Classification of Data on Manifold

by

MAULIK PANDYA
201111007

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

June, 2013

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgement has been made in the text to all the reference material used.

Maulik Pandya

Certificate

This is to certify that the thesis work entitled CLASSIFICATION OF DATA ON MANIFOLD has been carried out by MAULIK PANDYA for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* (DA-IICT) under my supervision.

Prof. Aditya Tatu
Thesis Supervisor

Acknowledgement

First and foremost, a special feeling of gratitude to my loving mother, Mrs. Rakshaben Pandya, who has been a towering source of strength to me. I would like to express my love to my family members for their words of encouragement throughout the life. My efforts may appear to be just a straw in the precious collection of research, but I am satisfied with my efforts for I feel lord Krishna's effulgence illuminating my whole being.

I would like to express deepest appreciation to Prof. Aditya Tatu with his guidance and persistence I am able to complete this dissertation. I like to express my gratitude to him for the continuous support during my M. Tech study and research, for his immense knowledge, enthusiasm and motivation. His advice helped me in all the time of research work and writing of the thesis. Also for his assistance in offering counsel and offering suitable suggestions while reviewing the thesis, without whom I might not have able to undertake this stupendous task and complete it. I could not have imagined having a better advisor and mentor for my M. Tech study.

Along with my advisor, I would like to thank thesis subject expert Prof. Mehul Raval, for his insightful comments, encouragement, and thought provoking questions. I am grateful to Dr. H. B. Shah for enlightening me the first glance of research. Also I would like to thank DA-IICT for providing the required resources.

My heart goes to my friends and relatives who have been instrumental in my endeavours. I thank my fellow Parth, Pratik, Nirav and Deep for the sleepless nights we were working together before deadlines, the stimulating discussions, and for all the fun we have had in the last two years.

Contents

Abstract

Data classification is one of the most challenging task in the field of pattern recognition and computer vision. Sometimes the values of signal or data are naturally described as points on a *manifold*. Such data arise from medial representations (m-reps) in medical images, Diffusion Tensor-MRI (DT-MRI), diffeomorphisms, etc. Since this may not be a *vector space*, one needs to be careful while choosing classification methods for such data. SVM is one of the popular methods which heavily relies on the fact that the underlying space is a vector space. In this thesis, we adapt SVM to classify data on manifold. We project data on the *tangent space* of a given manifold, which is a vector space and use SVM for classification on this vector space. We try to explore Euclidean SVM on multiple tangent planes so that we can generalize the idea of SVM for data on manifolds. We give an algorithm to find the point on whose tangent space the classification margin is maximized. We show results of classification with various manifolds like S^2, $SO(3)$ and $PD(2)$. As computer vision application, we classify textures using SVM on $PD(n)$ manifold.

List of Principal Symbols and Acronyms

$\frac{D}{dt}$	Covariant derivative
$GL(n, \mathbb{R})$	General linear group of $n \times n$ invertible matrices
$\mathbb{R}^m$	Real coordinate space of m-dimension
M	Manifold
$SO(n)$	Set of rotation matrices of size $n \times n$
$PD(n)$	Set of Positive definite matrices of size $n \times n$
T_pM	Tangent space of the manifold at point p
Γ	Christoffel symbol
γ	Curve on manifold
$\triangle$	Separation margin
$\parallel w \parallel$	L^2 norm of a vector w
L	Lagrangian
P^*	Value of a primal problem
D^*	Value of a dual problem
$k(\cdot, \cdot)$	Kernel function

DT-MRI	Diffusion Tensor Magnetic Resonance Image
DWD	Distance Weighted Discrimination
HDLSS	High Dimensional Low Sample Space
ITanSVM	Iterative Tangent plane SVM
KKT	Karush-Kuhn-Tucker
m-rep	Medial Representation
RBF	Radial Basis Function
SVM	Support Vector Machines

List of Tables

List of Figures

CHAPTER 1
Introduction

1.1 Motivation

The work done in this dissertation is inspired by recent advances in manifold learning in the field of pattern recognition. The need for manifold learning is to understand the characteristics of data that are naturally lying on some curved surfaces (i.e., manifold) and to use the geometric information of manifold. Textures, shapes, orientations and many other types data needed to be compared and analysed, and manifold learning can be an approach for such problems. Here, we try to classify data on a manifold using geometrical concepts.

Classification [1] is the process to assign a predefined category to an object. We can also say that it is a process of grouping similar objects together. We do classification from given observation so that we can understand the characteristics of each groups. Like from a class of student, we can categorise them based on some of their characteristics, i.e., height and weight. So, in general, its a method to describe the behaviour of an object.

Similarly, in the field of computer vision and pattern recognition, we do clustering of the data based on the selected features and then at the time of classification, we try to model a function which separates those clusters. There are several robust algorithms available for classification purpose if the data belong to Euclidean space. Support Vector Machines (SVM) [2] is one of the powerful algorithm for data classification. It works on the idea of separating classes with higher margin. SVM heavily relies on the fact that underlying space is a *vector space*.

In many cases, data belongs natural to n-dimensional manifold (example: DT-MRI, medial representations, set of orientation matrices $O(n)$) [3, 4, 5]. Manifold can be seen as a curved surface in high-dimensional space, which locally looks like an Euclidean space, however, globally turns out to be a *curved surface*.

Manifold valued image is a image $I : \Omega \to M$ where $\Omega \subset \mathbb{R}^2$ and M is a manifold space. Diffusion tensor magnetic resonance imaging (DT-MRI) is a typ-

ical example of manifold valued images. Manifold learning is useful in medical images (i.e., DT-MRI, m-rep). In face recognition to compute Manifold Manifold Distance (MMD). Also it is useful for robot navigation and Speech Processing for dimensionality reduction.

Gauss used an abstract spaces as mathematical objects. His *theorema egregium* gives a method for computing the *curvature* of a surface without considering the ambient space in which the surface lies as the curvature of the surface is an intrinsic property of the surface. Such a surface would be called a *manifold*. Manifold theory has come to focus due to these intrinsic properties while ignoring the extrinsic properties of the ambient space.

An object should be independent of their description in a particular coordinate system. If a n-dimension manifold is embedded in $\mathbb{R}^m$ then the description (i.e., derivative, curvature) on manifold should be independent of dimension-m.

As a manifold does not form a vector space, the usual SVM algorithm will not work for such kind of data. In this work, we want to try and extend the idea of SVM if data belongs to some curved manifold.

As our approach relies on the intrinsic properties of the manifold, we are working on a *tangent space* (which is of dimension-n) at each point of the manifold. We project manifold data on the tangent space details given in (Appendix B.2), which is a vector space by definition on which SVM classification is done using the natural *Riemannian inner product* of the manifold. Since the classification margin will depend on the point of the manifold, whose tangent space will be used for SVM, we also propose an iterative algorithm to find a point whose tangent space yields the maximum separating margin.

1.2 Related Work

Principle geodesic analysis, PCA for non-linear surfaces like manifolds has been developed in [3]. Also [3, 4, 5, 6] describe the computation of geodesic mean (mean of data cluster on manifold), geodesic curve (shortest path connecting two points) and distance.

To the best of our knowledge, little work has been done to adapt SVM to manifold data [7, 8]. The method in it successfully classify data on manifold. It uses the concept of control points representing the data of different classes. It finds a decision model which is given by the function of geodesic distance from a point to control points. It is needed to choose a pair of control points correctly. There may not be a unique pair of control points for the correct classification. The method

takes geodesic mean of clusters as a initial pair of control points and assuming data belongs to some convex neighbourhood, using gradient descent method it tries to find out the pair which maximizes some objective function.

The author has also described an Iterative Tangent plane SVM (ITanSVM) algorithm to generalise SVM on the manifold [8]. A common approach is to project the data on the tangent space of geodesic mean [3], however, it may not always give good decision function [8]. Hence the method tries to find a point, which is a geodesic mean of new computed pair of control points. Here by projecting data on the tangent space of point computed by ITanSVM method gives better results than on the tangent space of geodesic mean.

The efficient mapping from data on manifold (i.e., $SO(3)$ and $PD(n)$) to its tangent space and vice-versa has been done using the idea of Logarithm map and Exponential map respectively [9]. In the field of texture classification, [10] has tried to classify Brodatz textures using covariance matrices based on the selected features from images. Covariance matrices of size $n \times n$ are the set of symmetric positive definite matrices $PD(n)$, subgroup of a Lie group. In the next section, we give brief overview of contributions done in this dissertation.

1.3 Contributions

In this dissertation, to generalize the classification algorithm on manifold, we compare the data classification results of separating margin on different tangent spaces of the manifold. we also present an algorithm to compute a point on whose tangent space, we can classify data with higher margin. The method in [3] uses over all geodesic mean as a *base point*, on whose tangent space they project the data. Our proposed algorithm finds a local margin maximizer point using a gradient descent method.

1.4 Thesis Overview

The remainder of this dissertation is organized into chapters as follows:

Chapter 2 gives an brief idea of the required mathematics in this dissertation. This includes topics on manifold and Lie group. We give brief introduction on computation of geodesic path, geodesic mean (or Fréchet mean) of clusters, parallel transport of a vector from a tangent space of one point to another, Exponential map and Logarithm map for mapping of data from manifold to tangent space, and vice-versa, respectively.

Chapter 3 presents the idea of Euclidean SVM classifier which finds a hyperplane with higher separating margin. It includes mathematical background to understand working principle of SVM and presents simple example on linear classification and nonlinear classification using RBF kernel function.

Chapter 4 presents implementation algorithm of Exponential map, Logarithm map and geodesic mean for various manifolds (i.e., S^2,$SO(3)$ and $PD(n)$) used in this dissertation. It introduces proposed method with required mathematical derivations. A discussion on experiments carried out in this work is also included.

Chapter 5 concludes with a discussion of the contributions in this dissertation and possible future work.

Appendix A gives brief introduction on topology.

Appendix B throws more light on the charts, tangent space and covariant derivative on the manifold.

As we want to classify data on manifold, we need to be familiar with geometrical aspects of manifold. We give required background about differential geometry and and SVM in chapter 2 and 3, readers familiar with basics may skip these two chapters.

CHAPTER 2
Differential geometry

2.1 Introduction

In this chapter, we give some basic mathematical concepts to understand differential geometry. Differential geometry is a mathematical idea to study geometric properties of curves and surfaces or in other words differential manifolds using calculus. In this thesis, the data we consider for classification belongs to some curved manifold, for example surface of a sphere, set of rotation transformation $SO(n)$, set of positive definite and symmetric matrices $PD(n)$. We begin with brief introduction to manifold and Lie Group.

2.2 Manifold

The values of signals and data are sometimes naturally described as points in manifolds. Diffusion Tensor Magnetic Resonance Images (DT-MRIs) and vector field images are the typical example of manifold valued images.

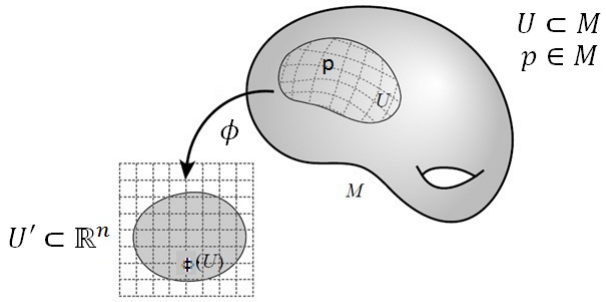


Figure 2.1: Local mapping on manifold.

A manifold (M) [11, 12, 13] is a mathematical concept which extends the idea of Euclidean spaces, in the sense that manifolds locally look like Euclidean spaces

or we can say that a neighbourhood around each point can be nicely mapped to an n-dimensional Euclidean space.

A manifold M of dimension n is a topological space with following property
(1) M is Hausdorff (Appendix A.4)
(2) M is locally Euclidean of n-dimension
(3) M has n countable basis (Appendix A.1)

Locally Euclidean means for each point $p \in M$, $\exists$ a neighbourhood U (subset of manifold) which is *homeomorphic* (details given in Appendix A.3) to U' in $\mathbb{R}^n$.

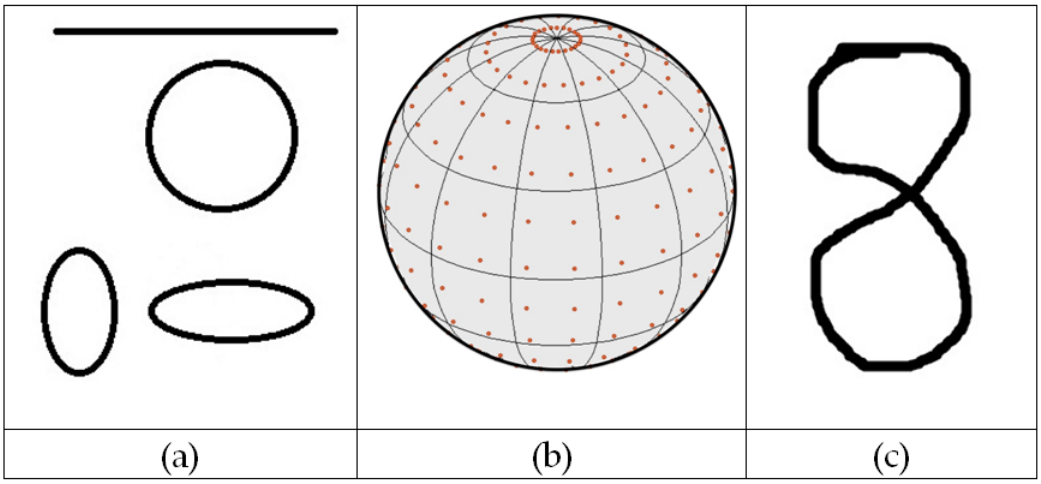


Figure 2.2: (a) 1-D manifold (b) 2-D manifold (c) not 1-D manifold.

