Bachelor Mathematics

Track: Data Science

Bachelor Thesis

Hyperspectral Image Classification for Absestos with Supervised and Semi-supervised learning

by

Anh Van Giang

July 5, 2022

Supervisor: prof. Joost Hulshof

Department of Mathematics
Faculty of Sciences
VU UNIVERSITY AMSTERDAM

Abstract

Hyperspectral image classification is an image classification/segmentation task that uses picture incorporating not only spatial but also spectral information of the various materials captured in the image. This paper will explain (some of) the theories behind the usual steps in an image classification task, such as preprocessing, modelling with three models. The main models that will be implemented is HybridSN, a deep learning model, Support Vector Machine, and Laplacian SVM, a semi-supervised learning method that is capable of regularizing the hypothesis function on the underlying geometry. The results show that HybridSN is better than SVM and LapSVM. The implementation of LapSVM code-wise is still rudimentary and in its basic form, thus lead to an underwhelming performance. The trained data are all windowed to minimize class-imbalance and if the size of the windows are increased, performance will drastically reduce. Overall, using a window of 30, HybridSN performs better than its peer at detecting asbestos vs non-asbestos pixels.

Title: Hyperspectral Image Classification for Absestos with Supervised and Semi-supervised learning
Author: Anh Van Giang, vangianganh@gmail.com, 2658762
Supervisor: prof. Joost Hulshof
Date: July 5, 2022

Department of Mathematics
VU University Amsterdam
de Boelelaan 1081, 1081 HV Amsterdam
http://www.math.vu.nl/

Contents

1 Introduction

Asbestos was a prominent building material in the 20th century but later, as it was known to cause mesothelioma [1], it was banned and had most of it replaced by the Dutch government [2]. Nevertheless, according to the database of hyperspectral images of the city of Drenthe given by the Dutch government and annotated by Heuff *et al* [3].

The data that is used in this paper are aerial hyperspectral images. These are images taken by a specialized equipment by airplanes, and are usually composed of hundreds of bands or features unlike ordinary images. These bands contain hyperspectral signature reflecting the chemical nature of the materials being captured. In this paper, the data has 420 bands, capturing the visible, near-infrared light, and part of the shortwave infrared light. The sensor used has a spectral range up to 2500 nm.

Image classification, segmentation using hyperspectral data have been studied extensively using different methods ranging from deep learning [4] [5] [6] to chemometric approach [7], usually on public datasets such as the Indian Pines [8] that does not contain asbestos classes. For asbestos-related studies, Krówczyńska *et al* [9] performed a study using Spectral Angle Mapper (SAM) to achieve a result of 61.54% correct classification of the asbestos roofs and 97.98% of the asbestos free roof. Closer to home, Ubels [2], as an extension of Heuff *et al* [3], used two deep learning models to obtain the highest mean intersection over union (or the Jaccard score) of 0.41.

In this paper, two main machine learning methods, namely supervised and semi-supervised learning, using Support Vector Machine (SVM), Hybrid Spectral Net (HybridSN), and Laplacian Support Vector Machine (LapSVM) will be applied to the task of classifying hyperspectral roofs. Furthermore, the relevant mathematics regarding dimensionality reduction methods such as Principal Component Analysis (PCA) will be explained to the extent permitted by a Bachelor thesis. The main motivation for choosing HybridSN is that it is a deep learning model, like many of its kind, is capable of learning complex representations of the data and is able to perform very well on a variety of different tasks. This particular model is documented to yield extremely good result on hyperspectral data-related tasks. The semi-supervised learning method, LapSVM, is chosen because of its ability to leverage the underlying geometry of the data, assuming the manifold hypothesis, and unlabeled points to learn the hypothesis function. This method is supported by rigorous mathematics and is reported to perform quite well on hyperspectral data [10].

Unlike some other papers, we will not delve to deep into the intrinsic properties of the hyperspectral images themselves, but instead will treat it like any other dataset in an Euclidean space with the assumption that for any two points, if their distance with respect to a metric is close, then the probability of them belonging to the same class is high. Furthermore, we will be classifying the two classes: asbestos and non-asbestos, with the latter not-necessarily being a roof.

The research question in this paper is: How well does semi-supervised learning method perform compared to traditional supervised learning ones in the context of asbestos hyperspectral image classification ?

Section 2 and 3 will discuss some of the theories behind dimensionlity reduction methods and neural network respectively. Section 4-7 will explain (some of the) necessary theories behind LapSVM and Section 8 will go into its derivation. The results and experimentation will be shown in Section 9 with conclusion and further research in Section 10 and 11 respectively.

2 Dimensionality Reduction

2.1 Principal Component Analysis (PCA)

Given a dataset matrix X of dimension $m \times n$ where m is the number of features and n the number of data points. Ideally, we would want each column or feature of the data to be independent from each other, otherwise it would be highly redundant and inefficient because of the limited computer resources. Hence, it is desirable to transform the data in such a way that the majority of the information are retained while ensuring independence. Denote the linear transformation matrix as P, we would want to find an orthogonal P that satisfies

$$PX = Y$$

with Y the independent re-presentation of X. Essentially, we would be performing a change of basis on the standard basis of $\mathbb{R}^m$. Obviously, by letting P be a linear transformation, we are assuming that there is a linear relationship in the data.

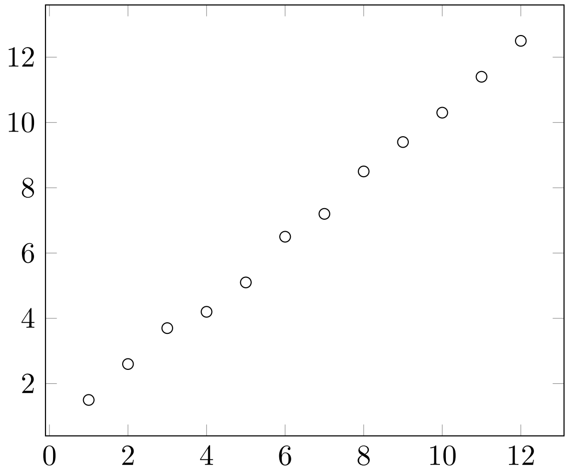


Figure 1: Linearly dependent two dimensional data

Given the clearly linearly dependent data in Figure 1, it is clear that it can be adequately represented in $\mathbb{R}$ instead of $\mathbb{R}^2$ without losing too much information. Thus, we would like to perform a change of basis with the new basis correspond to the direction of greatest variance, i.e the diagonal direction, assuming they contain the dynamic of interest. To that end, allow us to introduce the Principle Component Analysis (PCA) technique, which is heavily used in practice, that transform the standard basis into new ones, called principle components, such that linear independence is obtained.

Recall the covariance of two vectors measures the degree of the linear relationship between them. A covariance of zero indicates no linear relationship whatsoever and vice versa. Assuming zero mean, the covariance matrix of X can be given as

$$C_X = \frac{1}{n-1} X X^T,$$

where the off-diagonal values is the covariance between two features and the diagonal is the variance of a feature. Since the goal is to achieve linear independence between features, we would want the covariance matrix of Y to be diagonalized, i.e

$$C_Y = \frac{1}{n-1} Y Y^T = \frac{1}{n-1} P(XX^T)P^T = \frac{1}{n-1} PAP^T$$

where A is symmetric. Recall that if A is symmetric, using spectral theorem, it can be decomposed into

$$A = EDE^T$$

where E is the orthogonal matrix of eigenvectors as columns and D a diagonal matrix of eigenvalues. Select $P = E^T$, we have

$$A = P^T D P$$

and

$$\begin{aligned} C_Y &= \frac{1}{n-1} PAP^T \\ &= \frac{1}{n-1} P(P^T DP)P^T \\ &= \frac{1}{n-1} (PP^{-1})D(PP^{-1}) \quad (E \text{ is orthogonal}) \\ &= \frac{1}{n-1} D. \end{aligned}$$

The i-th diagonal value of C_Y is the variance of the projected X along the principal component p_i i.e the eigenvector.

Remark. In practice, to compute PCA of X usually entails making every feature of X have mean zero.

Remark. To perform dimensionality reduction by PCA, after the transformation, we usually choose the keep the first k features that correspond to the highest explained variance ratio, i.e the sum of the first k diagonal values of C_Y over its trace.

2.2 Maximum Noise Fraction (MNF)

In practice, PCA is the ubiquitous tool for dimensionality reduction across various problems, such as multi/hyperspectral image processing. However, it was shown by Green

[11] that the variance of multi/hyperspectral images did not necessarily reflect real Signal to Noise ratio (SNR), due to unequal noise variances in different bands. To deal with this problem, Green *et al* developed the Maximum Noise Fraction (MNF) transformation based on maximization of SNR. This transformation was later reinterpreted by Lee *et al* [12] by adding the noise-whitening step before PCA. We will briefly discuss PCA and ZCA whitening below.

Whitening refers to a linear transformation of a matrix such that the new covariance matrix is the identity. In a sense, it is similar to PCA transformation with an addtional step of scaling. Let X be the data set matrix and C_X its covariance matrix. Like before, we can decompose C_X into

$$C_X = EDE^T.$$

Then the PCA whitening process can be described as

$$Y_{PCA} = D^{-1/2}E^T X$$

and illustrated in the figure below.

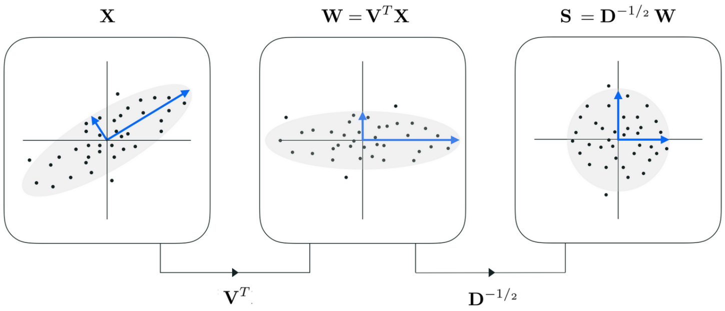

$\mathbf{X}$
$\mathbf{W} = \mathbf{V}^T\mathbf{X}$
$\mathbf{S} = \mathbf{D}^{-1/2}\,\mathbf{W}$
$\mathbf{V}^T$
$\mathbf{D}^{-1/2}$

Figure 2: PCA whitening illustrated [13].

