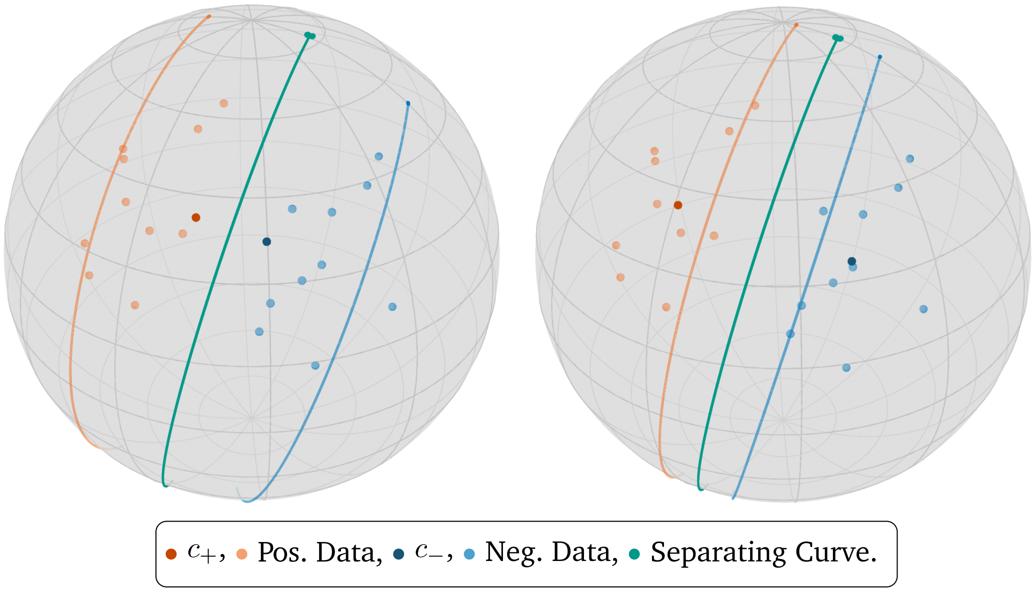

**Fig. 7.10.:** Control Point SVM classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $C_{\text{CP}-\text{SVM}} = 1$, right: $C_{\text{CP}-\text{SVM}} = 20$. Desired margin set to $k = \pi/16$ for both. Margin lines drawn at $f_{\text{CP}}(q \mid c_+, c_-) = \pm k$.

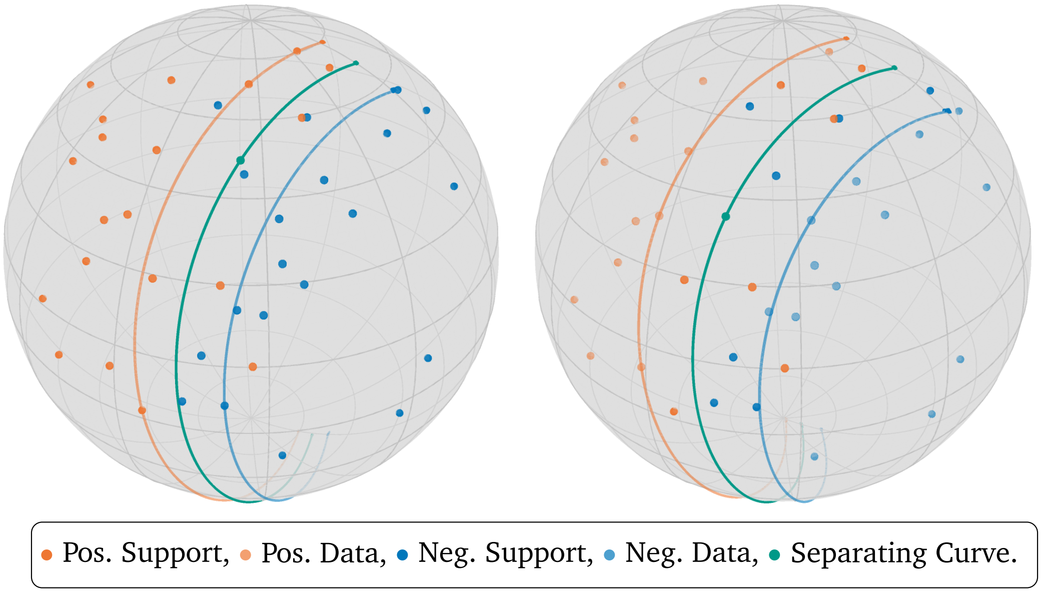

**Fig. 7.11.:** Distance SVM classifiers on $\mathbb{S}(2)$. Trained on overlapping dataset. Left: DSVM model, right: Sp-DSVM model. Misclassification penalty $C_{\text{DSVM}} = 10^3$.

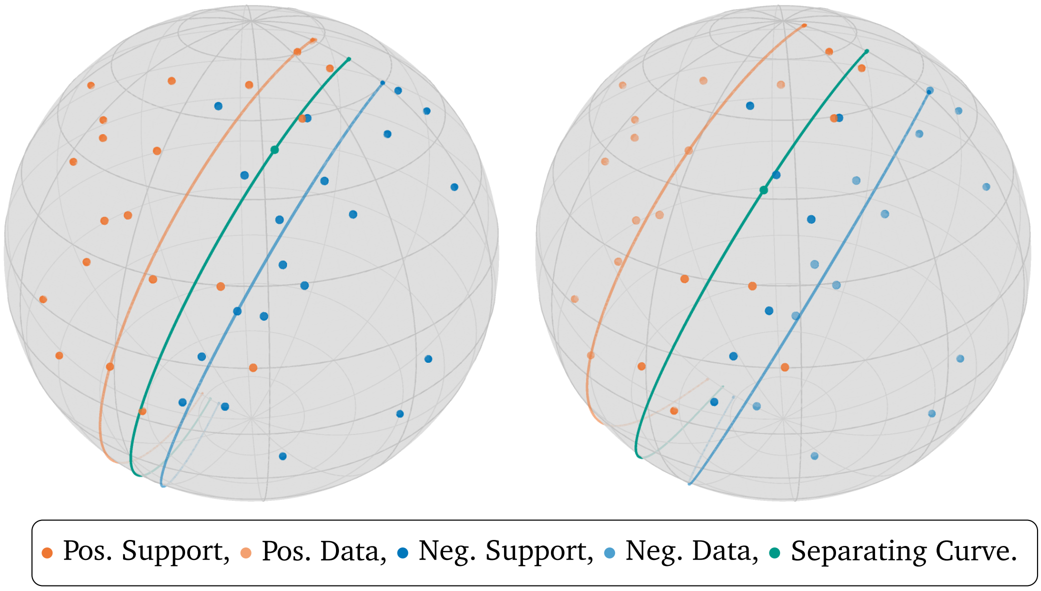

Fig. 7.12.: Zero curvature DSVM classifiers on $\mathbb{S}(2)$. Trained on a harder to separable dataset. Left: ZCDSVM model, right: Sp-ZCDSVM model. Misclassification penalty $C_{\text{DSVM}} = 10^3$.