As shown in figure (2.2) a line segment, circle, ellipse are 1-D manifold. However figure of eight is not a 1-D manifold as the neighbourhood of cross point are not homeomorphic to Euclidean space. Sphere and torus are 2-D manifold (Appendix B.1). Some special types of manifolds are given here.

2.2.1 Differential Manifold

A topological manifold M with family of charts $\{U_i, \phi_i\}$ (Appendix B.1) such that
(1) $M \subset \bigcup_i U_i$
(2) For $i, j \in \mathbb{Z}$, if $U_i \cap U_j \neq \phi$ then $\phi_i.\phi_j^{-1} : \phi_j^{-1}(U_i \cap U_j) \to \phi_i(U_i \cap U_j)$ is smooth.

$$\phi_i \ : \ U_i \to W_i \subset \mathbb{R}^n$$
$$\phi_j \ : \ U_j \to W_j \subset \mathbb{R}^n$$

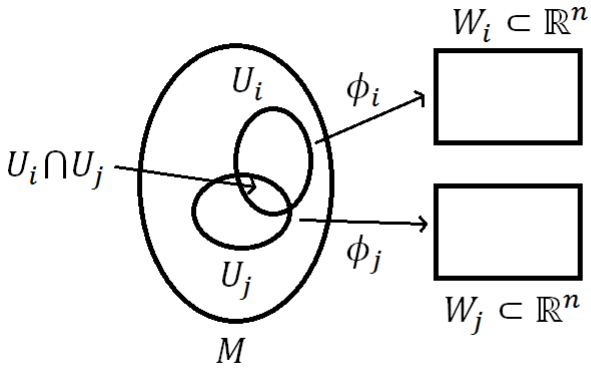


Figure 2.3: Local mapping using charts. After [14].

2.2.2 Riemannian Manifold

To study manifold statistics distance on manifold plays an important role. The idea of distance on manifold is coming from *Riemannian geometry*. A Riemannian manifold is a smooth manifold equipped with a Riemannian metric [11, 15], which is a smooth function that assigns an *inner product* at each point $p \in M$ on the tangent space T_pM.

Let $\gamma : [a, b] \to M$ be a differentiable curve of the Riemannian manifold M. The velocity at any point $t \in [a, b]$ on the curve is given as $\dot{\gamma}(t)$. Then, integrating the norm of velocity at every point on the curve segment give the *length* (whose generalization is metric) of that curve segment, i.e.,

$$L(\gamma) = \int_a^b \parallel \dot{\gamma}(t) \parallel dt$$

The norm at each point is defined on the *tangent space* (Appendix B.2). Riemannian manifold becomes a metric space with this definition of length such that the distance between two distinct point $p, q \in M$ is given by

$$d(p, q) = \inf \{L(\gamma) : \gamma \text{ is differentiable curve joining } p \text{ and } q\} \tag{2.1}$$

2.2.3 Submanifold

There are many cases where a manifold is a subset of other manifold for example, sphere is a subset of $\mathbb{R}^3$. However not all subset of any manifold is itself a manifold. In this section, we try to describe conditions when a subset is a manifold. Such manifolds are also referred as a *submanifolds*.

If there exist a *smooth* function $F : N \to M$, we define the rank of F at $p \in N$

equal to the rank of the *linear* map derivative of the function $DF : T_pN \to T_{F(p)}M$. If F has the same rank k at every point, we say F has *constant rank*, and write rank $F = k$.

An open subset N of a differential manifold M is a manifold with $\{V', \phi'\}$ defined as:

$$V' = N \cap V \text{ and } \phi' = \phi|V'$$

where $\{V, \phi\}$ are charts (Appendix B.1) on M then $\{N, \{V', \phi'\}\}$ is said to be a submanifold on M. Here, the smooth mapping $F : N \to M$ can be done by following ways:

1. Immersion: If there exist a smooth map $F : N \to M$ with the property that DF is *injective* (mapping is one-to-one) at each point or in other words rank $F = \dim(N)$.

   $$F : \mathbb{R} \to \mathbb{R}^2,$$
   $$F(t) = (2cos(t - \pi/2), sin(2(t - \pi/2))), \quad \forall t \in (0, 2\pi).$$

   Such submanifold are called *immersed* submanifold. Lie subgroup are also immersed submanifold.

2. Submersion: If there exist a smooth map $F : N \to M$ with the property that DF is *surjective* (mapping is onto) at each point or in other words rank $F = \dim(M)$.

3. Embedding: Le t $F : N \to M$ be an injective immersion. If F is a homeomorphic from N to $\tilde{N} = F(N)$ and $\tilde{N}$ is considered as a subset of M, then F is said to be a *embedding* and $\tilde{N}$ is said to be an *embedded submanifold* [13].

2.2.4 Lie group

Lie groups [11, 15] arises as a *transformation* groups of geometric objects. Transformations of Euclidean spaces such as translations, scaling ,rotations and affine transformations all are elements of Lie groups.

Lie group is a differentiable manifold such that group operations like the multiplication map $m : G \times G \to G$ and inverse map $i : G \to G$, given by

$$m(a, b) = ab \text{ and } i(a) = a^{-1} \tag{2.2}$$

are C^∞ or *smooth*. General linear group $GL(n, \mathbb{R})$, set of nonsingular $n \times n$ matrices with real entries is an example of a Lie group. Let $A, B \in GL(n, \mathbb{R})$, as A and B are

nonsingular matrices ($\det(A) \neq 0$ and $\det(B) \neq 0$), $m(A, B) = AB \in GL(n, \mathbb{R})$ and $i(A) = A^{-1} \in GL(n, \mathbb{R})$ are smooth.

Subgroups of $GL(n, \mathbb{R})$ are also Lie group. Some subgroups of $GL(n, \mathbb{R})$ are given below.

1. Special linear group $SL(n) = \{x \in GL(n, \mathbb{R}) |\ det(x) = 1\}$.
2. Set of orthogonal matrices $O(n) = \{x \in GL(n, \mathbb{R}) |\ xx^T = I_n\}$.
3. Set of rotation matrices $SO(n) = \{x \in GL(n, \mathbb{R}) |\ det(x) = 1,\ xx^T = I_n\}$.
4. Set of positive definite matrices $PD(n) = \{x \in GL(n, \mathbb{R}) |\ z^T xz > 0$, for any non zero $z \in \mathbb{R}^n\}$.

Action of a group

Let G be a group and X be a set. Then G is said to *act* on X (on the left) if there exist a mapping $\theta : G \times X \to X$ such that

$$\theta(e, x) = x, \quad \forall x \in X \quad \text{and} \quad e \in G$$
$$\theta(g_1, \theta(g_2, x)) = \theta(g_1 g_2, x), \quad \forall g_1, g_2 \in G$$

Here, e is an identity element of group G. If X is a smooth manifold and G is a Lie group, then we assume that θ is *continuous*.

Let $F : N \to M$ and Y be a vector field [16] on M such that $\forall q \in M$ and $p \in F^{-1} = q \subset N$ there exist $F_* : T_P N \to T_{F(p)} M$ as $F_*(X_p) = Y_q$ then we say that X and Y are *F-related*. If F is *diffeomorphism* then each vector field X on N is F-related to a uniquely determined vector field Y on M.

If $F : M \to M$ is a diffeomorphism and X is a C^∞ vector field on M such that $F_*(X) = X$, that is X is F-related to itself, then X is said to be invariant with respect to F or *F-invariant* [16].

$$F_*(X_p) = X_q \quad \text{if } F(p) = q; \quad \forall p \in M$$

In other words, for G be a Lie group and $T_e(G)$ the tangent space at identity. Then each $X_e \in T_e(G)$ determines uniquely a vector field X on G which is invariant under left translations.

$$g\, e = g \quad \forall g \in G$$

$$L_{g*} = T_e G \to T_g G$$

It can be given as a transformation from a fixed point. As a fixed point, we can take identity element e.

One parameter subgroup

Let $(\mathbb{R}, +)$ be the additive group of real number, considered as a Lie group. A one parameter subgroup H of a Lie group G is the homeomorphic image $H = F(\mathbb{R})$ of a homeomorphism $F : \mathbb{R} \to G$. It is said trivial if $H = e$.

Let X be left invariant vector field on a Lie group G. The integral curve $F(t) = \theta_e(t) = \theta(t, e)$ is a one parameter subgroup of a lie group.

Here, every X_e determines left invariant vector field X on G, there is a one-one correspondence between T_eG and one parameter subgroup.

Up till now, we have discussed basics required to study manifold and Lie group. In the next section, we give theoretical background for computation on manifolds like geodesics, Exponential map, Logarithm map and geodesic mean.

2.3 Computations on Manifolds

2.3.1 Geodesic

Geodesic path [17] is the generalization of straight line on curved surfaces (e.g., manifold). Geodesics are defined to be curves whose tangent vectors remain parallel if they are transported along it. A curve $c(t) \in M$ and $Z(t)$ is a tangential vector field on a manifold such that $Z(t) = \frac{dc(t)}{dt}$ then $\frac{DZ}{dt} \equiv 0$ if $c(t)$ is geodesic [17]. Here $\frac{D}{dt}$ is called *covariant derivative*, kindly refer [Appendix B.5] for more detail on covariant derivative.

If $c(t)$ is a geodesic, then it is a local minimizer of the length of path between two given points [11].

Now using equation (B.10) we can say for each point on geodesic curve of an manifold, we have

$$\frac{\partial b^k}{dt} + \sum_{i,j=1}^{m} \Gamma_{ij}^k(u) b^i \frac{\partial u^i}{dt} = 0 \tag{2.3}$$

So, for given a curve $c(t)$, we can find whether it is a geodesic or not using equation (2.3). However to compute the geodesic in a given direction starting from a point, we need to solve Initial Value Problem (IVP) corresponding to equation (2.3).

2.3.2 Riemannian Exponential Map

Exponential map [11, 13, 17], $Exp : TM \rightarrow M$ for a point $p \in M$ of the tangent vector $X \in T_pM$, gives a point on M along the geodesic from $\tilde{p}$ in the direction of

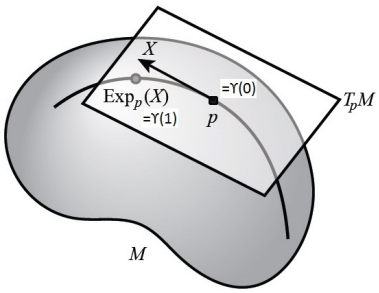


Figure 2.4: Riemannian Exponential map on a manifold. After [8].

X obtained by travelling along the geodesic for a unit time. Let γ be a geodesic with $\gamma(0) = p$ and $\dot{\gamma}(0) = X$. The exponential map at point p is defined as:

$$Exp_p(v) = \gamma(1) = \tilde{p} \tag{2.4}$$

Exponential Map on Lie Group

The tangent space of a Lie group G is denoted by $\mathfrak{g}$ which forms a Lie algebra. For a Lie group G with bi-invariant metric, the Lie group exponential map at the identity is same as ordinary matrix exponential given by series expansion, that is, for any tangent vector $X \in \mathfrak{g}$

$$Exp_e(X) = \sum_{k=1}^{\infty} \frac{X^k}{k!} = I + X + \frac{X^2}{2} + \frac{X^3}{6} + ... \tag{2.5}$$

where I is the identity matrix.

Using the left-invariance of the Riemannian metric, any geodesic at a point $p \in G$ may be written as the left multiplication of a geodesic at the identity. That is, the geodesic with initial conditions $\gamma(0) = p$ and $\gamma'(0) = L_{g*}(X)$ is given by

$$\gamma(t) = p\, exp(tX).$$

2.3.3 Riemannian Logarithm Map

For a point $\tilde{p} \in M$ which is in the neighbourhood of $p \in M$, we can find a unique path minimizer geodesic joining this two point. For given p and $\tilde{p}$, Log map $Log_p(\tilde{p})$ gives a tangent vector $v \in T_pM$ through which one can reach at $\tilde{p}$

travelling from p for a unit time along the geodesic or in other words Riemannian Logarithm map is a inverse of Riemannian Exponential map in the neighbourhood such that $\tilde{p} = Exp_p(Log_p(\tilde{p}))$.

Also Log Map is used to define distance between two points of the manifold, i.e.,

$$d(p,q) = \parallel Log_p(q) \parallel . \tag{2.6}$$

2.3.4 Parallel Transport

It is the method of transferring *geometric* data along the curve on manifold. Here, we want to transport a vector on the tangent space of a point to the tangent space of an another point. Parallel transport is done along the curve on a manifold such that it remains parallel with respect to covariant derivative (Appendix B.5). Let $c(t)$ be a curve on manifold such that p, q be points on the curve and a vector $v \in T_pM$. Now, we want to transport v from p to q such that $\frac{Dv}{dt} \equiv 0$ along the path. Note here the parallel transport of a vector is a path-dependent [11, 17].

2.3.5 Mean of Data

In the field of pattern recognition, to perform data classification, mean of a cluster plays key role. Considering a case that data points given in an Euclidean space $x_i \in \mathbb{R}^n$ for $i = 1, 2..., l$, mean is given as:

$$m = \frac{1}{l} \sum_{i=1}^{l} x_i, \quad \text{where } m \in \mathbb{R}^n$$

It is also called *arithmetic mean*. However, we can not use above expression to find mean of the cluster on manifold, because it may be possible that computed $m \notin M$.