It can be seen that the transformation did not preserve the spatial orientations of the points which can be highly undesirable when applied to some models. To remedy the problem, we can instead use ZCA whitening which is very similar to PCA but with a rotation step, it can be expressed as

$$Y_{ZCA} = ED^{-1/2}E^T X.$$

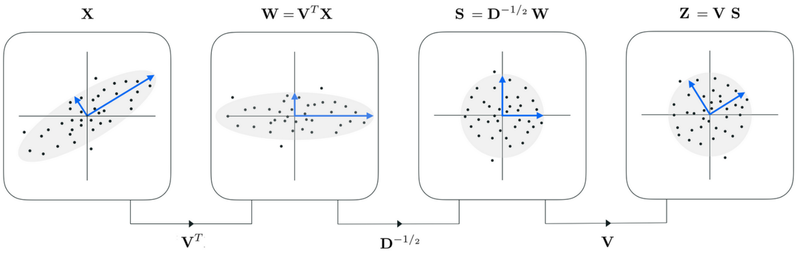

$\mathbf{X}$
$\mathbf{W} = \mathbf{V}^T\mathbf{X}$
$\mathbf{S} = \mathbf{D}^{-1/2}\,\mathbf{W}$
$\mathbf{Z} = \mathbf{V}\,\mathbf{S}$
$\mathbf{V}^T$
$\mathbf{D}^{-1/2}$
$\mathbf{V}$

Figure 3: ZCA whitening illustrated [13]

Remark. There exists a variety of other more sophisticated dimensionality reduction methods such as [14] but they are beyond the scope of this paper.

3 Neural Network and Deep Learning

3.1 Neural Network

A neural network, to put it simply, is a directed multipartite graph with every two contiguous layers forms a bipartite graph. This structure is composed of a list of layers with the first and last one called input and output layer. Every layers in between are called hidden layers because the values of these layers aren't usually shown (in practice). An example of a simple neural network is shown in Figure 4. This type of architecture can be called a feedforward neural network because the data from the input layer **feedforward** to the hidden layers and finally, the output.

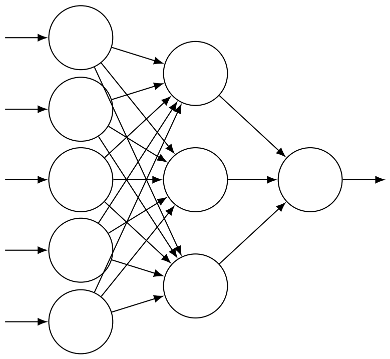


Figure 4: A simple feedforward neural network.

Each edge is associated with a weight that helps calculate the output of a node in the hidden layer as

$$h(x, w, b) = \sigma(x^T w + b)$$

where σ is an activation function, usually non-linear, and w, b the weights, biases. The non-linearity of the activation function helps the network nonlinear features thus making it extremely useful. For the network to actually learn from data, we use a loss function, such as the mean squared error, at the output layer to calculate the error from each training loop, i.e each time the data is propagated through the layers. Since the objective is to minimize the error, the network then calculate the partial derivative of the (multivariate, differentiable) loss function L w.r.t each of its parameters then update them accordingly, usually through an algorithm like gradient descent where a parameter w_i is updated as

$$w_i \leftarrow w_i - \gamma \frac{\partial L(x, w_i)}{\partial w_i}.$$

The entire process is called backpropagation because the network, in a sense, is propagating back the changes to the weights in the hidden layer to feed it forward and repeat. It has been shown that the gradient descent algorithm converges to a minimum given that the loss function is smooth, convex, etc.

One of the major shortcomings with this type of architecture is that it does not take into account spatial relationship like that of images. For example, given an arbitrary architecture similar to the one in Figure 4, if we want to input an $m \times n \times p$ matrix X, like a RBG image of a cat, then we first have to convert it to a matrix of size $mn \times p$, which clearly remove a lot of meaningful relationships of the neighborhoods of the pixels. Since our data is hyperspectral, meaning that it can be seen as images, we have to introduce a better architecture to remedy this problem.

3.2 Convolutional Neural Network (CNN)

A simple CNN, like the one in Figure 5, is a neural network with special layers such as the convolutional and pooling layer. The convolutional layer's parameters consist of a set of learnable filters. Every filter is spatially small but have the same depth as the input volume. During the forward pass, the filters are slid across the width and height of the input and compute the dot product products between the entries of the filter and the input at any position. The pooling layer operates independently of the depth of the input, usually associated with an operation like max. It applies a window with its operation over the input to reduce it size, making it more manageable for the fully connected layer (an ordinary neural network).

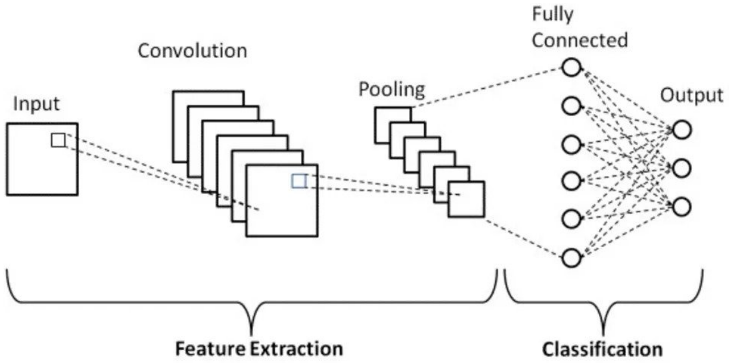


Figure 5: A simple CNN [15].

In practice, there exists a variety of different layers designed for a multitude of purposes. Interested readers can see more at [16]. The Hybrid Spectral Network (HybridSN) developed by Roy *et al* [5] is a CNN that is somewhat structurally similar to the one described in Figure 5 but its convolution layers are both three-dimensional and two-dimensional to take into account both the spectral and spatial features.

Let X be a matrix of shape 3×3 acting as an input for a simple CNN. Let F be a 2×2 filter. This filter F can act on X in the convolution layer as a window sliding through X. Call O the output of this operation, we have

$$O_{i,j} = X_{i,j}F_{11} + X_{i,j+1}F_{12} + X_{i+1,j}F_{21} + X_{i+1,j+1}F_{22},$$

which gradient can be easily computed albeit cumbersome. Thus, the backprobagation works exactly the same for CNN.

The process of finding gradients through either symbolic or numeric means are usually either too slow or not stable enough given an extremely deep network. Modern computer scientists have been employing a technique called Automatic Differentiation [**ad**] to overcome these difficulties.

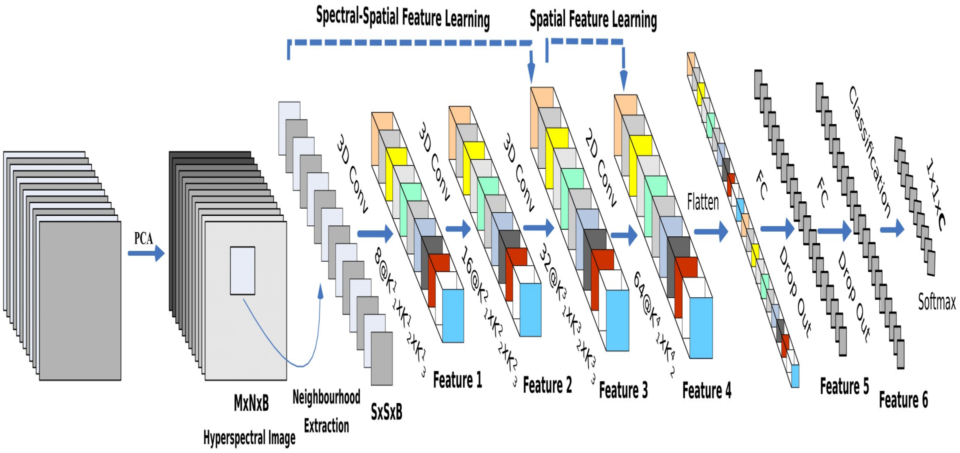


Figure 6: Hybrid Spectral Net [5].

Remark. This model was chosen due to its near-perfect accuracy score on other hyperspectral classification tasks [17] despite its simplicity structure.

4 Optimization Theory

Optimization theory is a branch of mathematics that concerns itself with characterizing the solutions of finding a set of parameters that minimizes (or maximizes) a certain cost function, usually with respect to some constraints. This section aims to provide a brief introduction of the field of optimization and the methods used to solve the Support Vector Machine (SVM) problem.

4.1 Problem Formulation

In the context of real-world applications not restricted to SVM, one may encounter problems that are of the form of maximizing or minimizing a function w.r.t some constraints. These problems can be formulated as:

Definition 4.1.1. (Primal optimization problem) Given functions f, g_i, $i = 1, \dots k$, and h_i, $i = 1, \dots, m$, defined on a domain $\Omega \subseteq \mathbb{R}^n$,

$$\begin{aligned} \min \quad & f(w), \quad w \in \Omega \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \dots, k,, \\ & h_i(w) = 0, \quad i = 1, \dots, m, \end{aligned} \tag{4.1}$$

where $f(w)$ is called the *objective* function, and the remaining relations are called, respectively, the inequality and equality constraints. The optimal value of the objective function is called the value of the optimization problem.

Notice that we can easily convert a maximization problem to a minimization one by reverting the sign of $f(w)$. Using the optimization problem in Definition 4.1.1, define

$$R = \{w \in \Omega \mid g_i(w) \leq 0, h_i(w) = 0\}$$

as the feasible region i.e the region of the domain where the objective function is defined and all the constraints are satisfied.
A solution of the optimization problem is a point $w^* \in R$ such that for any $w \in R$, $f(w) > f(w^*)$, also known as the global minimum (or maximum).

Definition 4.1.2. A point $w^* \in \Omega$ is called a *local minimum* of $f(w)$ if there exists some $\epsilon > 0$ such that for all w that satisfies $\|w - w^*\| < \epsilon$, $f(w^*) \leq f(w)$.

An inequality constraint $g_i(w) \leq 0$ is said to be *active* if the solution w^* satisfies $g_i(w^*) = 0$, i.e, on the boundary, and it is said to be *inactive* otherwise. To transform an inequality constraint into an equality one, *slack variables* denoted as ξ can be applied as follow:

$$g_i(w) \leq 0 \iff g_i(w) + \xi_i = 0, \text{ with } \xi_i \geq 0.$$

These variables will be used extensively when we introduce Soft Margin SVM, to indicate a certain amount of "looseness" in the constraint. Most if not all the loss functions used in this paper will be convex so we will restrict the content accordingly.