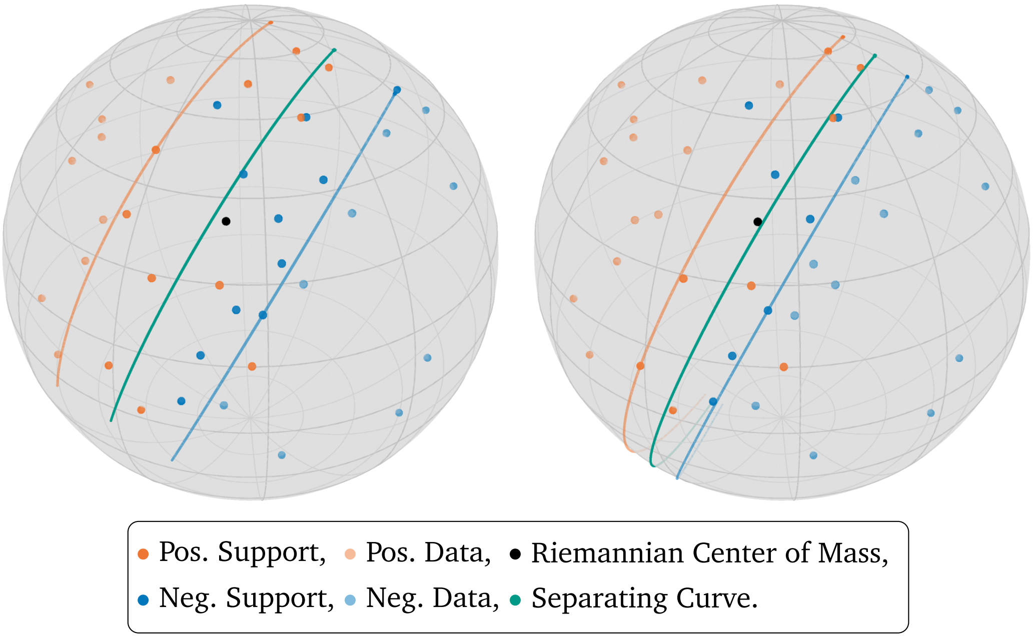

Fig. 7.13.: TS-SVM classifier on $\mathbb{S}(2)$, trained on the overlapping dataset. Left: $C_{\text{TS-SVM}} = 10^2$, right: $C_{\text{TS-SVM}} = 10^4$.

Switching over to considering models trained on a dataset where the training points from each class overlap and is harder to separate, the resulting DSVM models trained on this second dataset are shown in Figs. 7.11 and 7.12. On this dataset the DSVM and ZCDSVM classifiers are not identical, and we see that the zero curvature condition again has the effect of ”straightening” the separating curve and margin curves. The same effect is present for the sparse DSVM and sparse ZCDSVM classifiers on this dataset, but when we increase $C_{\mathrm{DSVM}}$ to $10^5$ all four classifiers coalesce to the same set of support points and appear visually quite similar.

The TS-SVM model trained on the overlapping dataset for two different values of $C_{\mathrm{TS-SVM}}$ are shown in Fig. 7.13, and they are both quite similar to the ZCDSVM classifiers in Fig. 7.12. In particular the TS-SVM classifier trained with $C_{\mathrm{TS-SVM}} = 10^4$ ends up using the same support points as the sparse ZC DSVM classifier.

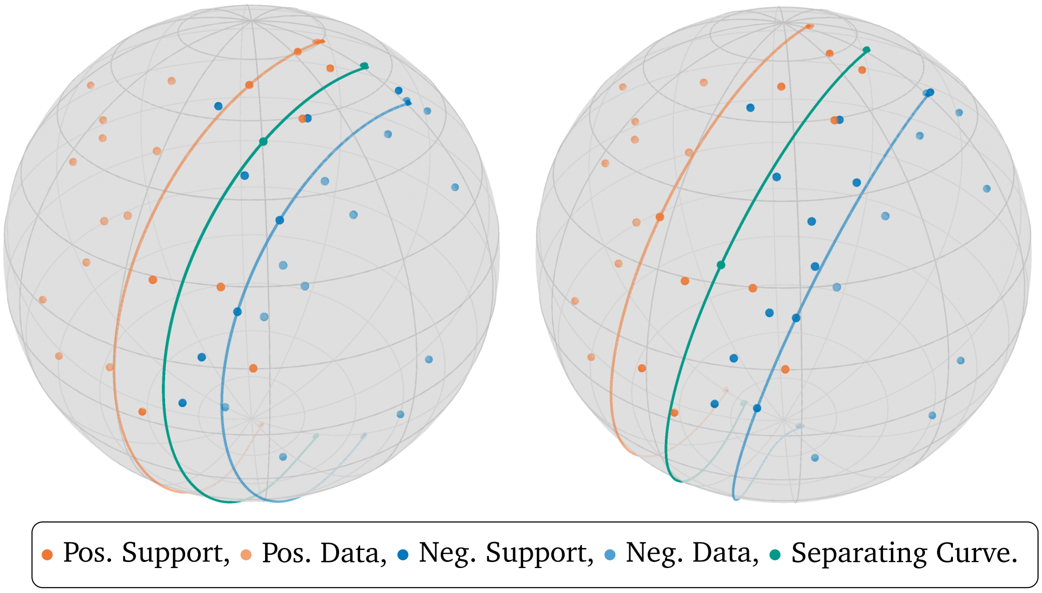

**Fig. 7.14.:** Manifold RBF classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $\sigma^2 = 2^1$, right: $\sigma^2 = 2^3$. Misclassification penalty set to $C_{\mathrm{MRBF}} = 10^3$ for both.

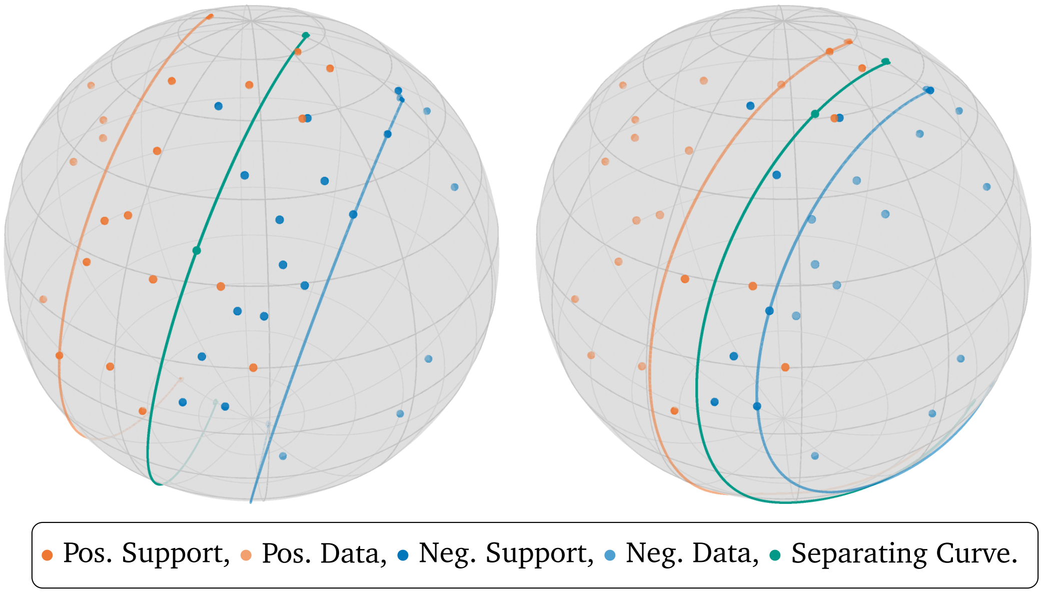

**Fig. 7.15.:** Manifold RBF classifiers on $\mathbb{S}(2)$. Trained on overlapping dataset. Left: $C_{\mathrm{MRBF}} = 10^2$, right: $C_{\mathrm{MRBF}} = 10^4$. Variance set to $\sigma^2 = 2^2$ for both.