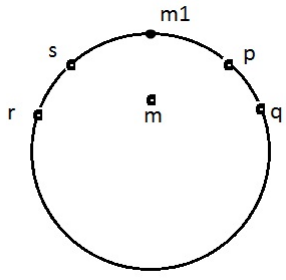


Figure 2.5: Arithmetic mean (m) and Geodesic mean ($m1$).

From, figure (2.5), it can be seen that arithmetic mean does not provide the cluster mean on manifold as it is a special case of mean in Euclidean space. So, we need to generalize the idea of mean which is called as a *Fréchet mean* or a *geodesic mean*.

The geodesic mean of set of points gives a center point such total of squared distance between points from set and center point is *minimum*. Let $x_i \in (M, d)$ for $i = 1, 2..., l$ be observation on manifold M them geodesic mean (m) is given as:

$$m = \arg\min_{\hat{m} \in M} \sum_{i=1}^{l} d(x_i, \hat{m})^2,$$

which is used to find geodesic mean of a cluster on manifold as shown in figure (2.5).

CHAPTER 3
Support Vector Machines

3.1 Introduction

There are many algorithms for data classification in case data belongs to some n-dimensional space. In our case, data belongs to some curved manifolds, we can not directly apply such algorithms for classification. Here, we want to try and extend such algorithm, for that we first choose to work with Support Vector Machine (SVM). In this chapter, we give an introduction to an Euclidean SVM.

3.2 Support Vector Machines

Support vector machines (SVM) [2, 18] is a supervised learning algorithm such that it analyses and classifies data into two different classes. Given two data clusters, it finds a separating hyperplane which makes it a *binary* linear classifier. SVM works on the idea of finding a separating hyperplane with a large *margin*. Given a set of training data along with labels, SVM training algorithm develops a model such that it can assign new testing data into one of the categories.

3.2.1 Notion of Margin

For data points $x_i \in \mathbb{R}^n$ with labels $y \in \{-1, 1\}$, consider two-class problem with a linear classifier. The number of features per sample is n and number of sample points are m. The idea is to find a separating hyperplane (or a decision function $g(z)$) such that a model $sign(h_{w,b}(x))$ is decision function separating input data into one class with $h_{w,b}(x) > 0$ and another class with $h_{w,b}(x) < 0$.

$$g(x) = sign(h_{w,b}(x)) = sign(w^T x + b), \tag{3.1}$$

where $g(x) = -1$ if $w^T x + b < 0$ and $g(x) = +1$ if $w^T x + b > 0$, [18].

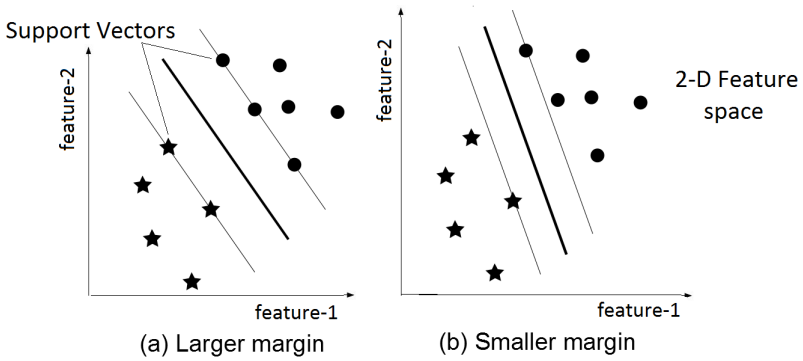


Figure 3.1: Choice of hyperplane.

The notion of margin is important in several classification algorithms, as it can be used to bound the total loss of the classifier. We can introduce a loss by model as the portion of training data can be explained perfectly well (zero-loss function) and the data can only be explained by some uncertainty (non-zero loss). In classification problem, margin is referred as the region with non-zero loss or in another words margin is refereed as the distance between the hyperplane with the nearest point from training data [18].

A good decision boundary can be achieved by:
(1) Minimizing total loss for sample lie inside the margin
(2) Correctly classify data clusters by achieving maximum separating margin

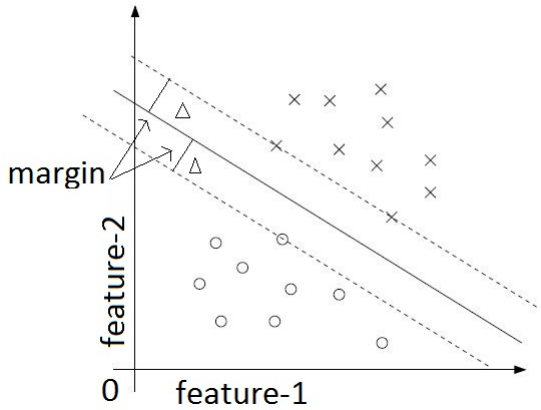


Figure 3.2: Separating hyperplane with maximum margin.

Clearly above two goals are contradictory, if we try to maximize the margin it will introduce higher loss as the amount of data lying inside the margin increases.

So, we need to choose appropriate margin size. Let $\triangle$ be margin then

$$\begin{aligned} y = +1 \;\; if \;\; w^T x + b \geq \triangle, \\ y = -1 \;\; if \;\; w^T x + b \leq \triangle. \end{aligned} \tag{3.2}$$

Above equations can be clubbed into one, we get

$$\begin{aligned} \frac{y_i(w^T x_i + b)}{||w||} \geq \triangle, \quad i = 1, 2, ..., m, \\ y_i(w^T x_i + b) \geq \triangle ||w||, \quad i = 1, 2, ..., m. \end{aligned}$$

Let us set $\triangle ||w|| = 1$, [18, 19]. So, our constraints are

$$y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, ..., m. \tag{3.3}$$

3.2.2 Distance

To find distance from $p = (x_0, y_0, z_0)' \in \mathbb{R}^3$ and a hyperplane $h_{w,b} = 0$ where $w = (w_1, w_2, w_3)'$.

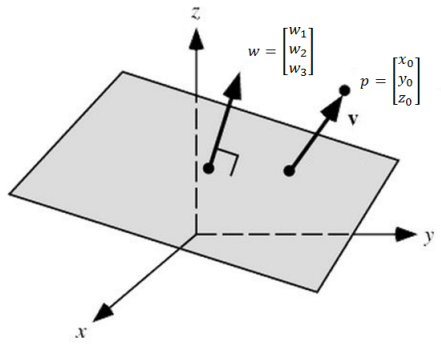


Figure 3.3: Distance from hyperplane to point.

Let us take any point $m = (x_1, y_1, z_1)'$ on the *hyper plane* then $v = \overrightarrow{mp}$

$$\begin{aligned} dist &= \langle w, v \rangle, \\ &= w^T v. \end{aligned}$$

From the above discussion, we can derive a distance function from any point $p \in R^n$ to a hyperplane $h_{w,b} = 0$ as:

$$dist = \frac{w^T x + b}{||w||}. \tag{3.4}$$

So to maximize the minimum distance from a hyperplane, we need to minimize $||w||$ which is equivalent to

$$\min \ \ ||w||^2 \ \text{ or } \ \frac{||w||^2}{2}. \tag{3.5}$$

3.2.3 Optimization Problem

From the above discussion, we have the following optimization problem [18, 19].

$$\min \ \frac{||w||^2}{2},$$

subject to

$$y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, ..., m. \tag{3.6}$$

The given optimization problem is a *convex quadratic* objective function [20] with linear constraints. Optimal margin classifier hyperplane can be computed by solving above optimization problem. We can solve such problem using quadratic programming (QP). For that we need to study Lagrange duality and Karush-Kuhn-Tucker (KKT) condition [20].

3.2.4 Lagrange Duality and Karush-Kuhn-Tucker Conditions

Consider the following optimization problem, often referred as primal problem:

$$\min \ f(w).$$

Subject to the following equality and inequality constrains

$$g_i(w) \leq 0, \quad i = 1, 2, ...m,$$
$$h_i(w) = 0, \quad i = 1, 2, ...l.$$

To solve it, first of all, we define generalized Lagrangian as:

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{m} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w), \tag{3.7}$$

where, the α_i and β_i are the Lagrange multipliers where *Primal* problem is given by

$$\theta_P(w) = \max_{\alpha, \beta : \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$$

Hence, for the minimization problem

$$P^* = \min_w \theta_P(w) = \min_w \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta), \tag{3.8}$$

where P^* is value of the primal problem [20]. Similarly the *Dual* optimization problem is given by

$$\theta_D(\alpha, \beta) = \min \mathcal{L}(w, \alpha, \beta),$$

$$D^* = \max_{\alpha,\beta:\alpha_i \geq 0} \theta_D(\alpha, \beta) = \max_{\alpha,\beta:\alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta), \tag{3.9}$$

where D^* is value of the dual problem [20].

This is exactly the same as our primal problem shown above, except that the order of the maximization and the minimization are now exchanged. The relation between primal and dual problem [20] is given as:

$$\max_{\alpha,\beta:\alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) = D^* \leq P^* = \min_w \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) \tag{3.10}$$

However, under certain conditions, we will have $P^* = D^*$. Here, $f(w) = \frac{1}{2} w^T w$ which is a convex function and all linear functions (i.e., g_i) are also convex function. As given constraints, g_i are feasible, there exist w such that $g_i < 0, \forall i$. Under our above assumptions and if Karush-Kuhn-Tucker (KKT) conditions are satisfied then there must exist optimal w^*, α^*, β^*, such that w^* is the solution to the primal problem, α^*, β^* are the solution to the *dual* problem, and moreover $P^* = D^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$ [18].

Karush-Kuhn-Tucker (KKT) conditions [18, 20], which are as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}(w^*, \alpha^*, \beta^*)}{\partial w_i} &= 0 \qquad i = 1, 2, ...n, \\
\frac{\partial \mathcal{L}(w^*, \alpha^*, \beta^*)}{\partial \beta_i} &= 0 \qquad i = 1, 2, ...l, \\
\alpha_i^* g_i(w^*) &= 0 \qquad i = 1, 2, ...m, \\
g_i(w^*) &\leq 0 \qquad i = 1, 2, ...m, \\
\alpha^* &\geq 0 \qquad i = 1, 2, ...m.
\end{aligned} \tag{3.11}$$

If there exist some w^*, α^*, β^* satisfying the KKT conditions, then it is the solution to the primal and dual problems. Having now laid the theoretical foundation in optimization theory, we next explain how to compute the decision boundary for the optimal margin classifier.

3.3 Optimal Margin Classifiers

3.3.1 Case 1: Linearly Separable

In case if given data is *linearly* separable, we can find out a optimum hyperplane which can separate data clusters with higher margin. The optimization problem is given as:

$$min \frac{||w||^2}{2},$$

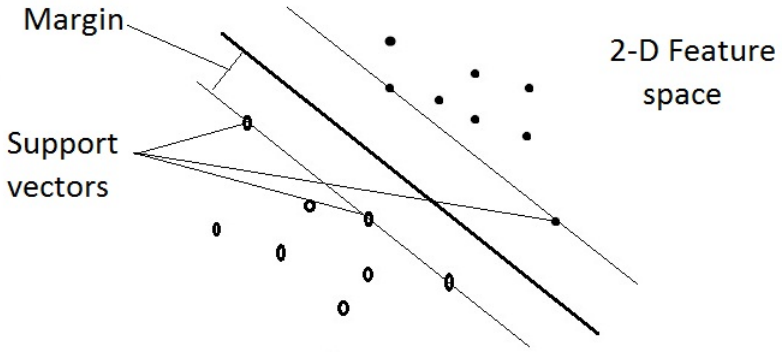
subject to

$$y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, ..., m.$$

We can write the constraints as

$$g_i(w) = -y_i(w^T x_i + b) + 1 \leq 0, \quad i = 1, 2, ..., m.$$

From the KKT conditions, we have $\alpha_i > 0$ only for constraints that hold with equality, $g_i(w) = 0$ or in other words, the training samples that have margin exactly equal to one.



The points that are having the smallest margins are the ones closest to the decision boundary. From figure (3.3.1), there are five points (three on left side and two on right side of the separating hyperplane) lying on the thinner lines parallel to the decision boundary (shown by darker line). Thus, only five of the α_i, are going to be non-zero at the optimal solution to our optimization problem. These five points are called the *support vectors* [18, 19]. Only support vectors actively take part in defining separating hyperplane from given training dataset. Here, we can see the number of support vectors are much smaller than the total size a training dataset.