Definition 4.1.3. A set $\Omega \subseteq \mathbb{R}^n$ is convex, if for all $x, y \in \Omega$, and for all $\theta \in [0, 1]$

$$\theta x + (1 - \theta)y \in \Omega.$$

A function $f : \Omega \to \mathbb{R}$ is called *convex* if for all $w, u \in \Omega$, and for $\theta \in [0, 1]$,

$$f(\theta w + (1 - \theta)u) \leq \theta f(w) + (1 - \theta)f(u).$$

If a strict inequality holds then the function is said to be *strictly convex*.

The reason why we mainly use convex functions in this paper will be clear from the following proposition.

Proposition 4.1.1. *If a function $f : \Omega \to \mathbb{R}$ is convex, then any local minimum of the unconstrained optimization with the objective function f is also a global minimum.*

Proof. Let $w^* \in \Omega$ be a local minimum then there exists $\epsilon > 0$ s.t

$$f(w^*) \leq f(w), \ \forall w \in B(w^*, \epsilon) = \{w : \|w - w^*\| < \epsilon\}.$$

Suppose that there is a $u \in \Omega$ with

$$f(u) < f(w^*)$$

then because Ω is convex, we have

$$\theta w^* + (1 - \theta)u \in \Omega, \quad \forall \theta \in [0, 1].$$

By the convexity of f,

$$\begin{aligned} f(\theta w^* + (1 - \theta)u) &\leq \theta f(w^*) + (1 - \theta)f(u) \\ &< \theta f(w^*) + (1 - \theta)f(w^*) \\ &= f(w^*). \end{aligned}$$

Choose a θ sufficiently close to 1 such that $1 \leftarrow \theta w^* + (1 - \theta)u \in B(w^*, \epsilon)$ but then $f(w^*) < f(w^*)$ which is a contradiction. □

Thus, by imposing the convexity condition on our optimization problem, we can guarantee that a unique solution exists. The next part will present the Lagrange multipliers technique to solve convex quadratic optimization problem.

4.2 Lagrangian Theory

The methods of Lagrange multipliers and the Lagrangian function were first developed by Lagrange in 1797 [18] and extended by Kuhn, Tucker [19] in 1951 to allow for inequality constraints. These theories will provide the sufficient solutions for the Support Vector Machine problem.

Theorem 4.2.1. *(Fermat) A necessary condition for w^* to be a minimum of $f(w)$, $f \in C^1$, is*

$$\frac{\partial f(w^*)}{\partial w} = 0.$$

This condition, together with convexity of f, is also a sufficient condition.

Definition 4.2.1. Given an optimization problem with objective function $f(w)$, and equality constraints $h_i(w) = 0$, $i = 1, \dots, m$, we define the *Lagrangian function* as

$$L(w, \alpha) = f(w) + \sum_{i=1}^{m} \alpha_i h_i(w)$$

where the coefficients α_i are called the *Lagrange multipliers.*

The Lagrangian function incorporates information about both the objective function and the constraints, whose stationary points can be used to find solutions.

Theorem 4.2.2. *(Lagrange) A necessary condition for a normal point w^* to be a minimum of $f(w)$ subject to $h_i(w) = 0$, $i = 1, \dots, m$ with $f, h_i \in C^1$, is*

$$\frac{\partial L(w^*, \alpha^*)}{\partial w} = 0,$$
$$\frac{\partial L(w^*, \alpha^*)}{\partial \alpha} = 0,$$

for some values α^*. The above conditions are also sufficient provided that $L(w, \beta^*)$ is a convex function of w.

A simple geometric interpretation of this theorem with only 1 equality constraint can be given as trying to "climb" a surface defined by $f(w)$ subjected to a path $h(w) = 0$. Then, the solution would be the point where the gradient of the surface is parallel to the gradient of the curve, $\nabla f = \lambda \nabla h$, $lambda \in \mathbb{R}$.

More often than not, one will encounter optimization problems with inequality constraints mixed in such as during a SVM problem. In light of this, we will introduce the generalized Lagrangian.

Definition 4.2.2. Given an optimization with domain $\Omega \subseteq \mathbb{R}^n$,

$$\begin{aligned} \min_{w \in \Omega} \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \dots, k, \\ & h_i(w) = 0, \quad i = 1, \dots, m, \end{aligned}$$

we define the *generalised Lagrangian function* as

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{m} \beta_i h_i(w).$$

Definition 4.2.3. The *Lagrangian dual problem* of the primal problem of Definition 4.1.1 is the following problem:

$$\begin{aligned} \max \quad & \theta(\alpha, \beta) \\ \text{s.t.} \quad & \alpha \geq 0 \end{aligned} \tag{4.2}$$

where $\theta(\alpha, \beta) = \inf_{w \in \Omega} L(w, \alpha, \beta)$.

The relationship between the dual and primal problems will be clear after the following theorems.

Theorem 4.2.3. *(Weak duality theorem) Let $w \in \Omega$ be a feasible solution of the primal problem of 4.1.1 and (α, β) a feasible solution of the dual problem of 4.2. Then $f(w) \geq \theta(\alpha, \beta)$.*

Proof. By definition,

$$\begin{aligned} \theta(\alpha, \beta) &= \inf_{u \in \Omega} L(u, \alpha, \beta) \\ &\leq L(w, \alpha, \beta) \\ &= f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{m} \beta_i h_i(w) \leq f(w), \end{aligned}$$

since $g_i(w) \leq 0$ and $h_i(w) = 0$, $\alpha \geq 0$ by the feasibility of w and (α, β). □

Hence, it is not difficult to see that the value of the dual formation is upper bounded by the value of the primal,

$$\sup\{\theta(\alpha, \beta) : \alpha \geq 0\} \leq \inf\{f(w) : g(w) \leq 0, h(w) = 0\}.$$

Corollary 4.2.1. If $f(w^*) = \theta(\alpha^*, \beta^*)$, where $(\alpha^*, \beta^*), w^*$ are feasible then they solve the dual and primal problems respectively. In this case, $\alpha_i^* g_i(w^*) = 0$, for $i = 1, \dots, k$.

Proof. Since $f(w^*) = \theta(\alpha^*, \beta^*)$ and $\alpha_i^* g_i(w^*) = 0$, for $i = 1, \ldots, k$,

$$\begin{aligned}
\theta(\alpha^*, \beta^*) &= f(w^*) \\
&= f(w^*) + \sum_{i=1}^{k} \alpha_i^* g_i(w) + \sum_{i=1}^{m} \beta_i^* h_i(w) \\
&= L(w^*, \alpha^*, \beta^*) \\
&= \inf_{u \in \Omega} L(u, \alpha^*, \beta^*).
\end{aligned}$$

□

By comparing the primal and dual values in parallel and check the **duality gap** i.e their difference, one may be able to find the optimal solution if this reduces to zero. However, it is not generally guaranteed that the primal and dual problems will have the same values as solution.

Theorem 4.2.4. *(Strong duality theorem) Given an optimization problem with convex domain $\Omega \subset \mathbb{R}^n$,*

$$\begin{array}{cll}
\min\limits_{w \in \Omega} & f(w) & \\
\text{s.t.} & g_i(w) \leq 0, & i = 1, \ldots, k, \\
 & h_i(w) = 0, & i = 1, \ldots, m,
\end{array} \tag{4.3}$$

where g_i, h_i are affine functions and f convex then the duality gap is zero.

The following theorem given by Kuhn-Tucker states the conditions for an optimal solution to a general optimization problem.

Theorem 4.2.5. *(Kuhn-Tucker) Given an optimization problem with convex domain $\Omega \subseteq \mathbb{R}^n$,*

$$\begin{array}{cll}
\min\limits_{w \in \Omega} & f(w) & \\
\text{s.t.} & g_i(w) \leq 0, & i = 1, \ldots, k, \\
 & h_i(w) = 0, & i = 1, \ldots, m,
\end{array} \tag{4.4}$$

with $f \in C^1$ convex and g_i, h_i affine. The necessary and sufficient conditions for a normal point w^* to be an optimum are the existence of α^*, β^* such that

$$\begin{aligned}
\frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial w} &= 0, \\
\frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial \beta} &= 0, \\
g_i(w^*) &\leq 0, \ i = 1, \ldots, k, \\
\alpha_i^* g_i(w^*) &= 0, \ i = 1, \ldots, k, \\
\alpha_i^* &\geq 0, \ i = 1. \ldots, k.
\end{aligned}$$

The second and third conditions are necessary to satisfy the primal feasibility while the last condition satisfies the dual's. The fourth condition is usually known as Karush-Kuhn-Tucker complementary condition, it implies that for active constraints, $\alpha_i^* \geq 0$, whereas for inactive constraints $\alpha_i^* = 0$. This also follow from 4.2.1 since we are assuming zero duality gap.

The dual representation of a primal problem often turns out to be easier to solve since handling inequality constraints directly is difficult. The primal can be transformed into a dual by setting the derivatives of the Lagrangian w.r.t the primal variables and substituting the obtained relations back into the Lagrangian therefore removing the dependence on said variables. This corresponds to computing the function

$$\theta(\alpha, \beta) = \inf_{w \in \Omega} L(w, \alpha, \beta).$$

Which leaves us with the step of maximizing the resulting function under a (much) simpler constraint.

5 Support Vector Machine

Given a set of (linearly separable) data points and labels like in Figure 7 . There are infinitely many hyperplanes that can be used to separate the two classes but optimally, we would want a hyperplane that can generalize well for unseen data. If a hyperplane that is too close to the data points is chosen then the permitted margin of error would be to small to predict unknown data accurately because it is easier for them to fall on either sides of the hyperplanes. Thus, an ideal separator would be the one that has the largest margin i.e the distance between the nearest data points to the plane is greatest [20].

5.1 Hard Margin SVM

We start off with the assumption that there exists a hyperplane that can perfectly separate or classifies the dataset.