Training the MRBF model on the overlapping dataset and varying the variance and misclassification penalty hyperparameters we get Figs. 7.14 and 7.15. Of note is the interplay between the hyperparameters. The classifier trained with the lower variance $\sigma^2 = 2$ and $C_{\mathrm{MRBF}} = 10^3$ in the left frame of Fig. 7.14 is very similar to the classifier in the right frame of Fig. 7.15 with $\sigma^2 = 2^2$ and $C_{\mathrm{MRBF}} = 10^4$. The classifier in the right frame of Fig. 7.14 with higher variance also generates a separating curve that is quite similar to the separating curve for the classifier in the left frame of Fig. 7.15, however the latter classifier has a larger distance between the separator curve and the margin curves.

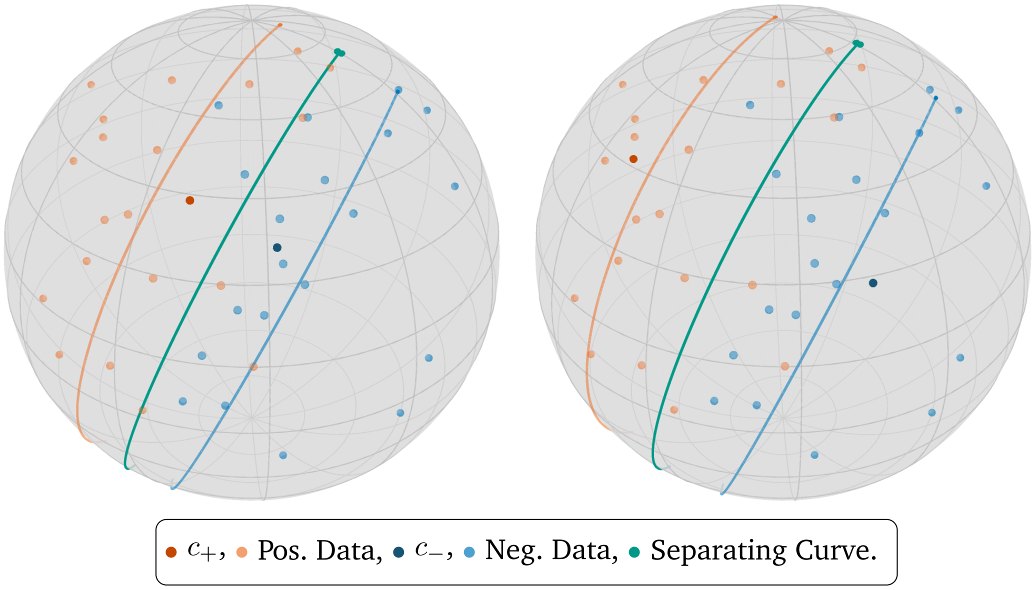

**Fig. 7.16.:** Control Point SVM classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $k = \pi/20$, right; $k = \pi/5$. Misclassification penalty set to $C_{\mathrm{CP-SVM}} = 10$ for both. Margin lines drawn at $f_{\mathrm{CP}}(q \mid c_+, c_-) = \pm k$.

Lastly, we present the CP-SVM model trained on the overlapping dataset for varying desired margin $k$ and misclassification penalty in Figs. 7.16 and 7.17, respectively. The two classifiers in Fig. 7.16 end up generating very similar separating curves, even though $k$ increases fourfold between the left and right frame. The distance between the control point does increase, though. In contrast, we see that when we increase the misclassification penalty from $C_{\mathrm{CP-SVM}} = 1$ to $C_{\mathrm{CP-SVM}} = 20$ in Fig. 7.17 the separating curve tilts and the margins decrease as expected.

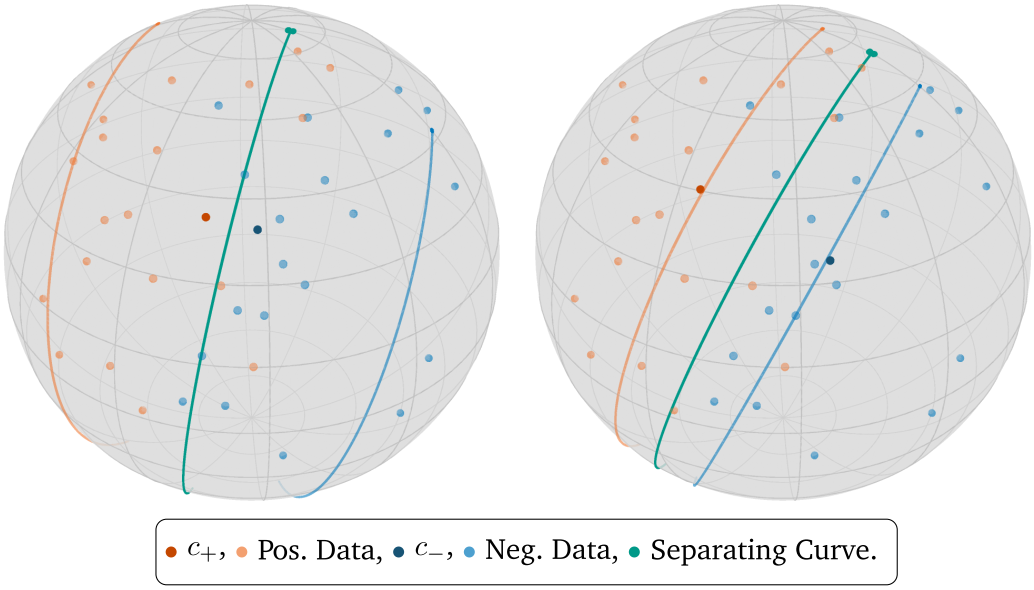

**Fig. 7.17.:** Control Point SVM classifiers on $\mathbb{S}(2)$. Trained on separable dataset. Left: $C_{\text{CP-SVM}} = 1$, right: $C_{\text{CP-SVM}} = 20$. Desired margin set to $k = \pi/16$ for both. Margin lines drawn at $f_{\text{CP}}(q \mid c_+, c_-) = \pm k$.

## 7.3 BCI-IV Experiments

To compare the existing manifold SVM models presented in Chapter 5 with the Distance SVM models presented in Chapter 6 on real manifold valued data we use EEG data from dataset 2a of the Brain Computer Imaging Competition IV (BCI-IV) [39]. The experiments performed by Naeem, Brunner, et al. [31] consist of recording the brain activity of subjects performing four different motor imaging tasks while a cue is shown to them on a screen. That is, the subjects are told to imagine either movement of the left hand (LH), the right hand (RH), both feet (BF), or the tongue (TO). A total of nine subjects participated in the experiments, recording 144 samples for each cue, for each of two different sessions.

The brain activity of each subject was recorded with a set of 22 EEG electrodes, and a sampling frequency of 250Hz. The duration of each cue was $1.25$ seconds. In the same manner as Barachant, Bonnet, et al. [9] the raw EEG data measured in Volts is scaled by $10^6$ to measure in $\mu$V, before passing the data from each sample through a $8 - 35$ Hz 5th order Butterworth band-pass filter [38]. The effect of the band-pass filter on the EEG signal is illustrated in Fig. 7.18, for a single sample ranging over 1.25 seconds of the LH cue for subject 8 in the first session.