Lets find the dual form of the problem.

$$\mathcal{L}(w, \alpha, b) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i [y_i(w^T x + b) - 1] \tag{3.12}$$

For that we will first minimize $\mathcal{L}(w, \alpha, b)$ with respect to w and b for fixed α. Derivative of $\mathcal{L}(w, \alpha, b)$ with respect to w is given by

$$\bigtriangledown_w \mathcal{L}(w, \alpha, b) = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0,$$

which implies

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i. \tag{3.13}$$

Derivative of $\mathcal{L}(w, \alpha, b)$ with respect to b

$$\bigtriangledown_b \mathcal{L}(w, \alpha, b) = \sum_{i=1}^{m} \alpha_i y_i = 0, \tag{3.14}$$

also

$$||w||^2 = w^T w. \tag{3.15}$$

Using equation (3.13) to (3.15) in equation (3.12), we get

$$\mathcal{L}(w, \alpha, b) = \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j (x_i)^T (x_j). \tag{3.16}$$

We reached to equation (3.16) by minimizing $\mathcal{L}$ with respect to w and b. Putting this together with the constraints $\alpha_i \geq 0$ and the constraint (3.14), we obtain the following dual optimization problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j (x_i)^T (x_j),$$

subject to

$$\alpha_i \geq 0 \quad i = 1, 2, ...k,$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{3.17}$$

As requires condition for $P^* = D^*$ are satisfied, we can solve dual problem instead of our original primal problem. The dual problem (3.17) is taken care by Sequential Minimal Optimization (SMO) algorithm [19]. After finding α_i (that

maximize $W(\alpha)$ subject to the constraints), we can use equation (3.13) to find the optimal w as a function of α. The optimal value for b can be obtained by:

$$b^* = -\frac{\max\limits_{i:y_i=-1} w^T x + \min\limits_{i:y_i=1} w^T x}{2}. \tag{3.18}$$

Finally, decision boundary function is given as:

$$\begin{aligned}(w^*)^T x + b^* &= (\sum_{i=1}^{m} \alpha_i y_i x_i)^T x + b^*, \\ &= \sum_{i=1}^{m} \alpha_i y_i \langle x_i, x \rangle + b^*. \end{aligned} \tag{3.19}$$

From equation (3.19), it is clear that we need to calculate the inner product between x and support vectors in order to classify data.

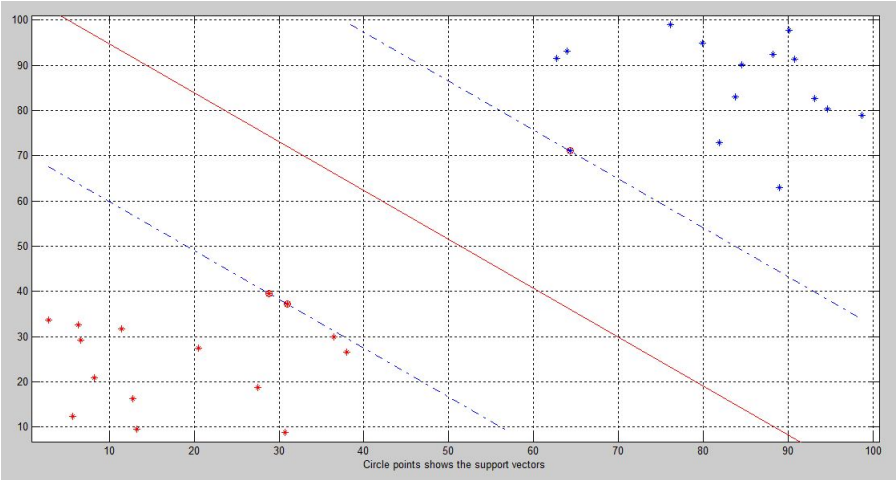


Figure 3.4: Implementation of linear classification.

In figure (3.4), doted line shows hyperplane by support vectors, continuous line shows the optimal hyperplane (in above case, it is a line) which is used to classify data.

3.3.2 Case 2: Non linearly Separable

In case given data is *not* linearly separable, we need to map into higher dimensional space using some *kernel* functions, like radial basis function (RBF) or polynomial kernel function. Since, the equation (3.19) can be written entirely in terms of the inner products $\langle x, y \rangle$. This means that we can replace all those inner products with $\langle \phi(x), \phi(y) \rangle$, given a feature mapping ϕ. We define the corresponding

kernel [19] to be

$$K(x, y) = \phi(x)^T \phi(y),$$

We can write equation (3.19) using kernel function as:

$$h(x) = \sum_{i=1}^{m} k(x_i, x) + b^*. \tag{3.20}$$

One of the common kernel function if RBF function which is given by

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

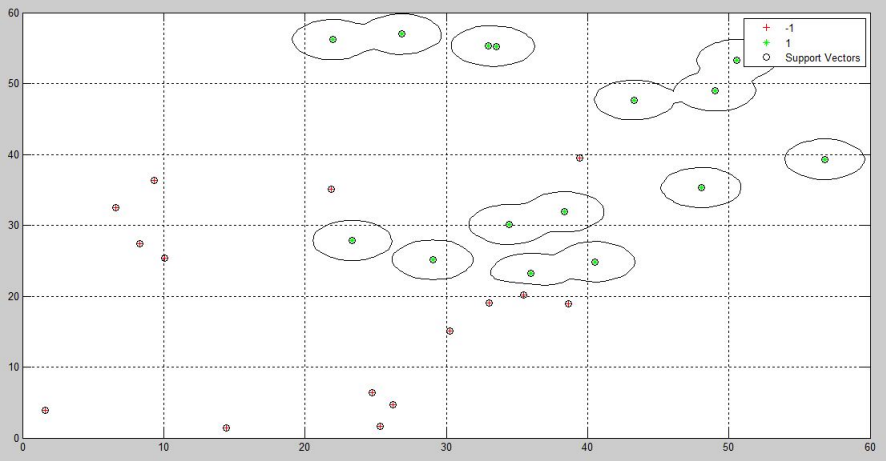


Figure 3.5: Implementation of nonlinear classification-RBF kernel.

From figure (3.5), we can see that given two classes are not linearly separable. Here, we classify such data using RBF kernel function.

3.4 Chapter Summary

In this chapter we try to explain working principle of usual SVM for data in $\mathbb{R}^n$. To classify data on manifold, we need generalize SVM for manifold data. In chapter 2 we give detailed idea about manifold. In the next chapter using manifold statistics we try to classify data on manifold. Here, we compare classification results on the tangent space of different point and give an algorithm to find best point on whose tangent space we need to project for better classification.

CHAPTER 4
Classification of Data on Manifold

4.1 Introduction

We use the idea of a tangent space SVM by applying SVM on the tangent space of a manifold. As the tangent space forms a vector space, we can successfully apply SVM on it. In this thesis work, the projection of data from manifold to the tangent space of a point $p \in M$ and vice-versa is done using Riemannian Logarithm map and Riemannian Exponential map, respectively, which are *diffeomorphic* in the neighbourhood of a point p. We propose a method to find a point $p \in M$ on whose tangent space we can classify the given data into different classes with maximum margin. Next section describes algorithms to implement Riemannian Exponential map, Riemannian Logarithm map and Geodesic mean on various manifolds followed by our proposed algorithm with required mathematical background and experiments. Here, we also give an application of texture classification using $PD(n)$ matrices.

4.2 Computations on Manifolds

For data classification, we require geodesic distance and geodesic mean of the data clusters for each manifold which require use of Exponential map and Logarithm map. As we know for $p \in M$ and a tangent vector $v \in T_pM$

$$\begin{aligned} &Log_p : M \to T_pM, \\ &Log_p(q) = v, \\ &Exp_p : T_pM \to M, \\ &Exp_p(v) = \gamma(1) = q, \\ &d(p,q) = ||v||_{T_pM}, \end{aligned} \tag{4.1}$$

To perform above operation on general manifold, we need to define local co-

ordinate frame on the tangent space of a manifold. Please refer (Appendix B) for detailed explanation of charts on manifold, tangent space and frame. Detailed explanation of Riemannian Exponential map and Riemannian Logarithm map implementation on general manifold are described in next subsection.

4.2.1 Riemannian Exponential Map on Parametric Manifolds

Let n-dimensional manifold M is embedded in $\mathbb{R}^m$ where $m > n$. Initially, all the parameter are given with respect to embedding space. Implementation steps for $Exp_P(X_p)$ (with $p \in M$ and $X_p \in T_pM \subset \mathbb{R}^m$ and defined charts $\{\phi_i, U_i\}$) are as follows:

1. Find local orthogonal coordinate frame $\{F_i\}$ for $i = 1, 2, ..., n$ at a point p which are linearly independent and represent X_p with local coordinate system

$$X_p = \sum_{i=1}^{n} F_i b_i$$

where b_i's are coefficient of local coordinate frame.

2. Solve given equation and compute local coordinate for the next point for a

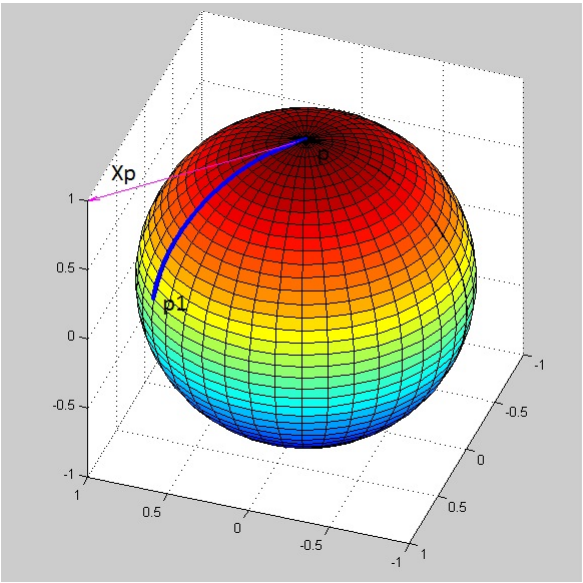


Figure 4.1: Example of Exponential map.

given step size (for computation, we took step size $dt = 0.001$). For more details, kindly refer (Appendix B.5).

$$\frac{\partial b^k}{dt} + \sum_{i,j=1}^{m} \Gamma_{ij}^k(u) b^i \frac{\partial u^i}{dt} = 0$$

3. Repeat for $p = p_{new}$ until unit distance is covered (or repeat for (1/step size) iterations, in our case, for 1000 iterations).

Exponential map on S^2 manifold for a given point $p = [0, 0, 1]$ and a tangent vector $X_p = [1, 1, 0]$ is given by $Exp_p(X_p) = q$ as shown in figure (4.1), where $q = [0.6989, 0.6989, 0.1544] \in M$. Kindly refer [21] for more details on close form solution of Exponential map on S^2 manifold.

4.2.2 Exponential map on Lie Group

Here, we give close form solutions and algorithm for Exp map on $SO(3)$ and $PD(n)$ manifolds. The tangent space of rotation matrices ($SO(n)$) and positive definite matrices ($PD(n)$) manifolds are set of skew-symmetric matrices ($so(n)$) and set of symmetric matrices ($Sym(n)$) respectively.

1. Exponential map On $SO(3)$ manifold [9] is given as:

$$\begin{aligned} &Exp : so(3) \to SO(3), \\ &Exp_p(v) = pExp(v) = p\left(I + \frac{sin(\parallel \hat{v} \parallel)}{\parallel \hat{v} \parallel} v + \frac{1 - cos(\parallel \hat{v} \parallel)}{\parallel \hat{v} \parallel^2} v^2\right), \end{aligned} \tag{4.2}$$

where $v \in T(SO(3)) = so(3)$ and $\hat{v}$ is $\mathbb{R}^3$ representation of v given as:

$$v = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \text{ then } \quad \hat{v} = (v_1, v_2, v_3) \in \mathbb{R}^3. \tag{4.3}$$

2. Exponential map $Exp : Sym(n) \to PD(n)$ On $PD(n)$ manifold [4, 6], is given by following algorithm

Box 4.1**: Algorithm for Riemannian Exponential Map**

Input: Initial point $p \in PD(n)$, Tangent vector $X \in Sym(n)$.
Output: $Exp_p(X)$
Let $p = u\Lambda u^T$ $(u \in SO(n), \Lambda$ diagonal)
$g = u\sqrt{\Lambda}$
$Y = g^{-1}X(g^{-1})^T$
Let $Y = v\Sigma v^T$ $(v \in SO(n), \Sigma$ diagonal)
$Exp_p(X) = (gv)exp(\Sigma)(gv)^T$

4.2.3 Parallel Transport on Manifolds

From previous discussion (in section 2.3.4), we know that parallel transport of a vector from the tangent space of one point to another point is path-dependent. We perform parallel transport along the geodesic path. Implementation steps for parallel transport of a vector $v \in T_pM \subset \mathbb{R}^m$ along the geodesic path joining p and $q \in M$ is given as:

1. Find local co-ordinate frame $\{F_i\}$ for $i = 1, 2, ..., n$ at a point p and represent v with local coordinate system

$$v = \sum_{i=1}^{n} F_i b_i,$$

   where b_i's are coefficient of local coordinate frame.

2. Solve given equation for the next point along the path

$$\frac{\partial b^k}{dt} + \sum_{i,j=1}^{m} \Gamma_{ij}^k(u) b^i \frac{\partial u^i}{dt} = 0,$$

   and compute $\widehat{b_i}$'s, coefficient of local coordinate frame on the tangent space of a new point.

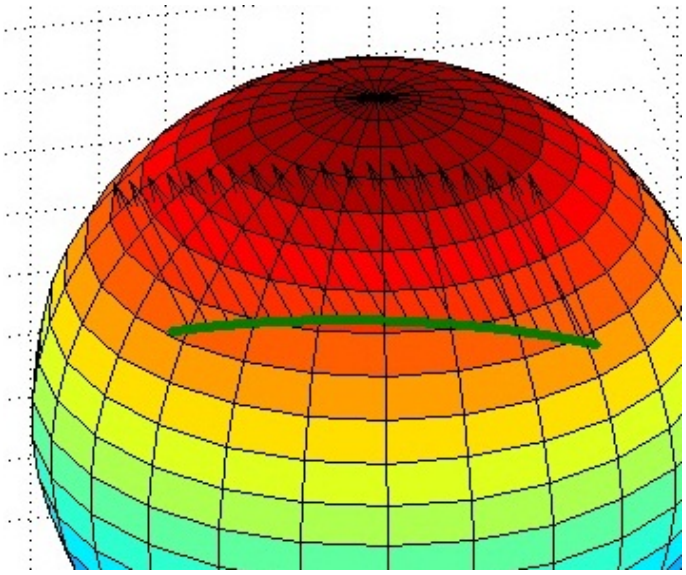


Figure 4.2: Parallel transport.