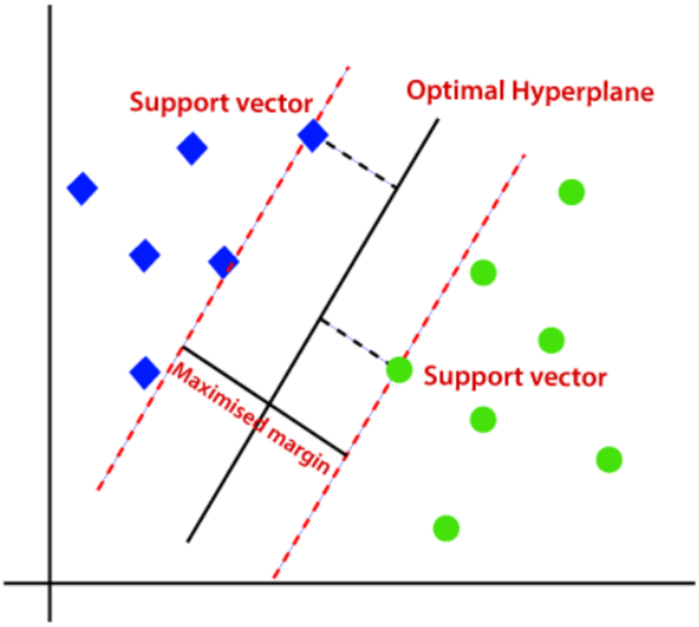


Figure 7: Support vectors [21]

To put it concretely, let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ be the set of data points with $x_i \in \mathbb{R}^m$ and $y_i \in \{1, -1\}$ as the labels. Let $h = w \cdot x + b$ be an arbitrary hyperplane then define the positive and negative support vectors, x^+ and x^-, of this hyperplane as the points closest to h. Note that h, ch, for $c \in \mathbb{R}$, define the same hyperplane so we can choose our normalization factor such that

$$\begin{aligned}\langle w, x^+\rangle + b &= 1\\ \langle w, x^-\rangle + b &= -1.\end{aligned}$$

Then the set S is called linear separable if there exists a hyperplane defined by (w, b) such that

$$\begin{cases}\langle w, x_i\rangle + b \geq 1 & \text{for } y_i = 1\\ \langle w, x_i\rangle + b \leq -1 & \text{for } y_i = -1\end{cases}$$

or

$$y_i(\langle w, x_i\rangle + b) \geq 1 \quad \forall i \in \{1, 2, \ldots, n\}.$$

The (geometric) margin is then given by

$$\gamma = \frac{|w \cdot x^+ + b|}{\|w\|} + \frac{|w \cdot x^- + b|}{\|w\|} = \frac{2}{\|w\|}$$

where $\|.\|$ is the standard Euclidean norm. Recall that the objective is to find a hyperplane such that γ is maximized while being subjected to $y_i(w \cdot x_i + b) \geq 1$ i.e

$$\begin{aligned}\max_{w,b} \quad & \gamma\\ \text{s.t.} \quad & y_i(\langle w, x_i\rangle + b) \geq 1,\\ & i = 1, 2, \ldots, n\end{aligned}$$

which is equivalent to

$$\begin{aligned}\min_{w,b} \quad & \frac{\|w\|^2}{2}\\ \text{s.t.} \quad & y_i(\langle w, x_i\rangle + b) \geq 1,\\ & i = 1, 2, \ldots, n.\end{aligned} \tag{5.1}$$

i.e a quadratic optimization problem.

Following the steps outlined in Section 4, the primal Lagrangian is

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^{n} \alpha_i[y_i(\langle w, x_i\rangle + b) - 1]$$

where $\alpha_i \geq 0$ are the Lagrange multipliers.

Differentiating the Lagrangian w.r.t to w, b to obtain

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \implies w = \sum_{i=1}^{n} \alpha_i y_i x_i,$$
$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0.$$

Resubstituting the above relations into the primal Lagrangian to obtain

$$\begin{aligned} L(w,b,\alpha) &= \frac{\langle w, w\rangle}{2} - \sum_{i=1}^{n} \alpha_i [y_i(\langle w, x_i\rangle + b) - 1] \\ &= \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle + \sum_{i=1}^{n} \alpha_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle. \end{aligned}$$

We have the resulting dual problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \\ \text{subject to} \quad & \sum_{i=1}^{n} y_i \alpha_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, \ldots, n. \end{aligned}$$

Remark. The relation $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ shows that the weights w can be described as a linear combination of the training data. This is called the dual representation.

Recall the Kuhn-Tucker complimentary condition 4.2.5 from the previous chapter that for a solution α^*, (w^*, b^*) to be optimal, it must satisfy

$$\alpha_i^* [y_i(\langle w^*, x_i\rangle + b^*) - 1)] = 0, \quad i = 1, \ldots, n.$$

Define $\langle w, x_i\rangle + b$ as the functional margin of x_i w.r.t (w, b). This implies that for points x_i which the functional margin is 1 and lies closest to the hyperplane has non-zero α_i^* and vice versa. Thus, the hyperplane is only determined by the points closest to it i.e, support vectors. The hypothesis function $f(x, \alpha^*, \beta^*)$ can then be given as

$$f(x, \alpha^*, b^*) = \sum_{i \in \text{sv}} y_i \alpha_i^* \langle x_i, x\rangle + b^*$$

where sv is the set of support vectors, with the decision rule given by $sgn(f(x))$.

Having solved the dual problem to obtain the Lagrange multipliers, we can easily find the weights w but b is still unknown. In [22], b is calculated by taking the average of $y_i - \langle w, x_i \rangle$ i.e

$$b = \frac{1}{S} \sum_{i=1}^{S} (y_i - \langle w, x_i \rangle)$$

where S is the number of support vectors. Other authors such as [23] only take the average of the nearest positive and negative support vectors

$$b = -\frac{\max_{y_i=-1} \langle w, x_i \rangle + \min_{y_i=1} \langle w, x_i \rangle}{2}.$$

The formulations stated above belong to the Hard Margin SVM problem because the hyperplane (w, b) must classify each point correctly. In practice, more often than not, this is unachievable and impractical for a variety of reasons. Thus, ideally, it is desirable to have a classifier that allow for "minor" mistakes while retaining its generalization. Note that there are many versions to the Hard Margin SVM problem [24].

5.2 Soft Margin SVM

Supposed that the data is not linearly separable, then the optimization problem 5.1 can not be solved since the condition $y_i(\langle w, x_i \rangle + b)$ is not satisfiable. Also, if the data is linearly separable but the separating hyperplane does not leave a wide enough margin to account for unseen data then this would indeed be a bad classifier. Thus arise the need to alter the original Hard Margin SVM problem to account for the trade-off between errors and generalizability.

Instead of forcing our classifier to be correct at every point x_i, we introduce the *slack* variables ξ_i to our constraint as

$$\begin{aligned} \min_{w,\, b} \quad & \frac{1}{2} \langle w, w \rangle \\ \text{s.t.} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & i = 1, 2, \ldots, n. \end{aligned} \tag{5.2}$$

Clearly, we could let $\xi_i \to \infty$ and the constraint would be satisfied but this has no added-value to the problem so we would want to constraint the values of ξ_i. There are many formulation of this approach but we shall use the one introduced by [23],

$$\begin{aligned} \min_{w,\, b,\, \xi} \quad & \frac{1}{2} \langle w, w \rangle + C \sum_{i=1}^{n} \xi_i^2 \\ \text{s.t.} \quad & y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & i = 1, 2, \ldots, n \end{aligned} \tag{5.3}$$

for some $C \in \mathbb{R}$. This problem is called L_2-norm soft margin. The Lagrangian and dual problem can then be obtained in the same way as the hard margin one.

The constant C was introduced as a way to control the importance of the *slack* variables. If $C \to \infty$ then it is necessary that $\xi_i \to 0$ for $i = 1, \ldots, n$ and thus become a Hard Margin SVM problem. Each different C can lead to a different classifier that may or may not be satisfactory so to find the optimal C, we have to try different values. Some of the recommended approaches can be found in [25] but they are beyond the scope of this paper.

So far, we have only assumed that the data is either linearly separable or linear separable but with a few outliers that can still be somewhat accurately classified using a separating hyperplane. What if, the data is non-linearly separable in such a way that a linear separator would not yield satisfactory results. Hence, it is necessary that we introduce a very important method to transform our feature vectors in such a way that they become linearly separable.

6 Positive Definite Kernels and the Reproducing Kernel Hilbert Space

This section aims to introduce the **kernel** method that is used to transform a feature vector into a higher or infinite dimensional space with the aim of making the dataset linearly separable. We will also present related and necessary objects to supplement the kernel method such as Hilbert space, Reproducing Kernel Hilbert Space (RKHS) and the Representer Theorem. Although there are many publications that omit these definitions, we decline to do the same since they are crucial for the goal of this paper, that is Laplacian SVM.

6.1 Positive Definite Kernels

We begin with a few necessary definitions of basic spaces to build up to the Hilbert space.

Definition 6.1.1. Let X be a non-empty set with a distance function (or metric) $d : X \times X \to \mathbb{R}^+$ then X is a metric space if for all $x, y, z \in X$, d satisfies the following conditions:

1. $d(x, y) = 0 \iff x = y$.
2. $d(x, y) = d(y, x)$.
3. $d(x, y) + d(y, z) \geq d(x, z)$.

Example 6.1.1. The set of real numbers $\mathbb{R}$ forms a metric space with $d(x, y) = |x - y|$.

Definition 6.1.2. Let V be a vector space over $\mathbb{R}$. A norm function on V is a function $\|.\| : V \to \mathbb{R}$ such that for all $v, w \; inV$ and $a \in \mathbb{R}$,

1. $\|v\| \geq 0$ with $\|v\| = 0 \iff v = 0$.
2. $\|av\| = |a| \, \|v\|$.
3. $\|v + w\| \leq \|v\| + \|w\|$.

This norm induces a metric $d(v, w) = \|v - w\|$ on V.

Example 6.1.2. The space $\mathbb{R}^n$ is a vector space over $\mathbb{R}$ with the L_2 norm:

$$\|v\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}.$$

Definition 6.1.3. A normed vector space V is complete if every Cauchy sequence in V converges to a vector $v \in V$ with respect to the norm.

Example 6.1.3. The normed vector space $(\mathbb{Q}, \|.\|)$ is not complete because if a sequence is recursively defined as

$$x_0 = \frac{4}{3}$$

$$\forall n \in \mathbb{N} : x_{n+1} = \frac{4 + 3x_n}{3 + 2x_n}$$

then it can be shown [26] that this sequence is Cauchy and converges to $\sqrt{2}$ which is not in $\mathbb{Q}$.

Definition 6.1.4. Let V be a vector space over $\mathbb{R}$. An inner product on V is a function $\langle ., . \rangle : V \times V \to \mathbb{R}$ such that for any $v, w, u \in V$ and $a, b \in \mathbb{R}$, we have

1. $\langle u, v \rangle = \langle v, u \rangle$.
2. $\langle au + bv, w \rangle = a\langle u, w \rangle + b\langle v, w \rangle$.
3. $\langle v, v \rangle \geq 0$ with $\langle v, v \rangle = 0 \iff v = 0$.