To classify which cue was shown to a given subject from the electrode measurements $\mathbf{X}_i \in \mathbb{R}^{T\times 22}$, $i = 1, \ldots, 144$, where $T \in \mathbb{N}$ is the cue duration measured in number of samples, we construct the empirical covariance matrices

$$p_i = \frac{1}{T-1}\mathbf{X}_i^T\mathbf{X}_i \in \mathbb{R}^{22\times 22} \tag{7.5}$$

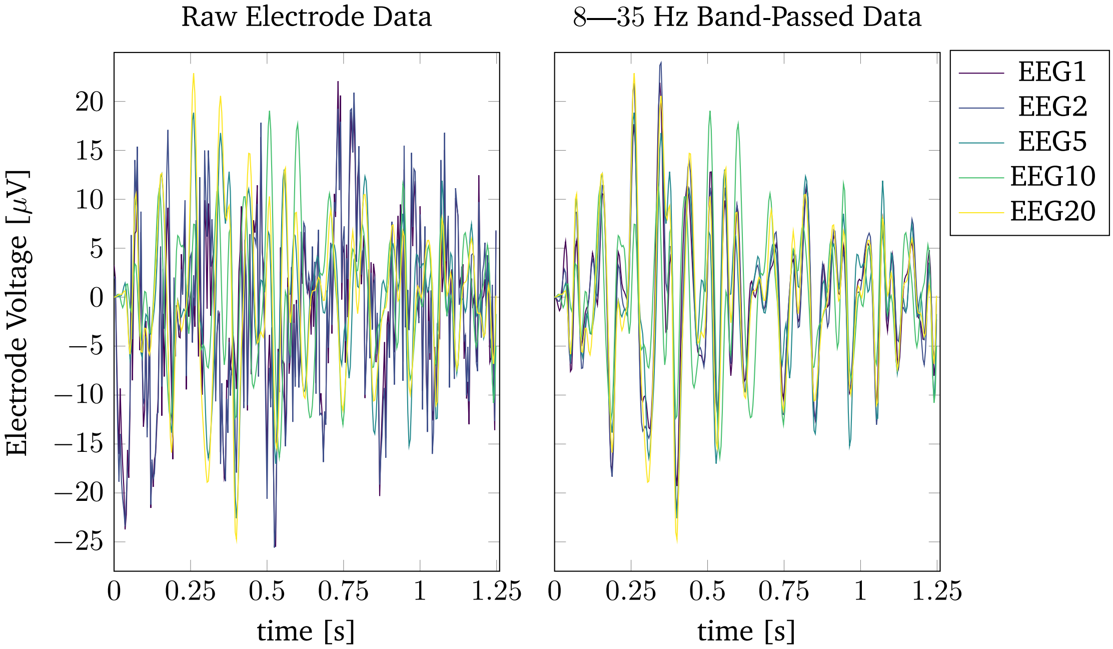

**Fig. 7.18.:** Sample training data for subject 8, LH cue. Comparison of raw data (Left) with data passed through an 8—35 Hz 5'th order Butterworth band-pass filter (Right).

for each cue sample in the dataset after scaling and band-pass filtering. Although the empirical covariance matrices $p_i$ in general are only positive semi-definite, numerically we tested that their eigenvalues are all greater than $10^{-3}$, and we treat them as points on $\mathcal{P}(22)$. The manifold SVM models are all binary classifiers, so we follow the authors of [9] in evaluating the average performance of each model over the six binary classifications

$$\begin{aligned}\{(1: LH \setminus RH),\ (2: LH \setminus BF),\ (3: LH \setminus TO),\\ (4: RH \setminus BF),\ (5: RH \setminus TO),\ (6: BF \setminus TO)\}\end{aligned} \tag{7.6}$$

for each subject. We train the models on the data from the first session and then evaluate the test accuracy on the data from the second session. As a measure of how well separated data set $\mathcal{X} = \{(p_i, y_i)\}_{i=1}^{144} \subset \mathcal{P}(22) \times \{-1, 1\}$ we compute its *relative class separation*

$$\mathrm{RCS}(\mathcal{X}) = \frac{\overline{\mathrm{BCD}}}{\left(\overline{\mathrm{PCD}} + \overline{\mathrm{NCD}}\right)/2}, \tag{7.7}$$

as the ratio between its average between-class distance defined in Eq. (5.10), and the mean of the average positive and negative class distances defined in Eqs. (5.11) and (5.12). The relative class separation of all the $9 \cdot 6 = 54$ different binary classification tasks from the BCI-IV 2a dataset are shown in Table 7.1. As the values are all slightly bigger than $1$ we present the values as $100 \cdot (\mathrm{RCS}(\mathcal{X}) - 1)$, and then we see clear differences across the subjects in the table.

|  | Binary Classification |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| Subj. | 1 | 2 | 3 | 4 | 5 | 6 | Mean |
| 1 | 1.48 | 2.05 | 4.61 | 2.40 | 6.21 | 1.15 | 2.98 |
| 2 | 0.69 | 0.43 | 0.26 | 0.40 | 0.34 | 0.40 | 0.42 |
| 3 | 2.02 | 1.33 | 2.55 | 1.46 | 1.85 | 0.50 | 1.62 |
| 4 | 0.32 | 1.02 | 1.53 | 0.93 | 1.32 | 0.25 | 0.89 |
| 5 | 0.17 | 0.55 | 0.51 | 0.71 | 0.55 | 0.12 | 0.44 |
| 6 | 0.43 | 0.38 | 0.42 | 0.43 | 0.31 | 0.36 | 0.39 |
| 7 | 0.31 | 1.38 | 2.64 | 1.28 | 2.05 | 0.70 | 1.39 |
| 8 | 2.58 | 1.07 | 2.78 | 0.97 | 2.21 | 1.94 | 1.92 |
| 9 | 3.94 | 1.70 | 6.46 | 3.19 | 7.3 | 3.67 | 4.38 |

**Tab. 7.1.:** Relative class separation, Eq. (7.7), of all the training data from the first session of the BCI-IV 2a dataset. Broken down by subject and binary classification between motor imagery tasks. Distances are computed using the Linear-Affine metric on $\mathcal{P}(n)$, and all values are given as $100 \cdot (\mathrm{RCS}(\mathcal{X}) - 1)$ for readability.

In order to choose which hyperparameters to use for the different manifold SVM models we've done a limited hyperparameter search for each model using 5-fold cross validation on all 6 binary classifications for each of the 9 subjects.

The cross validation results for the four DSVM models DSVM, Sp-DSVM, ZCDSVM, and Sp-ZCDSVM when varying the misclassification weight $C_{\mathrm{DSVM}}$ are shown in Fig. 7.19. We see a similar trend in cross validation error across all the four models, where the cross validation accuracy decreases when the misclassification penalty is too low at $C_{\mathrm{DSVM}} = 0.05$, and decreases when the misclassification penalty increases from $0.5$ and above, likely due to overfitting to the training data.

From Fig. 7.19 we also see that all the DSVM models achieve their highest average cross validation accuracy at $C_{\mathrm{DSVM}} = 0.1$. At that value of $C_{\mathrm{DSVM}}$ we see that both the full DSVM models fare better than their sparse counterparts in terms of cross validation accuracy, and furthermore the zero curvature models perform better than the models without that constraint.

The cross validation results for the existing manifold SVM models are shown in Figs. 7.20 and 7.21. Of note is the fact that the MRBF model over $(\mathcal{P}(22), g^{\mathrm{LE}})$ is relatively insensitive to changes in the misclassification penalty parameter $C_{\mathrm{MRBF}}$, but is highly sensitive to the choice of variance scale $\sigma_s$ related to $\sigma$ through Eq. (5.13).