3. Representation of v on the tangent space of a new point is given by

$$\widehat{v} = \sum_{i=1}^{n} \widehat{F}_i \widehat{b}_i,$$

where $\{\widehat{F}_i\}$ is the local coordinate frame of a new point.

4. Repeat until point q is reached

Here, we want to transport v from p to q such that $\frac{Dv}{dt} \equiv 0$ along the geodesic path. Parallel transport of $v = [0,\ -0.8274,\ 0.6752]$ at p along the geodesic line joining $p = \left[\frac{1}{2}, \sqrt{\frac{2}{10}}, \sqrt{\frac{3}{10}}\right]$ and $q = \left[-\frac{1}{2}, \sqrt{\frac{2}{10}}, \sqrt{\frac{3}{10}}\right]$ is as shown in figure (4.2).

4.2.4 Riemannian Logarithm Map on Parametric Manifolds

Implementation steps for $Log_p(q)$ (for $p, q \in M$) are as follows:

1. Find $X_0 = \Pi'(\overrightarrow{pq})$, where $\Pi' : T_p\mathbb{R}^n \to T_pM$ such that $\Pi'(\overrightarrow{pq}) \in T_pM$.
2. Compute $q' = Exp_p(X_0) \in M$ and geodesic $C_{pq'}(t) = Exp_p(tX_0)$ for $t \in [0, 1]$.
3. Let $e = \Pi'(\overrightarrow{q'q})$, where $\Pi'(\overrightarrow{q'q}) \in T'_qM$.
4. Take $C(t) = C_{pq'}(1-t)$ such that $C(0) = q'$ and $C(1) = p$.
5. Perform *Parallel Transport* of e along $c(t)$ from q' to p and find e_1 such that $\frac{De}{dt} \equiv 0$ where $e_1 \in T_pM$.
6. Calculate $X_k = X_{k-1} + \triangle t\ e_1$.
7. Repeat from [2.] for $X_0 = X_k$ until $\| e_1 \| < \delta$, where $\| e_1 \|^2 = \langle e_1, e_1 \rangle_{T_pM}$.

Logarithm Map between two points $p = \left[\frac{1}{2}, \sqrt{\frac{2}{10}}, \sqrt{\frac{3}{10}}\right]$ and $q = \left[-\sqrt{\frac{1}{4}}, \sqrt{\frac{1}{2}}, \sqrt{\frac{1}{4}}\right]$ is as shown in figure (4.3).

$Log_p(q) = X_p$ where $X_p \in T_pM$ and $X_p = [-0.1656,\ 0.9202,\ 0.7883]$. So $d(p,q) = \|Log_p(q)\| = \|X_p\| = 1.2230$. Using conventional notion of distance on unit sphere, $d(p,q) = r \cdot acos(p \cdot q) = 1 \cdot 1.2238 = 1.2238$. Please refer [21] for more detail on close form solution of Log map on S^2 manifold.

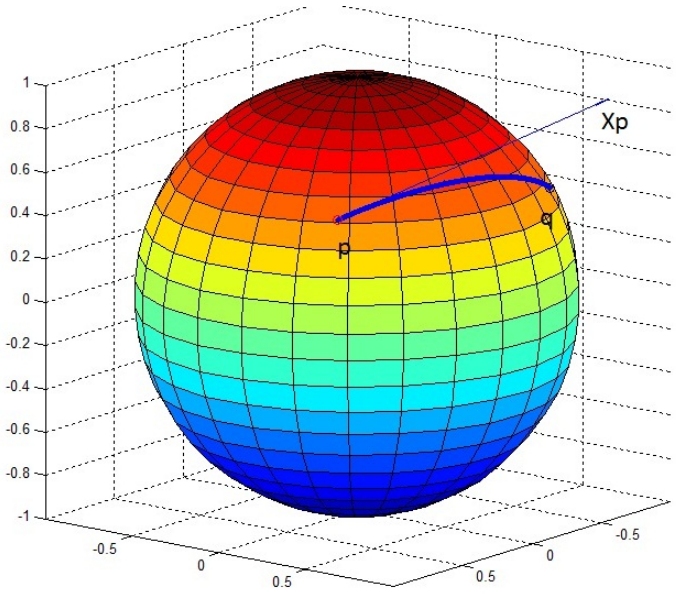


Figure 4.3: Example of Logarithm map.

4.2.5 Logarithm Map on Lie Group

Close form solutions and algorithm for the Log map on $SO(3)$ and $PD(n)$ manifolds are given as:

1. Logarithm map on $SO(3)$ manifold [9] is given as:

$$\begin{aligned} &Log : SO(3) \rightarrow so(3) \\ &Log(p^T q) = \frac{\theta}{2sin(\theta)} \begin{bmatrix} -R(1,2) + R(2,1) \\ R(3,1) - R(1,3) \\ -R(2,3) + R(3,2) \end{bmatrix}, \end{aligned} \tag{4.4}$$

where $R = p^T q$ and $\theta = \frac{acos(trace(R)-1)}{2}$, for detailed explanation refer [9].

To use tangent space SVM as a classifier, we need to find equivalent vector from of data. For $SO(3)$ matrices after projecting the data on a tangent space of a point $p \in SO(3)$, a tangent vector $w \in so(3)$ on the tangent space of a point p can be given as a matrix of the form $w = pv$ where v is *skew-symmetric* matrix.

We can represent v as a tangent vector at a Identity $v \in T_I(SO(3))$ considering a smooth lie group action $\phi_g : w \rightarrow g^{-1}w$ such that a tangent vector at the Identity in the Lie group is related to a point p such $\langle w, w \rangle_p = \langle v, v \rangle_I$ [9].

We need to find $\tilde{v} \in \mathbb{R}^3$ such that $\langle v, v \rangle_I = ||\tilde{v}||^2$ where $||\tilde{v}||^2 = \tilde{v}^T\tilde{v} = \hat{v}^T D\hat{v}$ and $\hat{v} \in \mathbb{R}^3$ is a equivalent vector form of $v \in so(3)$ (4.3).

$$v = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \text{ then } \quad \hat{v} = (v_1, v_2, v_3) \in \mathbb{R}^3.$$

Here, D is 3×3 matrix which can be decomposed as $D = D^{\frac{1}{2}T} D^{\frac{1}{2}}$ and we get, $\tilde{v} = D^{\frac{1}{2}}\hat{v} \in \mathbb{R}^3$. As we know SVM relies on the fact that underlying space is a vector space. Here, we use $\hat{v} \in \mathbb{R}^3$ as a data for classification purpose.

2. Log map $Log : PD(n) \to Sym(n)$ On $PD(n)$ manifold [4, 6], is given by following algorithm

Box 4.2**: Algorithm for Riemannian Logarithm Map**

Input: Initial point $p \in PD(n)$, end point $q \in PD(n)$.
Output: $Log_p(q)$
Let $p = u\Lambda u^T (u \in SO(n), \Lambda$ diagonal
$g = u\sqrt{\Lambda}$
$y = g^{-1}q(g^{-1})^T$
Let $y = v\Sigma v^T (v \in SO(n), \Sigma$ diagonal
$Log_q(X) = (gv)log(\Sigma)(gv)^T$

To use tangent space SVM as a classifier, we need to find equivalent vector form of data. In case of $PD(2)$ matrices after projecting the data on a tangent space of a point $p \in PD(2)$, a tangent vector $w \in Sym(2)$ at a point p can be given as a matrix of the form $w = gvg^T$.

Here, v is a *symmetric* matrix and for any matrix q such that $qp = I_2$, we find $q = gg^T$ with $g \in GL^+(2)$. For computational simplicity, we take $g = p^{-\frac{1}{2}}$. We can represent v as a tangent vector at the identity $v \in T_I(PD(2))$ considering a smooth lie group action $\phi_g : w \to gwg^T$ such that a tangent vector at the identity in the Lie group is related to a point p by $\langle w, w \rangle_p = \langle v, v \rangle_I$ [6].

We need to find $\tilde{v} \in \mathbb{R}^3$ such that $\langle v, v \rangle_I = ||\tilde{v}||^2$ where $||\tilde{v}||^2 = \tilde{v}^T\tilde{v} = \hat{v}^T D\hat{v}$ and $\hat{v} \in \mathbb{R}^3$ is a equivalent vector form of $v \in sym(2)$ (4.3).

$$v = \begin{bmatrix} v_1 & v_2 \\ v_2 & v_3 \end{bmatrix} \text{ then } \quad \hat{v} = (v_1, v_2, v_3) \in \mathbb{R}^3.$$

Here, D is 3×3 matrix which can be decomposed as $D = D^{\frac{1}{2}T} D^{\frac{1}{2}}$ and we get, $\tilde{v} = D^{\frac{1}{2}}\hat{v} \in \mathbb{R}^3$. Here, we use $\hat{v}$ as a data for classification purpose.

4.2.6 Geodesic Mean on Manifolds

We assume that given data belongs on a given manifold with defined Riemannian Exponential map and Riemannian Logarithm map, computational algorithm for geodesic mean is given as [4, 6]:

Box 4.3**: Algorithm for Geodesic Mean**

Input: $x_1, x_2, ..., x_n \in M$.
Output: $\mu \in M$, the geodesic mean
$\mu_0 = x_1$
$\tau = 1$ the initial step size
Do
$\Delta\mu_i = \frac{1}{N}\Sigma_{k=1}^{n} Log_{\mu_j}(x_k)$
$\mu_{j+1} = Exp_{\mu_j}(\tau\Delta\mu)$
if $||\Delta\mu_i|| > ||\Delta\mu_{i-1}||$
$\tau = \tau/2$ and $\Delta\mu_i = \Delta\mu_{i-1}$
While $\|\Delta\mu\| > \epsilon$

4.3 Proposed Algorithm

As we stated earlier, classification of data on a manifold is done by using the idea of tangent space SVM. We can perform data classification by projecting data on the tangent space of the base point. The method proposed in [3, 5] use geodesic mean as a base point. Choosing a base point is a crucial issue as we may not be able to separate data clusters *linearly* on the tangent space of a base point [8]. Here, we assume that data is linearly separable on the tangent space of a base point. Under this assumption, we further improve the classification results by finding the proposed point which is a local margin maximizer point using our algorithm. The proposed algorithm with detailed explanation and required derivation is given next.

We are given points $x_i \in M$ for $i = 1, 2, ...m$ with labels $\{-1, 1\}$ or in other words, we have two data clusters $C1$ and $C2 \subset M$. For given data points, first we calculate Log map at a base point p, $\{Log_p x_1, Log_p x_2, ..., Log_p x_m\} \subset T_pM$. In our case, the Lagrangian is also a function of a point p on whose tangent space we project the data. Hence the Lagrangian equation adapted from equation (3.12) is now given by

$$\mathcal{L}(p, w, \alpha, b) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i [y_i(w^T Log_p x + b) - 1]. \tag{4.5}$$

We solve the above problem iteratively. First, for fixed p, optimize the function with respect to $w, \alpha,$ and b. In other words, we apply usual SVM at the tangent space of a fixed point p. Lastly, considering $w, \alpha,$ and b as constants, solve the Lagrangian for p to find a next intermediate base point using the following:

$$\begin{aligned}\mathcal{L}(p) &= \frac{1}{2}\,||w||^2 - \sum_{i=1}^{m} \alpha_i [y_i(w^T Log_p x_i + b) - 1], \\ &= \frac{1}{2}\langle w, w\rangle_p - \langle w, w\rangle_p - b\sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i, \\ &= -\frac{1}{2}\langle w, w\rangle_p + \sum_{i=1}^{m} \alpha_i \qquad \text{where} \sum_{i=1}^{m} \alpha_i y_i = 0,\end{aligned}$$

where $w = \sum_{i=1}^{l} \alpha_i y_i Log_p x_i$ and $\left|\left|Log_p x_i\right|\right| = d(p, x_i)$. The partial derivative of $\mathcal{L}$ with respect to p is found as below

$$\begin{aligned}\bigtriangledown_p \mathcal{L} &= \frac{\partial \mathcal{L}}{\partial p}, \\ &= \frac{\partial}{\partial p}\left(-\frac{1}{2}\langle w, w\rangle_p + \sum_{i=1}^{m} \alpha_i\right), \\ &= -\sum_{i=1}^{l} \alpha_i y_i Log_p x_i, \\ &= -w.\end{aligned}$$

From the above derivation, we can say that to minimize our optimization function, we need to follow in the negative gradient direction. Here, representation of hyperplane is *not* unique, the separating hyperplane $h_{(w,b)} = 0$ can also be given as $h_{(-w,-b)} = 0$.

$$h_{(w,b)} = w^T x + b = 0$$

where x is a point on the hyperplane. Above equation can be also given by

$$\begin{aligned}-(w^T x + b) &= 0, \\ (-w)^T x + (-b) &= 0, \\ h_{(-w,-b)} &= 0,\end{aligned}$$

The direction of w obtained by SVM algorithm in MATLAB is decided by the order of labels. If we interchange the labels then the direction will change accordingly. Therefore, to compute the point which maximizes the margin at the time of practical implementation, we need to search in both direction w and $-w$, find a point which gives maximum separating margin.