A vector space that is equipped witn an inner product is called an inner product space.

Example 6.1.4. The vector space $\mathbb{R}^n$ is an inner product space with the inner product as the dot product.

Definition 6.1.5. Let $(V, \langle ., . \rangle)$ be an inner product space, the norm induced by $\langle ., . \rangle$ can be defined as

$$\|v\| = \sqrt{\langle v, v \rangle} \text{ for } v \in V.$$

Remark. Not every norm is induced by an inner product. In fact, it holds if and only if the parallelogram identity

$$\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2$$

holds.

Definition 6.1.6. (Hilbert space) A Hilbert space is an inner product space that is complete with respect to the norm induced by the inner product.

Hilbert space is important to our research because of its rich structure and the methods that we are going to introduce, all operate within Hilbert space. We now can define what is a kernel. All Hilbert spaces mentioned from this moment are assumed to be real until stated otherwise.

Definition 6.1.7. (Kernel) Let X be a non-empty set, let H be a Hilbert space, and let $\varphi : X \to H$ be a function called feature map. The function $K : X \times X \to \mathbb{R}$ given by

$$K(x, y) = \langle \varphi(x), \varphi(y) \rangle$$

is called a kernel function.

Example 6.1.5. Let $X \subset \mathbb{R}^2$, and let $\varphi : X \to \mathbb{R}^3$ be the map given by

$$\varphi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2).$$

We have

$$\begin{aligned} \langle \varphi(x_1, x_2), \varphi(y_1, y_2) \rangle &= \langle (x_1^2, x_1^2, \sqrt{2}x_1x_2), (y_1^2, y_1^2, \sqrt{2}y_1y_2) \rangle \\ &= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\ &= (x_1y_1 + x_2y_2)^2 = \langle x, y \rangle, \end{aligned}$$

where $\langle x, y \rangle$ is the usual inner product on $\mathbb{R}^2$. Hence,

$$K(x, y) = \langle x, y \rangle^2$$

is a kernel function associated with the feature space $\mathbb{R}^3$.

One of the reason why the kernel function is often used in various contexts of Machine Learning is that, as you have seen, the calculation and derivation of the feature map φ is time-consuming and resource intensive while it could be replaced by a simple inner product. Furthermore, if the Hilbert space H is infinite dimensional then clearly, computing φ is impossible but nevertheless, the expression $K(x, y)$ can be found, avoiding φ altogether.

Using the kernel function, we can then transform the hard margin dual problem into

$$\begin{array}{ll} \max\limits_{\alpha} & \displaystyle\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{s.t.} & \qquad \alpha_i \geq 1 \quad i = 1, \dots, n, \\ & \displaystyle\sum_{i=1}^{n} y_i \alpha_i = 0 \quad i = 1, \dots, n \end{array} \tag{6.1}$$

which can then be solved similarly to the old one with the exception that the dataset is now linearly separable, assuming that it is indeed the case in the feature space.

Remark. The same kernel can arise from different maps into different feature spaces. Example: $\varphi_1(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ in $\mathbb{R}^3$ and $\varphi_2(x_1, x_2) = (x_1^2, x_2^2, x_1x_2, x_1x_2)$ in $\mathbb{R}^4$.

One of the important property of the kernel functions is the positive definite-ness and the reason for its importance will become clear by the end of this section.

Definition 6.1.8. (Positive definite kernel) Let X be a non-empty set. A function $\kappa : X \times X \to \mathbb{C}$ is a *positive definite kernel* if for every finite $S \subset X$, if K_s is the $p \times p$ matrix

$$K_s = (K(x_j, x_i))_{1 \leq i,j \leq p}$$

then we have

$$u^* K_s u = \sum_{i,j=1}^{p} K(x_i, x_j) u_i \overline{u_j} \geq 0, \quad \text{for all } u \in \mathbb{C}^p$$

where u^* denotes the complex conjugate of u.

Remark.

Remark. A positive definite kernel that is also symmetric i.e $K(x, y) = K(y, x)$ is called a *Mercer Kernel.*

Proposition 6.1.1. *Let X be any non-empty set, let H be any Hilbert space, let $\varphi : X \to H$ be any function, and let $K : X \times X \to \mathbb{C}$ be the kernel given by*

$$K(x, y) = \langle \varphi(x), \varphi(y) \rangle, \quad x, y \in X.$$

For any finite subset $S = \{x_1, \ldots, x_p\}$ of X, if K_s is the $p \times p$ matrix

$$K_s = (K(x_j, x_i))_{1 \leq i,j \leq p}$$

then we have

$$u^* K_s u \geq 0, \quad \textit{for all } u \in \mathbb{C}^p.$$

Proof.

$$\begin{aligned} u^* K_s u &= \sum_{i,j=1}^{p} K(x_i, x_j) u_i \overline{u_j} \\ &= \sum_{i,j=1}^{p} \langle \varphi(x), \varphi(y) \rangle u_i \overline{u_j} \\ &= \left\langle \sum_{i=1}^{p} u_i \varphi(x_i), \sum_{j=1}^{p} u_j \varphi(x_j) \right\rangle \\ &= \left\| \sum_{i=1}^{p} u_i \varphi(x_i) \right\|^2 \geq 0. \end{aligned}$$

□

Proposition 6.1.2. *(I. Schur) If $K_1, K_2 : X \times X \to \mathbb{R}$ are two positive definite kernels, then the function $K : X \times X \to \mathbb{R}$ given by $K(x, y) = K_1(x, y)K_2(x, y)$ for all $x, y \in X$ is also a positive definite kernel.*

Here are some ways of obtaining new positive definite kernels from old ones

Proposition 6.1.3. *Let $K_1 : X \times X \to \mathbb{R}$ and $K_2 : X \times X \to \mathbb{R}$ be two positive definite kernels, let $f : X \to \mathbb{R}$, $\psi : X \to \mathbb{R}^N$ be functions, $K_3 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a positive definite kernel, and $a \in \mathbb{R}^+$, $p(z)$ be a polynomial with nonnegative coefficients. Then the following functions are positive definite kernels:*

(1) $K(x, y) \ = K_1(x, y) + K_2(x, y)$.

(2) $K(x, y) = aK_1(x, y)$.

(3) $K(x, y) = K_3(\psi(x), \psi(y))$.

(4) $K(x, y) = f(x)\overline{f(y)}$.

(5) $K(x, y) = p(K_1(x, y))$.

(6) $K(x, y) = e^{K_1(x,y)}$.

(7) If X is a real Hilbert space with inner product $\langle -, -\rangle_X$ and corresponding norm $\| \ \|_X$,

$$K(x, y) = e^{-\frac{\|x-y\|_X^2}{2\sigma^2}}$$

for any $\sigma > 0$.

Proof. (1), (2) and (3) are trivial.

(4) Let $S = \{x_1, \ldots, x_p\} \subset X$, if K is the $p \times p$ matrix

$$K = (\overline{f(x_k)}f(x_j))_{1\leq j,k\leq p}$$

then we have

$$u^*Ku = \sum_{k,j=1}^{p} u_j f(x_j)\overline{u_k f(x_k)} = \left|\sum_{j=1}^{p} u_j f(x_j)\right|^2 \geq 0.$$

(5) Let $p(z) = \sum_{i=0}^{m} a_i z^i$, then

$$p(K_1(x, y)) = a_m K_1(x, y)^m + \cdots + a_1 K_1(x, y) + a_0.$$

Since $a_i \in \mathbb{R}^+$ for $i = 0, \ldots, m$, by Proposition 6.1.2 and (2), each $a_i K_i(x, y)^i$ is a positive definite kernel. By (4), (1) with $f(x) = \sqrt{a_0}$, $p(K_1(x, y))$ is a positive definite kernel.

(6) We first show that if each $K_i : X \times X \to \mathbb{C}$ is a positive definite kernel,

$$\lim_{i\to\infty} K_i(x,y) = K(x,y)$$

is also a positive definite kernel if it exists. We have

$$\begin{aligned}\sum_{j,k=1}^{n} u_j u_k K(x_j,x_k) &= \sum_{j,k=1}^{n} u_j u_k \left(\lim_{i\to\infty} K_i(x_j,x_k)\right) \\ &= \sum_{j=k=1}^{n} \lim_{i\to\infty} (u_j u_k K_i(x_j,x_k)) \\ &= \lim_{i\to\infty} \underbrace{\left(\sum_{j,k=1}^{n} u_j u_k K_i(x_j,x_k)\right)}_{\geq 0} \geq 0.\end{aligned}$$

Thus, K is positive definite. Note that

$$e^{K_1(x,y)} = \lim_{n\to\infty} \sum_{k=0}^{n} \frac{K_1(x,y)^k}{k!}$$

but each partial sum

$$\sum_{k=0}^{n} \frac{K_1(x,y)^k}{k!}$$

is a positive definite kernel so $e^{K_1(x,y)}$ is also a positive definite kernel.

(7) By (2) and since the map $(x,y) \to \langle x,y\rangle_X$ is a positive definite kernel by Proposition 6.1.1 with identity feature map, the function

$$(x,y) \to \frac{\langle x,y\rangle_X}{\sigma^2}$$

is a positive definite kernel so it follows that

$$K_1 = e^{\frac{\langle x,y\rangle_X}{\sigma^2}}$$

is also a positive definite kernel. Let $f : X \to \mathbb{R}$ be defined as

$$f(x) = e^{-\frac{\|x\|_X^2}{2\sigma^2}}$$

then by (4),

$$K_2(x,y) = f(x)\overline{f(y)} = f(x)f(y) = e^{-\frac{\|x\|_X^2}{2\sigma^2}} e^{-\frac{\|y\|_X^2}{2\sigma^2}} = e^{-\frac{\|x\|_X^2+\|y\|_X^2}{2\sigma^2}}$$

is a positive definite kernel. Thus,

$$
\begin{aligned}
K_1(x,y)K_2(x,y) &= e^{\frac{2\langle x,y\rangle_X}{2\sigma^2}} e^{-\frac{\|x\|_X^2+\|y\|_X^2}{2\sigma^2}} \\
&= e^{-\frac{\|x-y\|_X^2}{2\sigma^2}}
\end{aligned}
$$

is a positive definite kernel. It is usually called *Gaussian kernel.*

□

6.2 Reproducing Kernel Hilbert Space

We are ready to introduce the Reproducing Kernel Hilbert Space (RKHS) which is a Hilbert space of functions with the reproducing property. The reason why this space is important has to do with the **Moore-Aronszajn theorem** which connect positive definite kernels to feature space and allow the kernel trick to work.