After cross validation, the final classifiers for each subject and model are optimized with the hyperparameters that achieved the highest average cross validation accuracy over all six binary classifications for that subject. Then we test the classifiers on the unseen data from the second session of the BCI-IV 2a dataset.

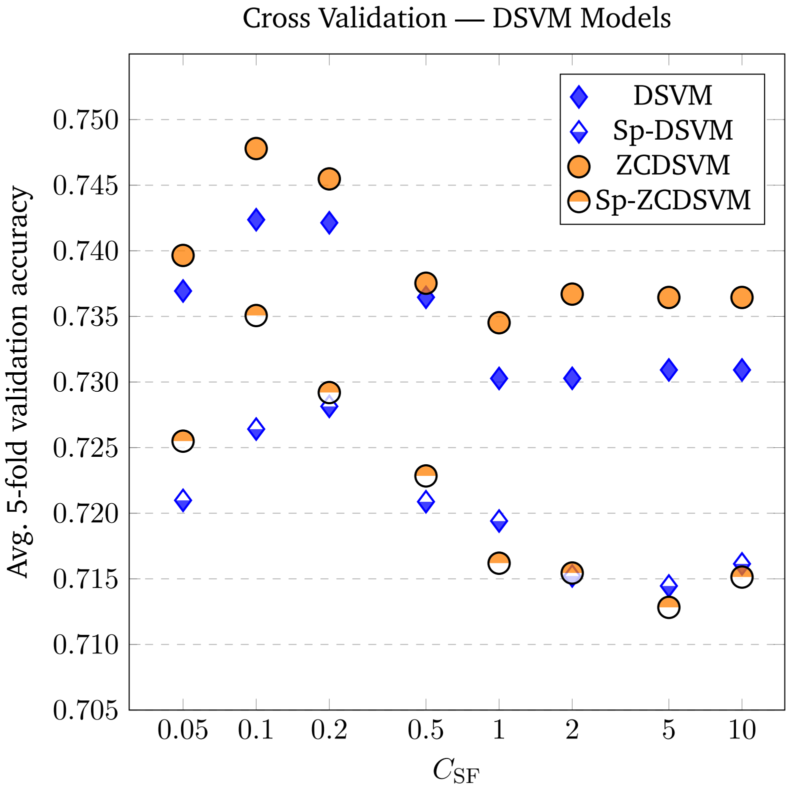

**Fig. 7.19.:** Cross validation results for DSVM models.

The test results for each model are shown in Tables 7.2 and 7.3, and they display a significant difference on a per-subject basis, consistent with the trends in relative class separation shown in Table 7.1. The test accuracy is particularly low for subject 2, where e.g. the DSVM model only achieves a test accuracy of $50\%$, equivalent to random chance. The hyperparameters used for each model and subject are given in Appendix A.

In Table 7.2 comparing the test results for the four different DSVM models, we see that both the Sp-DSVM and Sp-ZCDSVM models perform comparably well with average test accuracies of $72.15\%$ and $71.95\%$, respectively. In addition, they both perform better than their full-support counterparts with average test accuracies of $67.98\%$ and $69.71\%$ respectively. The existing MRBF and TS-SVM and models in comparison achieve an average test accuracy of $70.37\%$ and $72.54\%$, whilst the CP-SVM model achieves the highest test accuracy at $72.65\%$.

Of note is the fact that the support fraction of the training data used by the sparse Distance SVM models lower than for the existing M-RBF and TS-SVM models. Between the existing manifold SVM models the M-RBF model is the worst performing in terms of test accuracy, and is also the only model using the Log-Euclidean metric on $\mathcal{P}(n)$, which induces a flat manifold structure. It could be that it is harder to separate the different classes using that metric, but then again the choice of the Log-Euclidean metric was required to guarantee positive definiteness of the manifold RBF kernel.

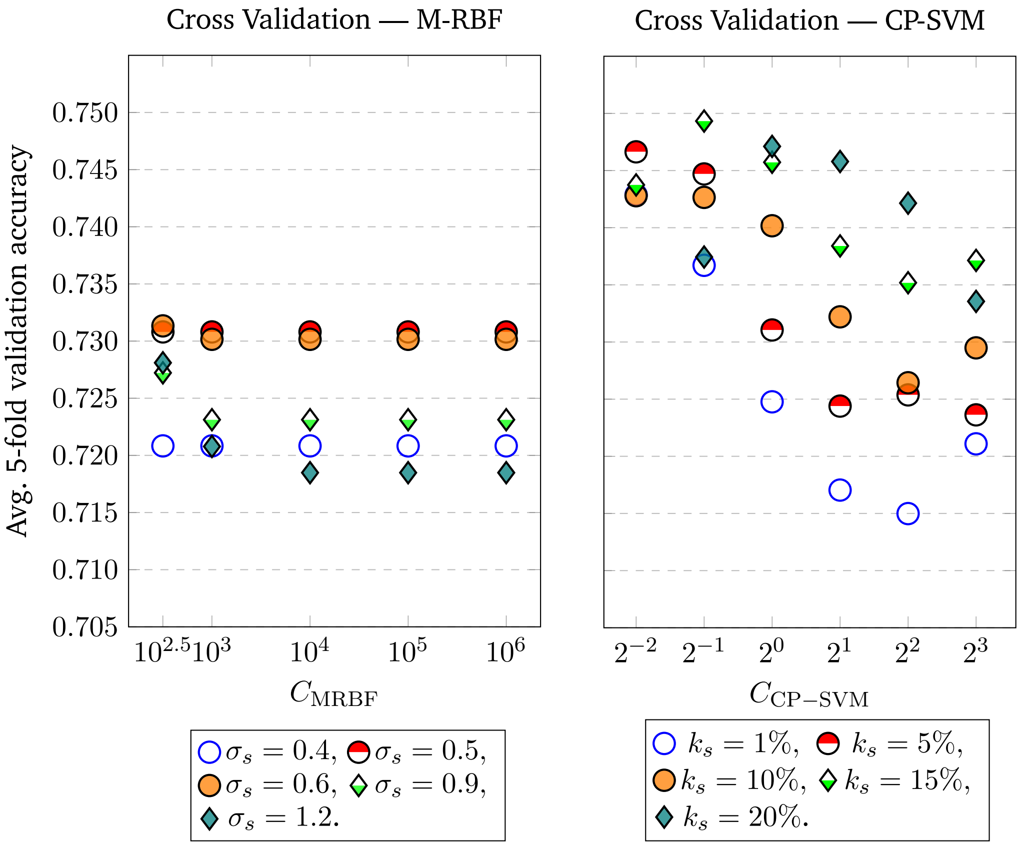

**Fig. 7.20.:** Cross Validation results for the Manifold RBF and Control point SVM models on the BCI-IV training data. Each data point is the average 5-fold validation accuracy, averaged over all binary classifications and subjects ($n = 6 \cdot 9 = 54$).