Box 4.4: Algorithm to find a point with maximum margin

Input: Data x_i with labels $y_i \in \{-1, 1\}$, for $i = 1, 2..., m$.
Output: Proposed point p_{old}
Initialization:Base point p;
Do
$x_i' = Log_p(x_i) \in T_pM$, for $i = 1, ..., m$
Apply SVM on (x_i', y_i), for $i = 1, ..., m$
Compute w and margin($\triangle$)
$\tau = min(1, \triangle)/10$
$\hat{w} = w/\,||w||$
$p_{new} = Exp_p(\tau\hat{w})$
$p_{old} = p$ and $p = p_{new}$
While $\nabla_{p_{new}}\mathcal{L} > \nabla_{p_{old}}\mathcal{L}$ (4.6)

Here, first we need to project the data on the tangent space of the base point p. Now apply SVM on projected data and find the w, α, b parameters. To achieve higher separating margin, we need to find an intermediate base point which is in the direction of w or $-w$. To handle the convergence issue, we need to set controlling parameter τ. After applying our algorithm for both the direction w and $-w$, we get lower margin compared to the margin at the base point in the direction of gradient descent while maximum margin in the negative gradient descent direction. The local margin maximizer point obtained by our algorithm is chosen as the proposed point.

4.4 Experiments and Application

For given two data clusters on manifold, we linearly classify data using SVM and compare the separating margin on the tangent space of three different points. For this purpose, we choose (a) a random point which is in the neighbourhood of given data clusters, (b) the base point (overall geodesic mean) and (c) the point computed by our algorithm.

Here, we experimented with different types of manifolds like surface of a sphere S^2 and some subgroups of lie group like rotation matrices $SO(3)$ and symmetric positive definite matrices $PD(2)$. We also give an application of texture classification using manifold statistics. For texture classification we use Brodatz texture dataset.

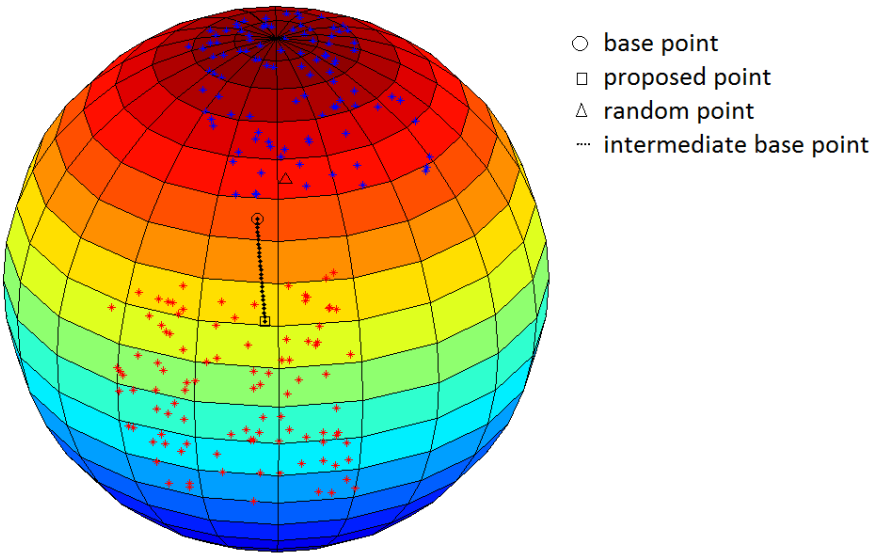


Figure 4.4: Data clusters on s^2 manifold.

In figure (4.4) the base point, the proposed point and a random point are represented by a triangle, a circle and a square, respectively. Dotted points indicates the intermediate base points as our approach iteratively finds a margin maximizing point. From table (4.1) we get improved margin at the proposed point than the base point as it is a local margin maximizer.

For the data shown in figure (4.4), we also compare the data on the tangent space of the base point and it's *antipodal* point. From figures (4.5,4.6), we get 0.1551

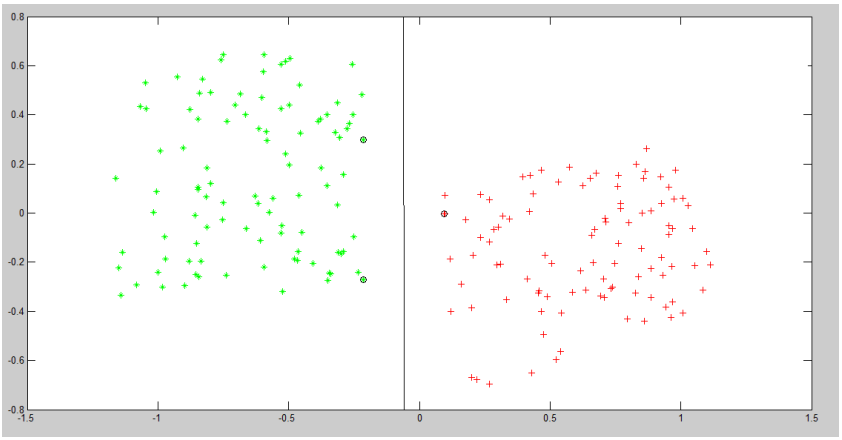


Figure 4.5: Margin of 0.1551 at the base point.

margin the tangent space of the base point and 1.2125 margin on the tangent space of an antipodal point of the base point.

Here, antipodal point of the base point gives higher margin than the base

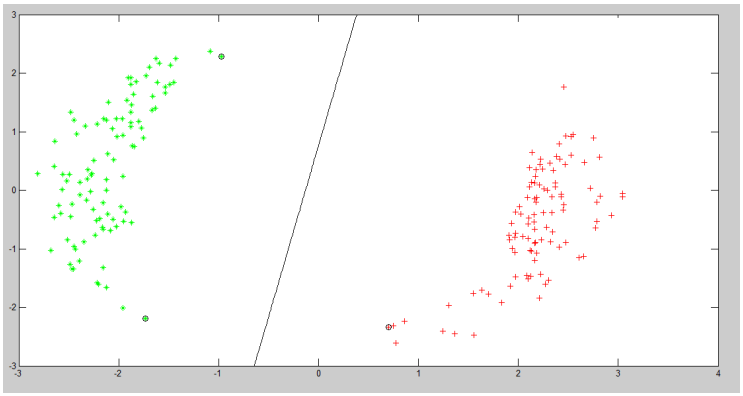


Figure 4.6: Margin of 1.2125 at the antipodal point.

point. As our algorithm finds a local margin maximizer point, we may get higher margin at some random point (i.e., random point selected around antipodal point) than the base point or the proposed point.

Table 4.1: Example on S^2 manifold

	Point	w	Margin
Random point	$(0.6267, -0.1720, 0.7600)$	$(6.4942, -0.7942)$	0.1528
Base point	$(0.6955, -0.3051, 0.6506)$	$(6.4461, 0.0195)$	0.1551
Proposed point	$(0.8807, -0.3361, 0.3339)$	$(6.2451, 0.0359)$	0.1601

For given data on S^2 manifold, margin comparison at described points is given

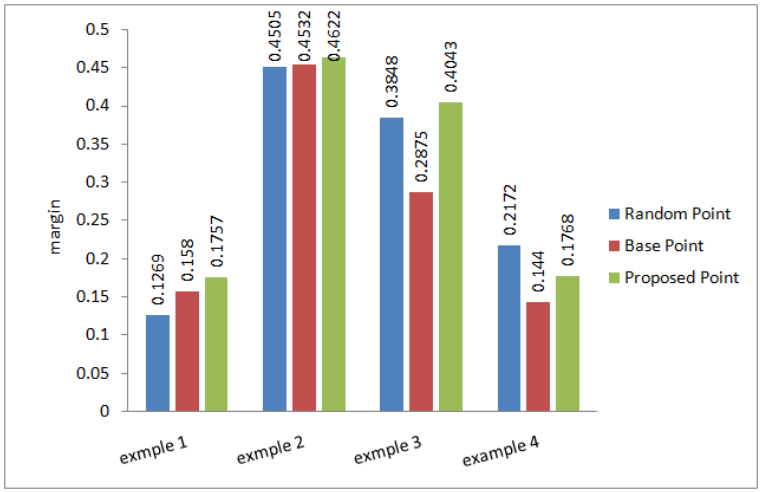


Figure 4.7: Margin comparison on S^2 manifold.

in Table (4.1). Experimental results on data clusters with different number of points (i.e., 70, 100, 150 points per class) and variance is also given in figure (4.7).

As shown in figure (4.7), we get higher margin at the proposed point than the

base point and a random point in *example* 1 and *example* 2. As discussed earlier, we get higher margin at a random point than the base point in *example* 3 and *example* 4.

To classify data on $SO(3)$ and $PD(2)$ manifolds (or Lie groups) as discussed in last chapter, first we project the data on the tangent space of a point. Then, we project the data on the tangent space of the Identity from the tangent space of a given point using Lie group action. Lastly, we compute an equivalent (distance preserving) vector representation in $\mathbb{R}^3$. Using them experimental results on $SO(3)$ and $PD(2)$ manifold are given below.

Table 4.2: Example on $SO(3)$ manifold

	Point	w	Margin
Random point	$\begin{bmatrix} 0.7749 & -0.6317 & 0.0377 \\ 0.6324 & 0.7751 & 0.0141 \\ 0.0218 & 0.0120 & 1.0006 \end{bmatrix}$	$(-1.9760,\ -0.6790,\ 2.0991)$	0.3376
Base point	$\begin{bmatrix} 0.7228 & -0.6470 & 0.2446 \\ 0.6874 & 0.7132 & -0.1412 \\ -0.0234 & 0.2657 & 0.9647 \end{bmatrix}$	$(-2.0377,\ -0.3568,\ 2.0720)$	0.3415
Proposed point	$\begin{bmatrix} 0.8696 & -0.1190 & 0.4801 \\ 0.3586 & 0.8218 & -0.4439 \\ -0.2869 & 0.5821 & 0.7620 \end{bmatrix}$	$(-1.9545,\ -0.3436,\ 2.0080)$	0.3542

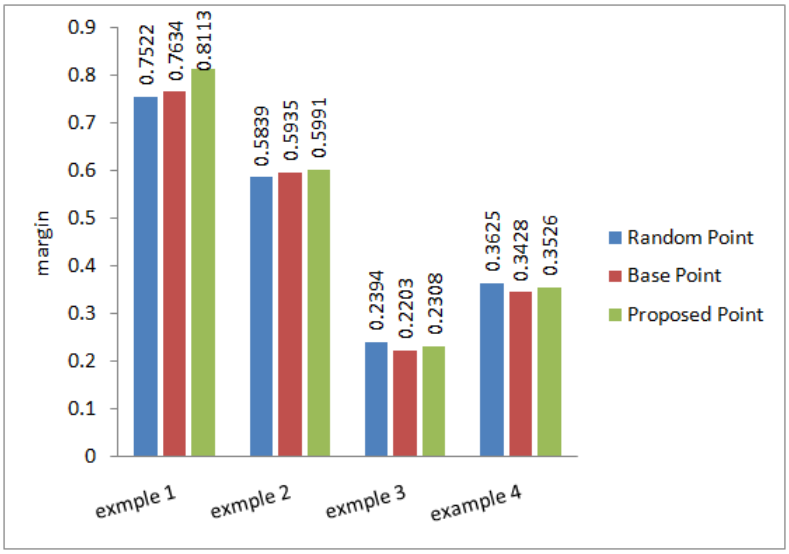


Figure 4.8: Margin comparison on $SO(3)$ manifold.

From table (4.2), we see that in case of $SO(3)$ manifold, we get significant improvement in margin at the proposed point compared to the base point using our algorithm. As shown in figure (4.8), we get higher margin at the proposed point

than the base point and a random point in all examples. We get higher margin at some random point compared to the base point in example 3 and 4.

Table 4.3: Example on $PD(2)$ manifold

	Point	w	Margin
Random Point	$\begin{bmatrix} 1.8859 & 0.1560 \\ 0.1560 & 1.0419 \end{bmatrix}$	$(-0.5458,\ 0.4662,\ -1.6752)$	0.5486
Base Point	$\begin{bmatrix} 2.5507 & 0.0320 \\ 0.0320 & 2.3437 \end{bmatrix}$	$(-0.5476,\ 0.4697,\ -1.6738)$	0.5486
Proposed Point	$\begin{bmatrix} 2.3601 & 0.1276 \\ 0.1276 & 1.8422 \end{bmatrix}$	$(-0.5469,\ 0.4680,\ -1.6745)$	0.5486

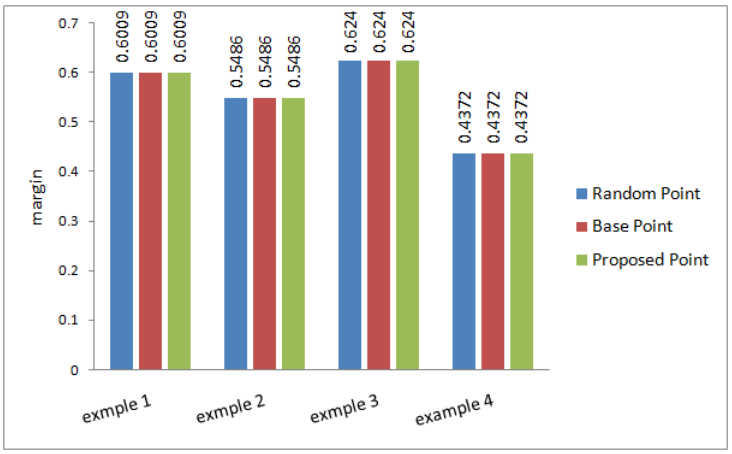


Figure 4.9: Margin comparison on $PD(2)$ manifold.