Definition 6.2.1. (RKHS) A Reproducing Kernel Hilbert Space is a Hilbert space $\mathcal{H}$ of functions $f : X \to \mathbb{R}$ with a reproducing kernel $K : X \times X \to \mathbb{R}$ where $K(x,.) \in \mathcal{H}$ and $f(x) = \langle K(x,.), f\rangle_{\mathcal{H}}$ [27].

The property $f(x) = \langle K(x,.), f\rangle$ is called the *reproducing* property.

Theorem 6.2.1 (Moore-Aronszajn)**.** *If K is a Mercer kernel on a set X. Then there is a unique RKHS space of functions on X for which K is the reproducing kernel.*

Proof. We will only provide a sketch of the proof. The full details can be found at [28].

Let $x \in X$ and define $K_x = K(x,.)$. Let H_0 be the linear span of $\{K_x : x \in X\}$. Let $f, g \in H_0$ given by

$$
f = \sum_{i=1}^{m} a_i K_{x_i}, \quad g = \sum_{j=1}^{n} b_j K_{y_j},
$$

then define an inner product on H_0 as

$$
\langle f, g\rangle_{H_0} = \sum_{i=1}^{m}\sum_{j=1}^{n} a_i b_j K(x_i, y_j).
$$

This shows that the inner product is symmetric bilinear. Using Cauchy-Schwartz inequality, it can be shown that H_0 is a pre-Hilbert space. Let H be the set of functions which are pointwise limits of Cauchy sequences in H_0. We can then show that H is complete w.r.t the inner product on H_0 then H is a RKHS which admit K as its reproducing kernel and every $f \in H$ can be written as a linear combination of K. □

This theorem combines with the definition of RKHS states that for each RKHS we have a reproducing Mercer kernel and for each Mercer kernel, we can get a unique RKHS.

To put it more concretely, suppose that instead of wanting to compute $\langle x_i, x_j \rangle$ in the dual formulation of the Hard Margin SVM problem

$$L(w, b, a) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle,$$

we want to compute the inner product of the feature map $\varphi : X \to H$, for H a Hilbert space, $\langle \varphi(x_i), \varphi(x_j) \rangle$ in a higher dimensional space. This is potentially problematic because it requires us to explicitly compute the feature map which can be computationally expensive or impossible if the dimension of H is infinite. However, if instead one chooses a Mercer kernel K then by 6.2.1, there exists a unique RKHS $\mathcal{H}$ where K has the reproducing property. Then this becomes much easier by for example, trivially letting $\varphi(x) = K_x \in \mathcal{H}$ then

$$K(x_i, x_j) = \langle K_{x_i}(.), K_{x_j}(.) \rangle = \langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}},$$

thus skipping the step of computing the feature map. However, Theorem 6.2.1 only guarantees the existence of feature maps but not how to find them, and for that we shall require one important theorem that provides a representation of the feature map.

Note that for a given Mercer kernel K, the corresponding feature map is not unique.

Example 6.2.1. Let K be a Mercer kernel given by $K(x, y) = \langle x, y \rangle^2$ with $X = \mathbb{R}^2$ then

$$K(x, y) = x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2.$$

It can be easily seen that the two feature maps

$$\phi(x) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$$
$$\varphi(x) = (x_1^2, x_2^2, x_1 x_2, x_1 x_2)$$

correspond to the same kernel K with different feature spaces.

Definition 6.2.2. (L_p space) Consider a function f with domain $[a, b] \subset \mathbb{R}$. For $p > 0$, let the L_p norm be defined as :

$$\|f\|_p = \left(\int |f(x)|^p dx \right)^{\frac{1}{p}}.$$

The L_p space is defined as the set of functions with bounded L_p norm:

$$L_p(a, b) := \{ f : [a, b] \to \mathbb{R} \mid \|f\|_p < \infty \}.$$

Theorem 6.2.2 (Mercer's Theorem)**.** *Suppose $K : [a,b] \times [a,b] \to \mathbb{R}$ is a continuous symmetric positive semi-definite kernel,*

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_i c_j K(x_i, x_j) \geq 0,$$

which is bounded:

$$\sup_{x,y} K(x,y) < \infty.$$

Define the operator T_k as

$$T_k f(x) = \int_a^b K(x,y) f(y) dy.$$

The operator T_k is called the Hilbert-Schmidt integral operator. This output function is positive definite:

$$\iint K(x,y) f(y) dx dy \geq 0.$$

Then there is a set of orthonormal basis $\{\phi_i(.)\}_{i=0}^{\infty}$ of $L_2(a,b)$ consisting of eigenfunctions of T_k such that the corresponding sequence of eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ are non-negative:

$$\int K(x,y)\phi_i(y) dy = \lambda_i \phi_i(x).$$

The eigenfunctions corresponding to the non-zero eigenvalues are continuous on $[a,b]$ and k can be represented as

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y),$$

where the convergence is absolute and uniform.

One may notice the similarity between Mercer and Moore-Aronszajn theorems that they both show there exist feature maps for a Mercer kernel K. The difference is Mercer's theorem creates a feature map by computing the eigenfunctions of T_k while Moore-Aronszajn's let the feature map $\phi = K_x$.

7 Riemannian Manifold

A manifold is a mathematical structure that locally resembles Euclidean space. This structure is useful in the field of Machine Learning is partly because of the Manifold Hypothesis: many high-dimensional data sets that occur in the real world actually lie along low-dimensional manifolds inside that high-dimensional space. Manifold regularization is a technique using the shape of the data set to constraint on the function that should be learned on it. This technique applies a constraint on the Laplace-Beltrami operator, defined on a Riemannian Manifold, of functions in the RKHS of functions from an arbitrary set to the real numbers. We start by introducing the basic ideas of topological manifolds, smooth manifolds to its tangent vectors, and then Riemannian manifolds and the Laplace-Beltrami operator.

7.1 Topology prerequisites

Definition 7.1.1. A topology on a set X is a collection $\mathcal{T}$ of subsets of X having the following properties:

(1) $\emptyset$ and X are in $\mathcal{T}$.

(2) The union of the elements of any subcollection of $\mathcal{T}$ is in $\mathcal{T}$.

(3) The intersection of the elements of any finite subcollection of $\mathcal{T}$ is in $\mathcal{T}$.

A set X for which a topology $\mathcal{T}$ has been specified is called a topology space.

If X is a topological space with topology $\mathcal{T}$, we say that a subset U of X is an *open set* of X if U belongs to the collection $\mathcal{T}$.

Definition 7.1.2. IF X is a set, a basis for a topology on X is a collection $\mathcal{B}$ of subsets of X such that

(1) For each $x \in X$, there is at least one basis element B containing x.

(2) If x belongs to the intersection of two basis elements B_1 and B_2, then there is a basis elements B_3 containing x such that $B_3 \subset B_1 \cap B_2$.

If $\mathcal{B}$ satisfies these two conditions, then we define the topology $\mathcal{T}$ generated by $\mathcal{B}$ as follows: a subset U is said to be open in X if for each $x \in U$, there is a basis element $B \in \mathcal{B}$ such that $x \in B$ and $B \subset U$. Note that each basis element is itself an element of $\mathcal{T}$.

Definition 7.1.3. Let X be a topological space with topology $\mathcal{T}$. If $Y \subset X$, the collection

$$\mathcal{T}_Y = \{Y \cap U \mid U \in \mathcal{T}\}.$$

is a topology on Y, called the subspace topology. With this topology, Y is called a subspace of X; its open sets consist of all intersections of open sets of X with Y.

If Y is a subspace of X, we say that a set U is open in Y if it belongs to the topology of Y. We say that U is open in X if it belongs to the topology of X.

Lemma 7.1.1. *Let Y be a subspace of X. If U is open in Y and Y is open in X, then U is open in X.*

Proof. Since U is open in Y, $U = Y \cap V$ for some set V open in X. Since Y and V are both open, so is $Y \cap V$. □

Definition 7.1.4. (Hausdorff space)
A topological space X is called a Hausdorff space if for each pair x_1, x_2 of distinct points of X, there exist U_1, U_2 such that $x_1 \in U_1$, $x_2 \in U_2$ and $U_1 \cap U_2 = \emptyset$.

Definition 7.1.5. (Continuous function)
Let X, Y be topological spaces. A function $f : X \to Y$ is said to be continuous if for each open subset V of Y, the set $f^{-1}(V)$ is an open subset of X.

Continuity of a function depends not only upon the function f itself, but also on the topologies specified for its domain and range.

Theorem 7.1.1. *Let X, Y, and Z be topological spaces.*

(a) If A is a subspace of X, the inclusion function $j : A \to X$ is continuous.

(b) if $f : X \to Y$ and $g : Y \to Z$ are continuous, then the map $g \circ f : X \to Z$ is continuous.

(c) If $f : X \to Y$ is continuous, and if A is a subspace of X, then the restricted function $f|_A : A \to Y$ is continuous.

(d) Let $f : X \to Y$ be continuous. If Z is a subspace of Y containing $f(X)$, then the function $g : X \to Z$ obtained by restricting the codomain of f is continuous.

Proof. (a) If U is open in X, then $j^{-1}(U) = U \cap A$, which is open in A by the definition of the subspace topology.

(b) If U is open in Z, then $g^{-1}(U)$ is open in Y and $f^{-1}(g^{-1}(U))$ is open in X but

$$f^{-1}(g^{-1}(U)) = (g \circ f)^{-1}(U).$$

(c) We have

$$f|_A = f \circ j$$

which is continuous.

(d) Let B be open in Z. Then $B = Z \cap U$ for some open set U of Y. Because Z contains the entire image set $f(X)$,

$$f^{-1}(U) = g^{-1}(B).$$

Somce $f^{-1}(U)$ is open, so is $g^{-1}(B)$. □

Definition 7.1.6. (Homeomorphism)

Let X and Y be topological spaces; let $f : X \to Y$ be a bijection. If both the function f and the inverse function

$$f^{-1} : Y \to X$$

are continuous, then f is called a homeomorphism.

Corollary 7.1.1. Let $f : X \to Y$ be a homeomorphism and A an open subset of X. The restriction of $f|_A$ is a homeomorphism.