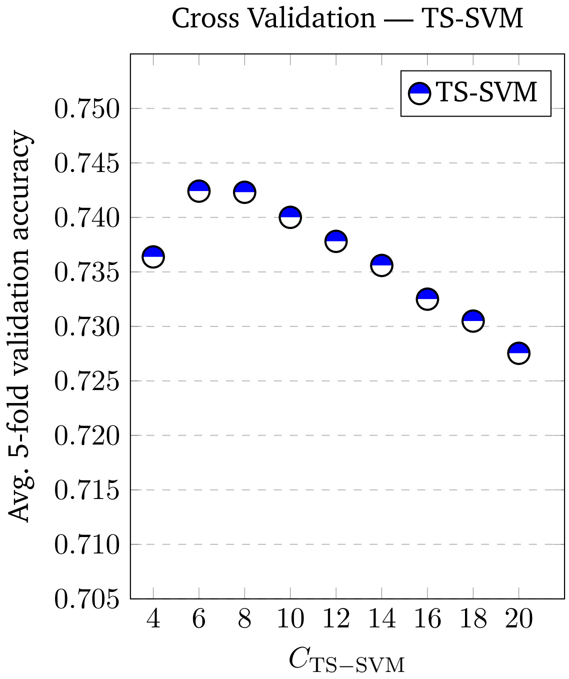

**Fig. 7.21.:** Cross Validation results for the TS-SVM model on the BCI-IV training data. Each data point is the average 5-fold validation accuracy, averaged over all binary classifications and subjects ($n = 6 \cdot 9 = 54$).

| Subj. | DSVM | Sp-DSVM | ZCDSVM | Sp-ZCDSVM |
|---|---|---|---|---|
| 1 | 73.96 (±7.95) | **79.17 (±7.49)** | 74.65 (±9.41) | 77.08 (±10.08) |
| 2 | 50.0 (±2.78) | 50.69 (±0.36) | **64.93 (±4.04)** | 56.94 (±3.53) |
| 3 | **82.29 (±8.33)** | 73.26 (±6.21) | 80.21 (±6.83) | 76.39 (±4.92) |
| 4 | 61.81 (±8.70) | **81.60 (±11.10)** | 63.89 (±8.47) | 78.47 (±8.18) |
| 5 | 61.81 (±8.81) | **70.14 (±5.319)** | 63.89 (±8.022) | **70.14 (±8.107)** |
| 6 | 68.06 (±2.23) | 62.15 (±5.51) | **70.14 (±2.97)** | 61.11 (±3.50) |
| 7 | 60.42 (±9.09) | **84.38 (±10.89)** | 55.208 (±1.20) | 77.083 (±1.09) |
| 8 | 82.64 (±4.87) | 72.57 (±5.82) | **84.72 (±7.20)** | 74.31 (±5.07) |
| 9 | 70.83 (±6.72) | 75.35 (±6.95) | 69.79 (±8.21) | **76.04 (±6.29)** |
| Mean test acc.: | 67.98 (±11.24) | **72.15 (±11.17)** | 69.71 (±10.05) | 71.95 (±9.26) |
| Support frac.: | 100 | 60.3 | 100 | 63.5 |

Tab. 7.2.: Average test accuracy and standard deviation over all six pairs of mental tasks for the BCI-IV 2a dataset, in percentages. Comparing results for the Distance SVM models. The ”support fraction” of each model is the fraction of the training data used by the model classifier, averaged over all trials.

| Subj. | MRBF | CP-SVM | TS-SVM |
|---|---|---|---|
| 1 | **75.52 (±6.87)** | 75.0 (±9.95) | 74.88 (±10.26) |
| 2 | 55.44 (±6.16) | 58.80 (±5.06) | **60.42 (±2.20)** |
| 3 | 77.89 (±6.33) | **78.24 (±4.51)** | 78.00 (±5.15) |
| 4 | 71.76 (±10.69) | **77.20 (±8.44)** | 76.97 (±8.07) |
| 5 | 59.95 (±6.374) | **61.57 (±6.449)** | 60.30 (±8.453) |
| 6 | 61.23 (±5.39) | 68.87 (±2.72) | **69.21 (±4.23)** |
| 7 | **76.041 (±0.928)** | **76.041 (±1.080)** | 75.694 (±1.383) |
| 8 | **77.55 (±5.84)** | 77.43 (±6.74) | **77.55 (±6.25)** |
| 9 | 78.24 (±6.64) | **80.67 (±7.18)** | 79.86 (±8.00) |
| Mean test acc.: | 70.37 (±10.84) | **72.65 (±10.00)** | 72.54 (±10.31) |
| Support frac.: | 94.7 | N/A | 81.5 |

Tab. 7.3.: Average test accuracy and standard deviation over all six pairs of mental tasks for the BCI-IV 2a dataset, in percentages. Comparing results for existing Manifold SVM models. The ”support fraction” of each model is the fraction of the training data used by the model classifier, averaged over all trials.

# Conclusion 8

In this thesis we considered binary classification models on Riemannian manifolds. The novel Distance SVM and Zero Curvature Distance SVM models were introduced inChapter 6, and we showed that the latter model produces linear separators in vector spaces. By numerical experiments in Fig. 7.3 we showed that even though we cannot in general expect the ZCDSVM separators to be the maximum margin separators produced by the Euclidean SVM model, we do in some cases recover the maximum margin separators with the ZCDSVM models.

Regarding sparsity in terms of set of support points for the DSVM and ZCDSVM models, the optimization procedures presented in Theorem 6.1 and Theorem 6.2 do not produce sparse support point weights directly. However, we can achieve DSVM and ZCDSVM classifiers which use sparse subsets of the training points as support points with the two-step optimization heuristic presented in Section 6.4. This sparsity heuristic appears to work well, and seemingly captures relevant training data for producing good separating curves, as illustrated in the figures comparing the full and sparse DSVM models in Sections 7.1 and 7.2.

Compared to the TS-SVM model of Section 5.1 the DSVM models have the advantage that they to not require a choice of reference point. Instead, they're inherent to the manifold and not confined to a single tangent vector space. The MRBF model of Section 5.2 also does not require the choice of a reference point, but instead requires that the manifold be isometrically embeddable in a Hilbert space. This limits the classes of manifolds on which the MRBF kernel is guaranteed to produce positive definite kernel matrices, but empirically we've shown that it can still work on non isometrically embeddable manifolds like $\mathbb{S}(2)$, as illustrated in Section 7.2.

The CP-SVM model detailed in Section 5.3 is conceptually simpler, taking a different direction towards classifying manifold valued data by which representative control point is closest. The CP-SVM model seems to be resistant to overfitting to training data, producing simpler separating curves than e.g. DSVM on the two $\mathbb{S}(2)$ datasets shown in Section 7.2. However, the optimization procedure to find optimal control points for the CP-SVM model is not convex over $\mathcal{M} \times \mathcal{M}$, and we can therefore generally only hope to find a local minimizer of the CP-SVM objective in Eq. (5.17). In contrast, the TS-SVM, MRBF and DSVM models all rely on solving a convex optimization problem in order to compute their classifiers, and therefore find globally optimal solutions.