From table (4.3) and figure (4.9), in case of $PD(2)$ manifold for a given data we get same separating margin at any point.

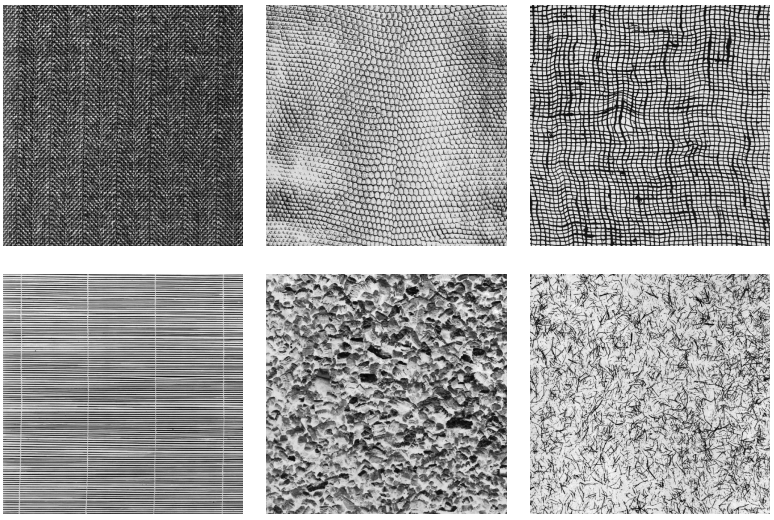
In case of $PD(n)$ manifold the cut locus is infinite or in other words it is a *complete*. [1] We can cover entire $PD(n)$ manifold using a single chart. The mapping from manifold to Euclidean space is done using a single chart, relative distance between two points on any tangent space remains same. So, we get same separating margin on the tangent space of every point.

As a computer vision application, we try to classify texture using manifold statistics. For classification purpose we take Brodats texture dataset. To classify texture, we first compute covariance matrices from patches of the texture image.

Here, we took $n \times n$ patches in the specified neighbourhood and found out covariance matrix of $m \times m$, which is by definition $PD(m)$ manifold where $m = n^2$. We tried to apply our classification approach for such data. As we have seen from table (4.3) in case of positive definite matrices separating margin is coming

[1] A manifold with the geodesic $\gamma(t)$ is defined for $-\infty < t < +\infty$ is called complete.

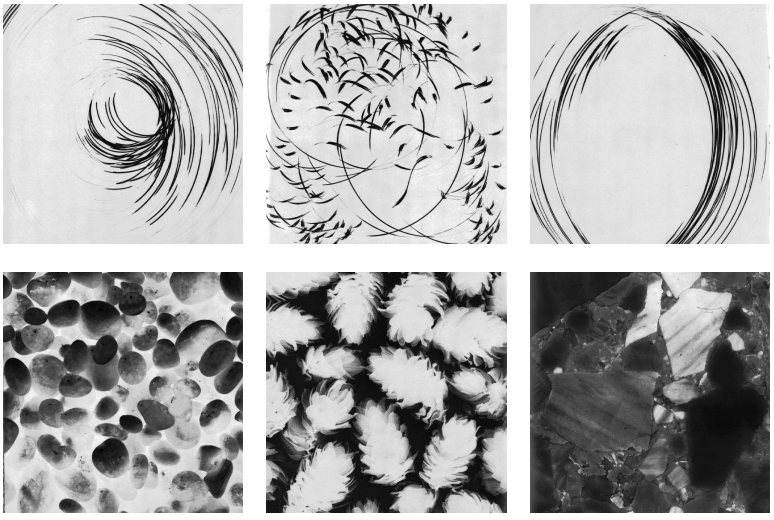
same on the the tangent space of any point, we apply SVM to the data projected on the base point.



(a) (b) (c)

Figure 4.10: Classified textures.

We classify textures shown in figure 4.10 successfully using $PD(16)$ manifold. Here, each image is of size 640×640. We took all possible 4×4 patches in the neighbourhood of 12 and compute the covariance matrix. Similarly, we compute 50 covariance matrices (data points) from the texture images. As shown in above example, we tried to classify textures that are of same size. We also show some misclassified textures in figure 4.11.



(a) (b) (c)

Figure 4.11: misclassified textures.

CHAPTER 5
Discussion and Future Work

This chapter reviews and discusses the contributions of this dissertation in the first section followed by possible future work.

5.1 Discussion

We use the idea of tangent space SVM by applying the usual SVM on the tangent sapce of manifold for classification of data. To generalise the idea of SVM for data on manifolds, we try to explore Euclidean SVM on multiple tangent planes. To apply SVM, we first project the data on a tangent space of a point on the manifold. Therefore, we need to choose a point on the manifold carefully on whose tangent space we are projecting data.

We propose an algorithm to find the point on whose tangent space we can classify data with larger margin. We apply the proposed algorithm for data on S^2, $SO(3)$ and $PD(2)$ manifolds and compare the results of separating margin on the tangent space of various points. We got improvement in separating margin using our algorithm. The proposed point obtained by our algorithm, provides the best tangent space for classification as it gives a local margin maximizer point. We also give an application of texture classification using manifold learning.

5.2 Future work

Here, we find the proposed point under the assumption that data is *linearly* separable on the tangent space of the base point. We want to extend our algorithm in case if data is *not* linearly separable on the tangent space of the base point.

To classify High Dimensional Low Sample Size (HDLSS) data, Distance Weighted Discrimination (DWD) method proved to be better than usual SVM. The DWD method is developed from usual SVM using modern computationally intensive optimization techniques. In medical images, *m-rep* is an example of HDLSS data

where a small population of 3-D shape is represented by vectors of many parameters. For classification purpose, we want to generalise DWD method for data on *m-rep* $\{\mathbb{R}^3 \times \mathbb{R} \times SO(3) \times SO(2)\}$ manifold using our proposed algorithm.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification and scene analysis. Part 1, Pattern classification*. Wiley, 2000.

[2] V. N. Vapnik, *Statistical learning theory*, ser. Adaptive and learning systems for signal processing, communications, and control. Wiley, 1998.

[3] P. Fletcher, C. Lu, S. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 995–1005, 2004.

[4] P. Fletcher and S. Joshi, "Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors," in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, vol. 3117. Springer Berlin Heidelberg, 2004, pp. 87–98.

[5] P. T. Fletcher, C. Lu, and S. Joshi, "Statistics of shape via principal geodesic analysis on lie groups," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2003, pp. I–95.

[6] P. T. Fletcher and S. Joshi, "Riemannian geometry for the statistical analysis of diffusion tensor data," *Signal Processing*, vol. 87, no. 2, pp. 250–262, 2007.

[7] S. K. Sen, M. Foskey, J. S. Marron, and M. A. Styner, "Support vector machine for data on manifolds: An application to image analysis," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008, pp. 1195–1198.

[8] S. Sen and T. U. of North Carolina at Chapel Hill. Statistics, *Classification on Manifolds*. The University of North Carolina at Chapel Hill, 2008.

[9] Q. Rentmeesters and P. Absil, "Algorithm comparison for karcher mean computation of rotation matrices and diffusion tensors," in *EUSIPCO*, vol. 2011, 2011, p. 19th.

[10] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *In Proc. 9th European Conf. on Computer Vision*, 2006, pp. 589–600.

[11] W. M. Boothby, *An introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986, vol. 120.

[12] J. M. Lee, *Introduction to Topological Manifolds*. Springer, 2000, vol. 202.

[13] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[14] G. Simmons, *Introduction to Topology and Modern Analysis*, ser. International series in pure and applied mathematics. Krieger Publishing Company, 1963.

[15] J. M. Lee, *Introduction to Smooth Manifolds*. Springer, 2012, vol. 218.

[16] N. V. Huan and D. N. Toan, "Vector fields in expressing hairstyles," in *2010 2nd International Conference on Advanced Computer Control (ICACC)*, vol. 2, 2010, pp. 537–541.

[17] S. Kobayashi and K. Nomizu, *Foundations of Differential Geometry*, ser. A Wiley Publication in Applied Statistics. Wiley, 1963, no. v. 1.

[18] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. Wiley-Interscience, 1998.

[19] A. Ng, "Support vector machine," Machine Learning Lecture Notes. [Online]. Available: http://see.stanford.edu/materials/aimlcs229/cs229-notes3.pdf

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*, ser. Berichte über verteilte messysteme. Cambridge University Press, 2004.

[21] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical analysis on manifolds and its applications to video analysis," in *Video Search and Mining*. Springer, 2010, pp. 115–144.

CHAPTER A
Topology

In the theory of manifolds one of the most fundamental building block is topology. Topology means to study properties of an object which are preserved under *continuous* deformations (small changes in input results in small change in output). To generalise the idea of *continuity* from Euclidean space to more general space the study of a topological spaces is required. Recent applications include computer graphics, computer-aided design (CAD), and structural biology, all of which involve understanding the intrinsic shape of some abstract space.

A.1 Topological Spaces

Given a set X and a family of its subsets $\mathcal{T}$, if
(1) $\emptyset, X \in \mathcal{T}$
(2) $\bigcup_{i=1}^{\infty} \mathcal{T}_i \in \mathcal{T}$
(3) $\bigcap_{i=1}^{n} \mathcal{T}_i \in \mathcal{T}$

then it is a topology on X and the pair $(X, \mathcal{T})$ is called a topological space[12]. The elements of $\mathcal{T}$ are called *open sets*. Like subset of $\mathbb{R}^n$ is a collection of *open ball* [12] of the form $\mathcal{B}(x, r) = \{x' \in \mathbb{R}^n : \parallel x - x' \parallel < r\}$. A set $Y \in X$ is a *closed set* if its complement is an open set. A set can be both open and closed also neither open nor closed, i.e., $\emptyset$, X are both open and closed.

For the topology on a set X consist of only $\emptyset$ and X, which are open set is called the *trivial topology*. If all the subset of X are open then the $\mathcal{T}$ is called the *discrete topology* on X.

Topological Basis

A collection of open set $\{U_i\}$ is said to be a basis of $(X, \mathcal{T})$ if every open subset of X can be written as union of elements of $\{U_i\}$. If the topological space has countable number of basis the it is said to be a *Second Countable*.

Basis in linear algebra ((a) spanning the vector space (b) linearly independent) can be given as a special case of topological basis. The difference arises from the fact that, this expression for the union need not be unique.

A.2 Metric Space

We can define the topology on $\mathbb{R}^n$ by the Euclidean distance between points. Similarly it can be defined to any general space where distance is defined.

A metric space [12] is a ordered pair (X, d) consisting of a set X with a function $d : X \times X \to \mathbb{R}$ for all $x_1, x_2, x_3 \in X$ satisfies
(1) $d(x_1, x_2) \geq 0$, and $d(x_1, x_2) = 0$ if and only if $x_1 = x_2$
(2) $d(x_1, x_2) = d(x_2, x_1)$
(3) $d(x_1, x_2) + d(x_2, x_3) \geq d(x_1, x_3)$

The function d is called a *metric* or a *distance function*. A metric d on a set X is called *complete* if every Cauchy sequence converges in X. A Cauchy sequence is a sequence $x_1, x_2, ... \in X$ such that for any $\delta > 0$ there exists an integer N such that $d(x_i, x_j) < \delta$ for all $i, j > N$.

A.3 Continuity

As stated earlier topology is required to generalize notion of *continuity* on Euclidean space. Continuity between two topological spaces is given as:

Let X and Y be topological spaces. A mapping $f : X \to Y$ is continuous if for each open set $U \in Y$, the set $f^{-1}(U)$ is open in X. Also a mapping $f : X \to Y$ is a *homeomorphism* if it is one-one and onto (bijective) and both f and f^{-1} are continuous.

A.4 Hausdorff Space

A topological space X is said to be Hausdorff if for any two distinct points $x_1, x_2 \in X$ we can find a open neighbourhood U_1 of x_1 and U_2 of x_2 such that U_1 and U_2 are disjoint ($U_1 \cap U_2 = \emptyset$). In another words, a Hausdorff space is a topological space with a separation property.

Every metric space is a Hausdorff space. Given a metric space X we have two distinct points $x_1, x_2 \in X$ such that $d(x_1, x_2) > 0$. Then, we can find two open ball $\mathcal{B}(x_1, r)$ and $\mathcal{B}(x_2, r)$, where $r < d(x_1, x_2)/2$ such that they are disjoint and containing x_1 and x_2 respectively [12].

Topological properties (e.g., compactness, connectedness) are useful in extensions of qualitative analysis beyond finite-dimensional Euclidean space (e.g., manifolds, functional analysis, calculus of variations). In physics, to understand the behaviour of the material, we need to study Lie groups are an example of topological groups as Lie groups are important in learning the standard model of particle physics.

CHAPTER B
More on Manifold

In this Appendix, we discuss about operations on manifold like charts and a tangent space on a manifold where chart is given as a structure preserving mapping (i.e., diffeomorphic) from a subset of a manifold to Euclidean space. We give brief introduction on vector field and covariant derivative on a manifold [11].