Definition 7.1.7. Let X be a topological space, then a cover C of X is a collection of subsets $\{U_\alpha\}$ of X whose union is X. If each of U_α is open then C is an open cover. X is called compact if each of its open cover has a finite subcover.

7.2 Smooth manifolds

For discussions involving manifolds, we will follow the structure given by Lee [29].

Definition 7.2.1. (Topological manifold) Let M be a topological space. We say that M is a topological manifold of dimension n or a topological n-manifold if it has the following properties:

- M is a Hausdorff space.
- M is second-countable: there exists a countable basis for the topology of M.
- M is locally Euclidean of dimension n: each point of M has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^n$.

The third property means, more specifically, that for each $p \in M$ we can find

- an open subset $U \subseteq M$ containing p,
- an open subset $\hat{U} \subseteq \mathbb{R}^n$ and
- a homeomorphism $\varphi : U \to \hat{U}$.

Every topological manifold has, by definition, a specific well-defined dimension.

Theorem 7.2.1. *A nonempty n-dimensional topological manifold cannot be homeomorphic to an m-dimensional manifold unless $m = n$.*

The basic example of a topological n-manifold is $\mathbb{R}^n$ itself with the identity as homeomorphism. The reasons why manifolds have to satisfy the three properties is to ensure that they behave in a way that is similar to Euclidean space.

Proposition 7.2.1. *Every open subset of a topological n-manifold is itself a topological n-manifold.*

Proof. Let M be a topological n-dimensional manifold and U an open subset of M. It suffices to show that U is locally Euclidean since the other two requirements are trivial. Let x be a point in U then there exists a neighborhood V containing x that is homeomorphic to an open subset $\hat{V} \subseteq \mathbb{R}^n$. W.l.o.g let $\varphi : V \to \hat{V}$ be a homeomorphism such that $\varphi(x) = 0$. It is clear that $U \cap V = N$ is an open subset of V so $\varphi(N) \subseteq \mathbb{R}^n \ni 0$. Let $0 \in B \subset \varphi(N)$ then $\varphi^{-1}(B)$ is a neighborhood of x contained in V, which is homeomorphic to $B \subseteq \mathbb{R}^n$. □

Definition 7.2.2. (Coordinate chart) Let M be a topological n-manifold. A coordinate chart on M is a pair (U, φ), where U is an open subset of M and $\varphi : U \to \hat{U}$ is a homeomorphism.

Thus far, we have yet to define any calculus operations on manifolds such as derivatives, etc which are the heart of machine learning. To do that, we will introduce a structure called *smooth manifold.* The definition will be based on maps between Euclidean spaces and from now on, we define a map between open subsets of Euclidean spaces to be smooth if all of its component functions has continuous partial derivatives of all orders.

Definition 7.2.3. (Diffeomorphism) Given two topological manifolds M and N. A smooth map $f : M \to N$ is a *diffeomorphism* if it is bijective and its inverse is also smooth.

Definition 7.2.4. (Transition map) Let M be a topological n-manifold. If (U, φ), (V, ψ) are two charts such that $U \cap V \neq \emptyset$, the composite map $\psi \circ \varphi^{-1} : \varphi(U \cap V) \to \psi(U \cap V)$ is called the *transition map* from φ to ψ. Two charts (U, φ) and (V, ψ) are said to be *smoothly compatible* if either $U \cap V = \emptyset$ or the transition map is a diffeomorphism.

Since the transition map has domain and codomain as open subsets of Euclidean spaces, its smoothness can be defined as having continuous partial derivatives of all orders.

Definition 7.2.5. (Smooth Atlas) An atlas for a topological manifold M is a collection of charts whose domain cover M. An atlas $\mathcal{A}$ is called a *smooth atlas* if for any two charts in $\mathcal{A}$, they are smoothly compatible.

Note that for a topological manifold M, there may exist different atlases that determine the same collection of smooth functions on M. To that end, we shall define *maximal atlas.*

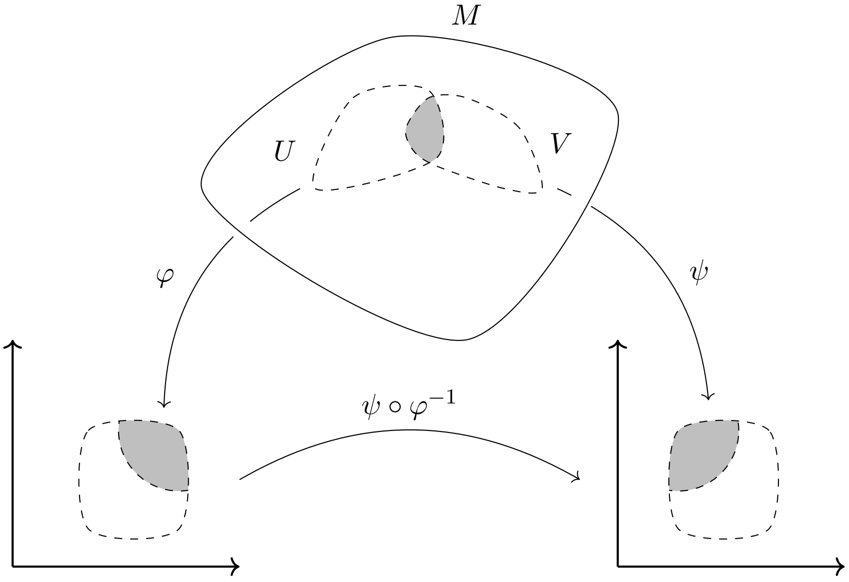


Figure 8: Transition map

Definition 7.2.6. (Maximal Smooth Atlas) A smooth atlas A on M is maximal if it is not properly contained in any larger smooth atlas.

Definition 7.2.7. (Smooth manifold) If M is a topological manifold, a *smooth structure* on M is a maximal smooth atlas. Then the pair $(M, \mathcal{A})$ is called a *smooth manifold.*

Example 7.2.1. For each non-negative integer n, the space $\mathbb{R}^n$ is a smooth n-manifold with its smooth structure determined by a single chart $(\mathbb{R}^n, \mathrm{Id}_{\mathbb{R}^n})$. This is called the standard smooth structure on $\mathbb{R}^n$.

If M is a smooth manifold, any chart (U, φ) contained in the given maximal smooth atlas is called a *smooth chart*, and the corresponding coordinate map φ is called a *smooth coordinate map.* We can represent a point $p \in U$ by its local coordinates $\varphi(p) = (x^1, \ldots, x^n)$ using Einstein notation. It is customary to say that $p = (x^1, \ldots, x^n)$ in local coordinates.

7.3 Smooth maps

From now on, any manifold is considered smooth unless stated otherwise. We usually use "functions" as maps from an object to an Euclidean space and "map" or "mapping" as maps between any arbitrary objects, in this case, manifolds.

Definition 7.3.1. (Smooth functions) Let M be a manifold, k a non-negative integer, and $f : M \to \mathbb{R}^k$ any function. We say that f is a *smooth function* if for every $p \in M$, there exists a smooth chart (U, φ) for M whose domain contains p and such that the function $f \circ \varphi^{-1}$ is smooth on $\varphi(U) \subseteq \mathbb{R}^n$.

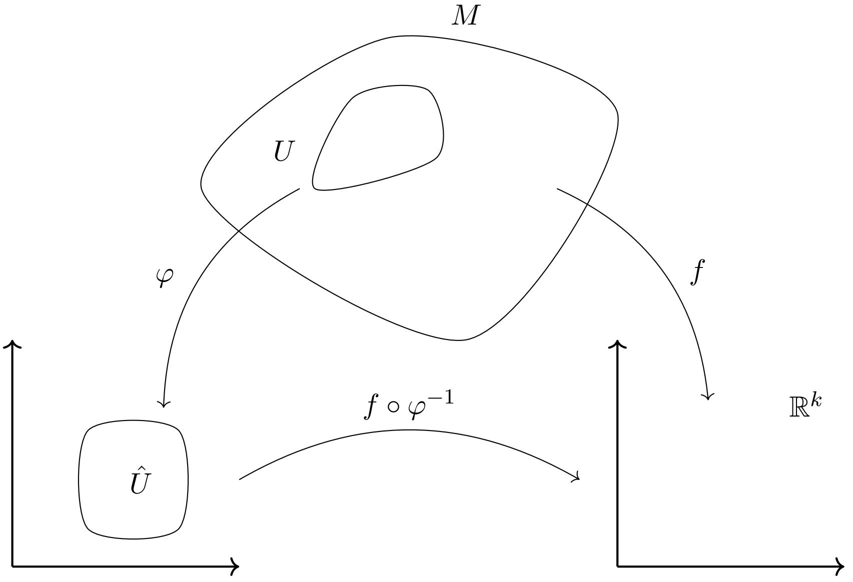


Figure 9: Smooth functions

The map $\hat{f} : \varphi(U) \to \mathbb{R}^k$ defined by $\hat{f} = f \circ \varphi^{-1}$ is called the *coordinate representation* of f. By definition, f is smooth if and only if its coordinate representation is smooth in some smooth chart around each point. We can easily generalize the definition of smooth functions to maps between between manifolds.

Definition 7.3.2. (Smooth maps) Let M, N be manifolds, and let $F : M \to N$ be any map. We say that F is a *smooth map* if for every $p \in M$, there exists a smooth chart (U, φ) containing p and (V, ψ) containing $F(p)$ such that $F(U) \subseteq V$ and the composite map $\psi \circ F \circ \varphi^{-1}$ is smooth from $\varphi(U)$ to $\psi(V)$.

7.4 Tangent Vectors

In this section, we will introduce *tangent space* to a manifold at a point to help us better define calculus on manifolds and *differential* which generalizes the total derivative of a map between Euclidean spaces. We start off with concrete objects like geometric tangent vectors in $\mathbb{R}^n$ given by [29].

Given a point $a \in \mathbb{R}^n$, define the geometric tangent space to $\mathbb{R}^n$ at a, denoted as $\mathbb{R}^n_a$, to be the set $\{(a, v) : v \in \mathbb{R}^n\}$. A geometric tangent vector in $\mathbb{R}^n$ is an element of $\mathbb{R}^n_a$ for some $a \in \mathbb{R}^n$. Clearly, the set $\mathbb{R}^n_a$ is a vector space under standard operations

$$cv_a + cw_a = c(v + w)_a$$

where v_a is a vector starting at a. Thus, with this definition, one could think of the tangent space to a sphere with ambient space $\mathbb{R}^3$ at a point a on the sphere as a space of vectors that are orthogonal to the radial unit vector through a using the inherited natural inner product from $\mathbb{R}^n$. The glaring problem with this definition is that in an arbitrary manifold with non-Euclidean ambient space, we have no idea how to take an inner product or how to define orthogonality.