In Section 7.3 we compared the existing manifold SVM models to the DSVM models on real world data on $\mathcal{P}(22)$. We see from the test results in Tables 7.2 and 7.3 that the

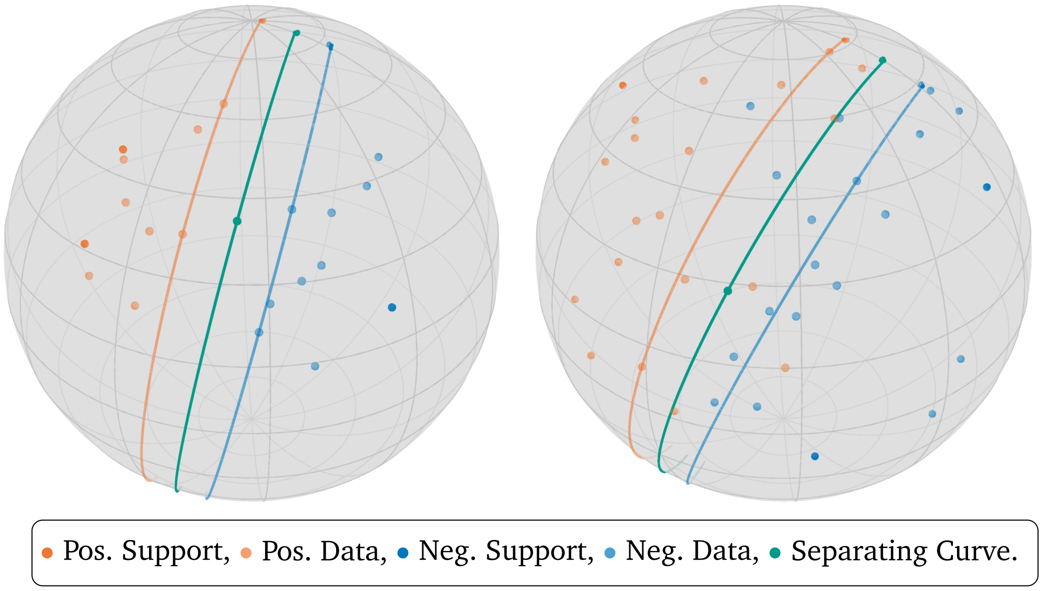

**Fig. 8.1.:** Example of L1-norm regularized ZCDSVM classifiers on $\mathbb{S}(2)$. Misclassification penalty set to $C_{\text{MRBF}} = 10^2$ for both.

sparse DSVM models are competitive with the best performing existing models, CP-SVM and TS-SVM. The Sp-DSVM model achieved the highest average test accuracy among the DSVM models of $72.15\,(\pm 11.17)\%$, within half a percentage point of the CP-SVM model which achieved the highest average test accuracy of $72.65\,(\pm 10.00)\%$.

## 8.1 Future Work

It would be interesting to look into other ways of achieving sparsity for the DSVM and ZCDSVM models. The optimization problems (6.11) and (6.19) do not result in a naturally sparse set of support points, but one natural question is how the support point weights would change if we instead minimize the L1-norm of $\beta$, as the L1-norm usually promotes sparsity.

We've implemented the L1-regularized ZCDSVM model numerically, and illustrated the resulting classifiers on the two $\mathbb{S}(2)$ datasets from Section 7.2 in Fig. 8.1. As expected the resulting $\hat{\beta}$ is sparse, and interestingly, the support points picked out by the optimization procedure are the training points the farthest away from the separating curve.

Another interesting direction for future work would be to investigate how one could generalize the idea of support vector machines for regression on Euclidean spaces, as presented in [16, Chap. 12.3.6]. One switches context from a hinge loss in classification to a hinge loss on the absolute error in the regression, but in Euclidean spaces the framework is largely similar, and one ends up solving quadratic minimization problems of similar to Prob. (3.18).

# A Hyperparameter Values for BCI-IV 2a Dataset

The hyperparamter values for the DSVM models are given in Table A.1, and the values for the existing manifold SVM models are given in Table A.2.

| | Model | | | |
|---|---|---|---|---|
| Subj. | DSVM | Sp-DSVM | ZCDSVM | Sp-ZCDSVM |
| 1 | 0.1 | 0.2 | 0.1 | 0.1 |
| 2 | 0.2 | 0.1 | 0.1 | 0.1 |
| 3 | 0.2 | 0.5 | 0.2 | 0.1 |
| 4 | 0.5 | 0.2 | 1.0 | 0.5 |
| 5 | 0.5 | 5.0 | 0.2 | 0.1 |
| 6 | 0.1 | 0.1 | 0.1 | 0.2 |
| 7 | 0.2 | 0.1 | 0.05 | 0.1 |
| 8 | 0.1 | 1.0 | 0.1 | 0.2 |
| 9 | 0.05 | 0.05 | 0.05 | 0.05 |

Tab. A.1.: Hyperparameter $C_{\mathrm{DSVM}}$ used for a specific model on a subject when training the DSVM models on the first session of the BCI-IV 2a dataset.

| | Models | | |
|---|---|---|---|
| Subj. | MRBF $(C, \sigma_s)$ | TS-SVM $(C)$ | CP-SVM $(C, k_s)$ |
| 1 | $(10^{2.5}, 0.5)$ | $8$ | $(2^{-1}, 0.1)$ |
| 2 | $(10^{2.5}, 0.5)$ | $6$ | $(2^{-3}, 0.005)$ |
| 3 | $(10^{2.5}, 0.6)$ | $6$ | $(2^{-2}, 0.05)$ |
| 4 | $(10^{2.5}, 0.9)$ | $14$ | $(2^{-2}, 0.01)$ |
| 5 | $(10^{3}, 0.9)$ | $12$ | $(2^{-1}, 0.005)$ |
| 6 | $(10^{3}, 0.9)$ | $6$ | $(2^{-1}, 0.05)$ |
| 7 | $(10^{2.5}, 0.5)$ | $4$ | $(2^{-3}, 0.005)$ |
| 8 | $(10^{2.5}, 0.5)$ | $8$ | $(2^{-1}, 0.005)$ |
| 9 | $(10^{2.5}, 0.4)$ | $4$ | $(2^{-3}, 0.01)$ |

Tab. A.2.: Hyperparameters used for training MRBF, TS-SVM, and CP-SVM models on a subject when training on the BCI-IV 2a dataset.

# Bibliography

[1] Vladimir Vapnik, Steven E Golowich, and Alex Smola. „Support vector method for function approximation, regression estimation, and signal processing“. In: *Advances in neural information processing systems* (1996), pp. 281–287 (cit. on p. 2).

[2] Corinna Cortes and Vladimir Vapnik. „Support vector networks“. In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: 10.1023/a:1022627411411 (cit. on p. 2).

[3] E. Osuna, R. Freund, and F. Girosit. „Training support vector machines: an application to face detection“. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc. DOI: 10.1109/cvpr.1997.609310 (cit. on p. 2).

[4] Manfredo Perdigao do Carmo. *Riemannian geometry*. Vol. 6. Springer, 1992. ISBN: 978-0817634902 (cit. on pp. 2, 22, 26, 28–33).

[5] P.T. Fletcher, Conglin Lu, and S. Joshi. „Statistics of shape via principal geodesic analysis on Lie groups“. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 1. 2003. DOI: 10.1109/CVPR.2003.1211342 (cit. on p. 2).

[6] Martin Styner, Jeffrey A. Lieberman, Dimitrios Pantazis, and Guido Gerig. „Boundary and medial shape analysis of the hippocampus in schizophrenia“. In: *Medical Image Analysis* 8.3 (Sept. 2004), pp. 197–203. DOI: 10.1016/j.media.2004.06.004 (cit. on p. 2).

[7] Oncel Tuzel, Fatih Porikli, and Peter Meer. „Pedestrian Detection via Classification on Riemannian Manifolds“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1713–1727. DOI: 10.1109/TPAMI.2008.75 (cit. on p. 2).

[8] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. „Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.12 (2015), pp. 2464–2477. DOI: 10.1109/TPAMI.2015.2414422 (cit. on p. 2).

[9] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. „Classification of covariance matrices using a Riemannian-based kernel for BCI applications“. In: *Neurocomputing* 112 (July 2013), pp. 172–178. DOI: 10.1016/j.neucom.2012.12.039 (cit. on pp. 3, 42, 43, 75, 76).