B.1 Charts on Manifold

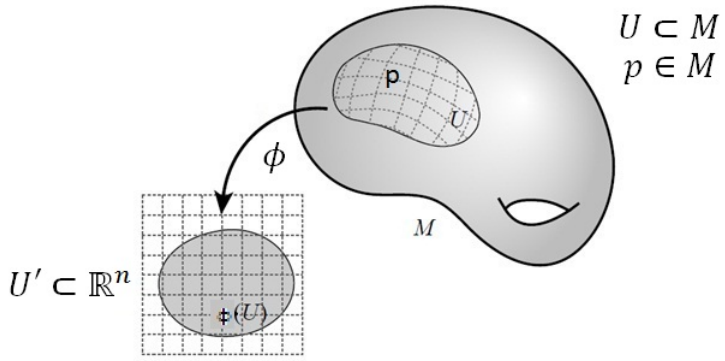


Figure B.1: Local co-ordinate system on a manifold.

As mentioned earlier that a manifold locally looks like an Euclidean space. So, local mapping from a subset of a manifold to an Euclidean space is done by using the idea of a chart [15].

A chart of a topological n-manifold is a map such that any point $p \in M$ has neighbourhood U which is *diffeomorphic* to U' in $\mathbb{R}^n$.

$$\phi_i : U_i \in M \rightarrow U_i' \in \mathbb{R}$$

We may require more than one chart to describe an entire manifold. The collection of charts $\{ U_i, \phi_i \}$, covers entire manifold is called an *atlas*. An atlas of a

manifold may not be unique, as we can cover it by multiple way using different set of charts.

The sphere is a 2-D manifold, each chart will map an open subset of the sphere (manifold) to an open subset of $\mathbb{R}^2$. Here, surface of the sphere as a subset of $\mathbb{R}^3$ is given by

$$S = \{(x, y, z) \in \mathbb{R}^3 | x^2 + y^2 + z^2 = r^2\}.$$

Let us define chart ($\phi : S^2 \to \mathbb{R}^2$) for northern hemisphere as

$$\phi(x, y, z) = (x, y) \tag{B.1}$$

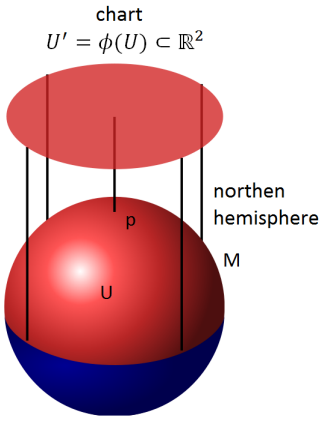


Figure B.2: Chart on sphere.

Similarly, we can define an atlas of total six charts (top, bottom and four sides) which covers the entire sphere.

B.2 Tangent Space

The tangent space [11, 15] of a manifold allows the generalization of vectors from Euclidean spaces to general manifolds. Tangent space at a point p on an n-dimensional manifold is an n-dimensional hyperplane in $\mathbb{R}^n$ that best approximates M around p with p as an origin of hyperplane. To define tangent space, first we need to define directional derivative.

Directional Derivative

The *directional* derivative of a multivariate differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ where $f(x) = f(x_1, x_2, ..., x_n)$ denotes the value at $x = (x_1, x_2, ..., x_n)$, along a given

directional vector $u \in \mathbb{R}^n$ at a given point p gives the instantaneous rate of change of the function through p in the direction of u.

$$\nabla_u f(x) = \lim_{h \to 0^+} \frac{f(x+hu) - f(X)}{h} = \nabla f(x) \frac{u}{\|u\|} = \sum_{i=1}^{k} \left(\frac{\partial f}{\partial x_i}\bigg|_{x_i = p} \right) \frac{u}{\|u\|} \tag{B.2}$$

From that we can generalize the notion of a partial derivative where directional derivation is taken in the directive *parallel* to any of the coordinate axes. The set of all possible direction vectors u that can be used in this construction forms a n-dimensional vector space that happens to be the tangent space $T_p(\mathbb{R}^n)$.

Now to define a tangent vector at $p \in M$, let $\gamma : (-\varepsilon, \varepsilon) \to M$ be smooth curve where $\gamma(0) = p$ then for any smooth function $f : M \to \mathbb{R}$, we can have the composition $f \circ \gamma$ which is also smooth so the derivative of $f \circ \gamma$ exist:

$$\frac{d}{dt}(f \circ \gamma)(0)$$

Let $\gamma 1$ and $\gamma 2$ be two smooth curve passing through $p \in M$ then the *equivalence relation* $\sim$ is between $\gamma 1$ and $\gamma 2$ is given as $\gamma 1 \sim \gamma 2$ if

$$\frac{d}{dt}(f \circ \gamma 1)(0) = \frac{d}{dt}(f \circ \gamma 2)(0) \tag{B.3}$$

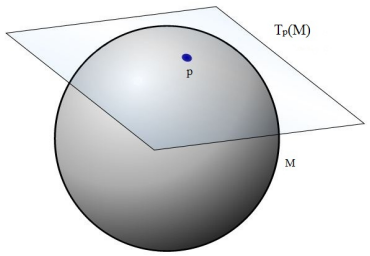


Figure B.3: Tangent space.

A tangent vector X_p can be defined as one of the equivalence classes of curves. In addition these equivalence classes form a vector space like the tangent space T_pM, which is having the same dimension as M. Tangent space at a point $p \in M$ is defined as the space of all equivalence classes $[\gamma]$ of curves γ through p. A tangent vector X_p, as the *equivalence class* of curves, is given by

$$X_p := [\gamma] = \{\tilde{\gamma} : \tilde{\gamma}(0) = \gamma(0) = p, (f \circ \tilde{\gamma})'(0) = (f \circ \gamma)'(0)\} \tag{B.4}$$

which is an element of T_pM. Collection of all tangent space over the manifold

is called the *Tangent Bundle* [11]. Now, as we have defined tangent space of a manifold, we can define vector fields on manifold.

B.3 Vector Field

An idea of a vector field is to assign a vector to each point of a manifold belonging to its tangent space [11, 16]. Vector fields are often used to model, for example, the speed and direction of a moving particle in form of a velocity vector throughout space. It given as a continuous map $Z : M \to TM$ such that for each $p \in M$, $Z_p \in T_pM$. Let x^i for $i = 1, ..., n$ be local coordinates on open set $U \subset M$ then Z at $p \in U$ for some numbers $Z^1(p), ..., Z^n(p)$ is given by

$$Z_p = Z^i(p) \frac{\partial}{\partial x^i}\bigg|_{x=p}$$

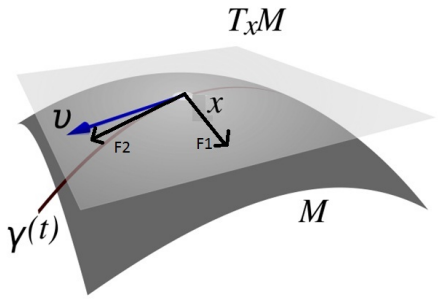


Figure B.4: Velocity vector on a tangent space.

As shown in figure (B.4), $v \in T_xM$ is a vector field where $F1$ and $F2$ are coordinate frame.

B.4 Frame

In linear algebra, a basis of a vector space V can be given as, for any $x \in V$

$$x = \sum_{i=0}^{n} a_i e_i \tag{B.5}$$

where $\{ e_i \}$ is a set of basis which are (1)linearly independent and (2)span vector space V

Frame is a generalization of basis which may be linearly dependent. Let $\{F_i\}$ be a set of frame such that each $F_i \in V$ and we want to express any $y \in V$ as linear

combination of $\{F_i\}$ as

$$y = \sum_{i=0}^{n} a_i F_i$$

For manifold, there is a concept of *moving frame*, which is a generalization of ordered basis in vector space used to study differential geometry of smooth manifold *embedded* in Euclidean space.

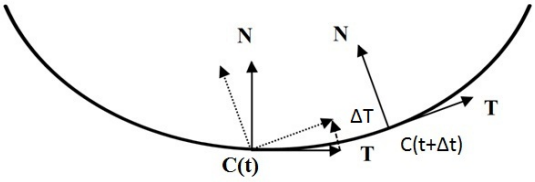


Figure B.5: Moving frame.

An idea of moving frame on 1-D manifold embedded in $\mathbb{R}^2$ is shown in figure (B.5). Here T and N vectors at a point of a curve on a plane, a translated version of the second frame is shown by dotted line and the change in T is shown by $\triangle T$.

The *Tangent Bundle* is the most commonly encountered case of moving frame of a manifold. The moving tangent frame on manifold M with vector field $X_1, X_2, ..., X_n$ form the basis of tangent space at each point on a manifold M.

B.5 Covariant Derivative

It is used to denote derivative along tangent space on manifold M embedded in R^n. For $p \in M$

$$T_p R^n = T_p M \oplus T_p^{\perp} M \tag{B.6}$$

where $T_p^{\perp} M$ is orthogonal complement of $T_p M$. As can be seen from figure (B.6)

We can define the following orthogonal projections

$$\Pi' : T_p \mathbb{R}^n \to T_p M$$

$$\Pi'' : T_p \mathbb{R}^n \to T_p^{\perp} M$$

such that

$$Z(t) = Z'(t) + Z''(t) \tag{B.7}$$

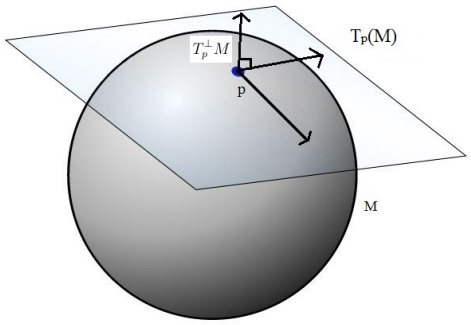


Figure B.6: T_pM and $T_p^{\perp}M$.

where $Z'(t) = \Pi'(Z(t)) \in T_pM$ and $z''(t) = \pi''(Z(t)) \in T_p^{\perp}M$. From equation (B.7), we can write

$$\dot{Z}(t) = \Pi'\left(\dot{Z}(t)\right) + \Pi''\left(\dot{Z}(t)\right)$$

$\pi'(\dot{Z}(t))$ is called tangent vector field on manifold M and

$$\frac{DZ}{dt} = \Pi'\left(\frac{\partial Z}{\partial t}\right) \tag{B.8}$$

where $\frac{D}{dt}$ is called covariant derivative, which is used to compute tangential component of the derivative of a tangential vector field [17].

Expression of Covariant Derivative

Let M be a n-dimension manifold embedded in $\mathbb{R}^m$. Let $p \in M$ and U be a neighbourhood of p. Let (ϕ, U) be a chart on M as shown in figure (B.7)

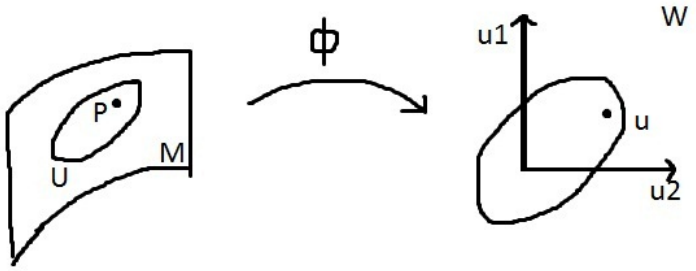


Figure B.7: Local mapping.

$$\phi^{-1}(u) = (g^1(u), g^2(u), ..., g^m(u)) \text{ and } u \in W \subset \mathbb{R}^n$$

where $g^1, g^2, ..., g^m$ are mapping function from local coordinate to manifold. Let $F_1, F_2, ..., F_n$ be orthogonal coordinate frame (i.e., basis for the vector space) on manifold and $b^1, b^2, ..., b^n$ be coefficient then

$$\begin{aligned} Z(t) &= \sum_{i=1}^{n} b^i(t) F_i(t) \\ \frac{d}{dt}\left(Z(t)\right) &= \frac{d}{dt}\left(\sum_{i=1}^{n} b^i(t) F_i(t)\right) \\ &= \sum_{i=1}^{n} \frac{db^i(t)}{dt} F_i(t) + b^i(t)\left[\frac{d}{dt} F_i(t)\right] \end{aligned}$$

So, covariant derivative is given by

$$\frac{D}{dt}\left(Z(t)\right) = \sum_{i=1}^{n} \frac{db^i(t)}{dt} F_i(t) + b^i(t)\Pi'\left[\frac{d}{dt} F_i(t)\right]$$

where

$$\begin{aligned} \frac{D}{dt} F_i(t) &= \sum_{j,k=1}^{n} \Gamma_{ij}^k(u) \frac{\partial u^j}{\partial t} F_k \\ \Gamma_{ij}^k(u) &= \sum_{\alpha=1}^{n} \frac{\partial^2 g^\alpha}{\partial u^i \partial u^j} a_\alpha^k(u) \quad \text{and} \\ \Pi'\left(\frac{\partial}{\partial x^\alpha}\right)\Big|_p &= \sum_{k=1}^{n} a_\alpha^k(u) F_k \end{aligned} \tag{B.9}$$

finally

$$\frac{D}{dt} Z(t) = \sum_{k=1}^{n} \left[\frac{db^k}{dt} + \left(\sum_{i,j=1}^{n} \Gamma_{i,j}^k(u) b^i \frac{\partial u^i}{\partial t}\right)\right] F_k(t) \tag{B.10}$$

where $\Gamma_{i,j}^k$ is called *Christoffel symbol* [17] which gives the k^{th} component for the covariant derivative of the i^{th} basis along the coordinate curve u^j. Let $x^1, x^2, ..., x^n$ be a local coordinate system in M. Set $F_i = \partial/\partial x^i$, $i = 1, 2, ..., n$. Then the component $\Gamma_{i,j}^k$ with respect to $x^1, x^2, ..., x^n$ is given by:

$$\nabla_{F_i} F_j = \sum_k \Gamma_{i,j}^k F_k \tag{B.11}$$