[10] O. Tuzel, F. Porikli, and P. Meer. „Pedestrian Detection via Classification on Riemannian Manifolds“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (Oct. 2008), pp. 1713–1727. DOI: 10.1109/tpami.2008.75 (cit. on pp. 3, 43).

[11] Sadeep Jayasumana, Richard Hartley, and Mathieu Salzmann. „Kernels on Riemannian Manifolds“. In: *Riemannian Computing in Computer Vision*. Ed. by Pavan K. Turaga and Anuj Srivastava. Cham: Springer International Publishing, 2016, pp. 45–67. ISBN: 978-3-319-22957-7. DOI: 10.1007/978-3-319-22957-7_3 (cit. on pp. 3, 44, 45).

[12] Yixiao Yun, Irene Yu-Hua Gu, and Hamid Aghajan. „Riemannian manifold-based support vector machine for human activity classification in images“. In: *2013 IEEE International Conference on Image Processing*. IEEE. 2013, pp. 3466–3469. DOI: 10.1109/ICIP.2013.6738715 (cit. on p. 3).

[13] Suman K Sen, Mark Foskey, James S Marron, and Martin A Styner. „Support vector machine for data on manifolds: An application to image analysis“. In: *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2008, pp. 1195–1198. DOI: 10.1109/ISBI.2008.4541216 (cit. on pp. 3, 46, 48).

[14] Jorge Nocedal and Stephen Wright. *Numerical optimization*. 2nd Edition. Springer Science & Business Media, 2006. ISBN: 978-0-387-30303-1. DOI: https://doi.org/10.1007/978-0-387-40065-5 (cit. on pp. 5–7, 9–11, 24, 37, 38).

[15] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. 1st Edition. Cambridge university press, 2004. ISBN: 978-0-521-83378-3 (cit. on pp. 7–10).

[16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. DOI: 10.1007/978-0-387-84858-7 (cit. on pp. 12–14, 16, 82).

[17] Christopher J.C. Burges. „A Tutorial on Support Vector Machines for Pattern Recognition“. In: *Data Mining and Knowledge Discovery* 2.2 (1998), pp. 121–167. DOI: 10.1023/a:1009715923555 (cit. on pp. 12, 14, 16).

[18] Roger A Horn and Charles R Johnson. *Matrix Analysis*. en. 2nd ed. Cambridge, England: Cambridge University Press, Oct. 2013. ISBN: 978-0521548236 (cit. on pp. 16, 19).

[19] Andreas Wächter and Lorenz T. Biegler. „On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming“. In: *Mathematical Programming* 106.1 (Apr. 2005), pp. 25–57. DOI: 10.1007/s10107-004-0559-y (cit. on pp. 16, 62).

[20] Brendan O'Donoghue. „Operator Splitting for a Homogeneous Embedding of the Linear Complementarity Problem“. In: *SIAM Journal on Optimization* 31 (3 Aug. 2021), pp. 1999–2023. DOI: https://doi.org/10.1137/20M1366307 (cit. on p. 16).

[21] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Springer New York, 1984. DOI: 10.1007/978-1-4612-1128-0 (cit. on pp. 17–20, 45).

[22] N. Aronszajn. „Theory of reproducing kernels“. In: *Transactions of the American Mathematical Society* 68.3 (1950), pp. 337–404. DOI: 10.1090/s0002-9947-1950-0051437-7 (cit. on p. 20).

[23] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. ISBN: 9780691132983 (cit. on pp. 21–31, 37–40).

[24] Adam Bowers and Nigel J Kalton. *An introductory course in functional analysis*. Springer, 2014. ISBN: 978-1-4939-1944-4. DOI: https://doi.org/10.1007/978-1-4939-1945-1 (cit. on p. 22).

[25] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press. Apr. 2022, to appear. (Cit. on pp. 23, 24, 28).

[26] John M Lee. *Introduction to Riemannian manifolds*. Springer, 2018. ISBN: 978-3-319-91754-2. DOI: https://doi.org/10.1007/978-3-319-91755-9 (cit. on pp. 33–35).

[27] Bijan Afsari, Roberto Tron, and René Vidal. „On the convergence of gradient descent for finding the Riemannian center of mass“. In: *SIAM Journal on Control and Optimization* 51.3 (2013), pp. 2230–2260. DOI: https://doi.org/10.1137/12086282X (cit. on pp. 35, 36).

[28] Miroslav Bacak. *Convex analysis and optimization in Hadamard spaces*. en. De Gruyter Series in Nonlinear Analysis & Applications. Berlin, Germany: De Gruyter, Sept. 2014. ISBN: 978-3110361032 (cit. on pp. 35, 36, 41).

[29] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. „Geometric means in a novel vector space structure on symmetric positive-definite matrices“. In: *SIAM journal on matrix analysis and applications* 29.1 (2007), pp. 328–347. DOI: https://doi.org/10.1137/050637996 (cit. on pp. 40, 41).

[30] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. „A Riemannian Framework for Tensor Computing“. In: *International Journal of Computer Vision* 66.1 (Jan. 2006), pp. 41–66. DOI: 10.1007/s11263-005-3222-z (cit. on pp. 40, 41).

[31] M Naeem, C Brunner, R Leeb, B Graimann, and G Pfurtscheller. „Seperability of four-class motor imagery data using independent components analysis“. In: *Journal of Neural Engineering* 3.3 (June 2006), pp. 208–216. DOI: 10.1088/1741-2560/3/3/003 (cit. on pp. 43, 75).

[32] Oleksandr Zadorozhnyi, Gunthard Benecke, Stephan Mandt, Tobias Scheffer, and Marius Kloft. „Huber-Norm Regularization for Linear Prediction Models“. In: *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 2016, pp. 714–730. DOI: 10.1007/978-3-319-46128-1_45 (cit. on p. 47).

[33] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Ltd, May 2000. DOI: 10.1002/9781118723203 (cit. on p. 59).

[34] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. „Julia: A Fresh Approach to Numerical Computing“. In: *SIAM Review* 59.1 (Jan. 2017), pp. 65–98. DOI: 10.1137/141000671 (cit. on p. 62).

[35] Seth D. Axen, Mateusz Baran, Ronny Bergmann, and Krzysztof Rzecki. „Manifolds.jl: An Extensible Julia Framework for Data Analysis on Manifolds“. In: (2021). arXiv: 2106.08777 (cit. on p. 62).

[36] Ronny Bergmann. „Manopt.jl: Optimization on Manifolds in Julia“. In: *Journal of Open Source Software* 7.70 (2022), p. 3866. DOI: 10.21105/joss.03866 (cit. on p. 62).

[37] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration*. 2nd ed. Springer Series in Computational Mathematics. Berlin, Germany: Springer, Feb. 2006 (cit. on p. 63).

[38] Stephen Butterworth et al. „On the theory of filter amplifiers“. In: *Wireless Engineer* 7.6 (1930), pp. 536–541 (cit. on p. 75).

## Websites

[39] Benjamin Blankertz. *BCI Competition IV*. 2022. URL: https://bbci.de/competition/iv/ (visited on May 9, 2022) (cit. on pp. 4, 43, 62, 75).

[40] User:ZackWeinberg. *SVM Separating Hyperplanes*. 2012. URL: https://commons.wikimedia.org/w/index.php?curid=22877598 (visited on May 10, 2022) (cit. on p. 13).

[41] 2022. URL: https://juliamanifolds.github.io/ManifoldDiffEq.jl/stable/ (visited on June 20, 2022) (cit. on p. 63).

## Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at http://cleanthesis.der-ric.de/.

NTNU

Norwegian University of
Science and Technology