Using the directional derivative of functions as an inspiration, we call a map $w : C^\infty(\mathbb{R}^n) \to \mathbb{R}$ as a *derivation* at a if it is linear over $\mathbb{R}$ and satisfies the following:

$$w(fg) = f(a)w(g) + g(a)w(f).$$

Let $T_a\mathbb{R}^n$ denote the set of all derivations of $C^\infty(\mathbb{R}^n)$ at a. It can be seen that $T_a\mathbb{R}^n$ is a vector space under the operations

$$(w_1 + w_2)(f) = w_1(f) + w_2(f).$$

Lemma 7.4.1. *Suppose $a \in \mathbb{R}^n$, $w \in T_a\mathbb{R}^n$, and $f, g \in C^\infty(\mathbb{R}^n)$.*

(a) If f is a constant function, then $w(f) = 0$.

(b) If $f(a) = g(a) = 0$, then $w(fg) = 0$.

Proof. (a) Let $f_1(x) \equiv 1$ then $f(x) \equiv cf_1$. We have

$$w(f_1) = w(f_1 f_1) = f_1 w(f_1) + f_1 w(f_1) = 2w(f_1),$$

which implies that $w(f_1) = 0$. It follows that $w(f) = w(cf_1) = cw(f_1) = 0$.

(b) By the product rule,

$$w(fg) = f(a)w(g) + g(a)w(f) = 0.$$

□

Proposition 7.4.1. *Let $a \in \mathbb{R}^n$.*

(a) For each geometric tangent vector $v_a \in \mathbb{R}^n_a$, the map $D_{v|_a} : C^\infty(\mathbb{R}^n) \to \mathbb{R}$ defined as

$$D_{v|_a} f = D_v f(a) = \frac{d}{dt}\bigg|_{t=0} f(a + tv)$$

is a derivation at a.

(b) The map $v_a \to D_{v|_a}$ is an isomorphism from $\mathbb{R}^n_a$ onto $T_a\mathbb{R}^n$.

Proof. (a) This is immediate from the definition of derivative.

(b) Note that the map is linear. Suppose $v_a \in \mathbb{R}^n_a$ such that $D_{v|_a}$ is the zero derivation. Let $v_a = v^i e_i|_a$ in Einstein notation ($v^i e_i|_a := \sum_i v^i e_i|_a$) where x^i means the i-th component of x and e_i is the standard i-th basis. Let f be the $x^j : \mathbb{R}^n \to \mathbb{R}$ smooth coordinate function on $\mathbb{R}^n$ to obtain

$$D_{v|_a}(x^j) \overset{\text{chain rule}}{=} v^i \frac{\partial}{\partial}(x^j)\bigg|_{x=a} = v^j = 0.$$

It follows that v_a is the zero vector so the map is injective.

To prove surjectivity, let $w \in T_a\mathbb{R}^n$ be arbitrary. Let $v = v^i e_i$, where $v^1, \ldots, v^n \in \mathbb{R}$ are given by $v^i = w(x^i)$. We will show that $w = D_{v|_a}$. Let f be a smooth real-valued function on $\mathbb{R}^n$, by Taylor's theorem, we have

$$\begin{aligned} f(x) =& f(a) + \sum_{i=1}^{n} \frac{\partial f}{\partial x^i}(a)(x^i - a^i) \\ &+ \sum_{i,j=1}^{n} (x^i - a^i)(x^j - a^j) \int_0^1 (1-t) \frac{\partial^2 f}{\partial x^i \partial x^j}(a + t(x-a))dt. \end{aligned}$$

Then,

$$\begin{aligned} w(f) &= w(f(a)) + \sum_{i=1}^{n} w(\frac{\partial f}{\partial x^i}(a)(x^i - a^i)) \\ &= \sum_{i=1}^{n} \frac{\partial f}{\partial x^i}(a)(w(x^i) - w(a^i)) \\ &= \sum_{i=1}^{n} \frac{\partial f}{\partial x^i}(a) v^i = D_{v|_a} f. \end{aligned}$$

□

Corollary 7.4.1. For any $a \in \mathbb{R}^n$, the n derivations

$$\left.\frac{\partial}{\partial x^i}\right|_a, \ldots, \left.\frac{\partial}{\partial x^n}\right|_a$$

defined by

$$\left.\frac{\partial}{\partial x^i}\right|_a (f) = \frac{\partial f}{\partial x^i}(a)$$

form a basis for $T_a\mathbb{R}^n$, which therefore has dimension n.

Now we can define tangent vectors on an arbitrary manifold. Let M be a manifold, and let p be a point of M. A linear map $v : C^\infty(M) \to \mathbb{R}$ is called a *derivation* at p if it satisfies

$$v(fg) = f(p)v(g) + g(p)v(f) \quad \text{for all } f, g \in C^\infty(M).$$

Definition 7.4.1. (Tangent space to a manifold at a point) The set of all derivations of $C^\infty(M)$ at p, denoted as T_pM, is a vector space called the *tangent space* to M at p. An element of T_pM is called a *tangent vector* at p.

The tangent vectors on manifolds have the same properties as that of Lemma 7.4.1. Recall that in Euclidean space, the total derivative of a smooth map at a point is a linear map that represents the best linear approximation to the map near the given point. In the case of manifolds, we will talk about the linear map between tangent spaces induced by smooth maps.

Definition 7.4.2. (Differential) Let M, N be manifolds and $F : M \to N$ a smooth map, for each $p \in M$ define a map

$$dF_p : T_pM \to T_{F(p)}N,$$

called the *differential* of F at p. Given $v \in T_pM$, we let $dF_p(v)$ be the derivation at $F(p)$ that acts on $f \in C^\infty(N)$ by the rule

$$dF_p(v)(f) = v(f \circ F).$$

Note that $f \circ F \in C^\infty(M)$ so $v(f \circ F)$ is defined. The operator $dF_p(v) : C^\infty(N) \to \mathbb{R}$ is linear because v is, and is a derivation at $F(p)$ because for any $f, g \in C^\infty(N)$, we have

$$\begin{aligned} dF_p(v)(fg) &= v((fg) \circ F) = v((f \circ F)(g \circ F)) \\ &= (f \circ F)(p)v(g \circ F) + (g \circ F)(p)v(f \circ F) \\ &= (f \circ F)(p)dF_p(v)(g) + (g \circ F)(p)dF_p(v)(f). \end{aligned}$$

Proposition 7.4.2. *Let M, N, and P be manifolds, let $F : M \to N$ and $G : N \to P$ be smooth maps, and let $p \in M$.*

(a) $dF_p : T_pM \to T_{F(p)}N$ is linear.

(b) $d(G \circ F)_p = dG_{F(p)} \circ dF_p : T_pM \to T_{G\circ F(p)}P$.

(c) $d(\mathrm{Id}_M)_p = \mathrm{Id}_{T_pM} : T_pM \to T_pM$.

(d) If F is a diffeomorphism, then $dF_p : T_pM \to T_{F(p)}N$ is an isomorphism, and $(dF_p)^{-1} = d(F^{-1})_{F(p)}$.

Using corollary 7.4.1, we have preimages of the derivations (basis) $\partial/\partial x^1|_{\varphi(p)}, \ldots, \partial/\partial x^n|_{\varphi(p)}$ of $T_{\varphi(p)}\mathbb{R}^n$ under the isomorphism $d\varphi_p : T_pM \to T_{\varphi(p)}\mathbb{R}^n$ form a basis for T_pM.

Let $F : M \to N$ be a smooth map between manifolds, (U, φ) a smooth chart of M containing p and (V, ψ) a smooth chart of N containing $F(p)$, we have $\hat{F} = \psi \circ F \circ \varphi^{-1}$, $\hat{p} = \varphi(p)$ then

$$dF_p\left(\left.\frac{\partial}{\partial x^i}\right|_p\right) = dF_p\left(d(\varphi^{-1})_{\hat{p}}\left(\left.\frac{\partial}{\partial x^i}\right|_{\hat{p}}\right)\right) = \frac{\partial \hat{F}^j}{\partial x^i}(\hat{p})\left.\frac{\partial}{\partial y^j}\right|_{F(p)}$$

where (x^i), (y^j) is the local coordinate representation of the domain, codomain of F. The matrix of dF_p in terms of the coordinate basis or the Jacobian matrix of F is

$$\begin{pmatrix} \frac{\partial F^1}{\partial x^1}(p) & \dots & \frac{\partial F^1}{\partial x^n}(p) \\ \vdots & \ddots & \vdots \\ \frac{\partial F^m}{\partial x^1}(p) & \dots & \frac{\partial F^m}{\partial x^n}(p) \end{pmatrix}.$$

Definition 7.4.3. (Tangent bundle) Given a manifold M, we define the tangent bundle of M, denoted by TM by the disjoint union of the tangent spaces at all $p \in M$:

$$TM = \bigsqcup_{p \in M} T_p M$$

For any manifold M, the tangent bundle TM has a natural topology and smooth structure that makes it into a $2n$-dimensional smooth manifold and the projection $\pi : TM \to M$ is smooth.

7.5 Submersions and Embeddings

In this section we will briefly talk about what does it mean for a manifold to be embedded in another manifold since for the purpose of the manifold assumption, we assume that the low dimensional manifold is embedded into a high dimensional Euclidean space.

Definition 7.5.1. (Submersion) A smooth map $F : M \to N$ is called a smooth submersion if its differential is surjective at each point (rank F = dim N). It is called a smooth submersion if its differential is injective at each point (rank F = dim M).

Definition 7.5.2. (Embedding) A smooth immersion $F : M \to N$ that is also a homeomorphism onto $F(M) \subseteq N$ in the subspace topology is called a smooth embedding of M into N.

7.6 Covector fields and the differential of a function

Definition 7.6.1. (Covector) A linear functional $\omega : V \to \mathbb{R}$ with V a finite-dimensional vector space is called a covector. The space of all covectors on V, called the dual space of V, is denoted as V^* and is itself a real vector space.

For a finite-dimensional vector space V with its basis $(E_1, \dots, E_n)$, let $\varepsilon^1, \dots, \varepsilon^n \in V^*$ be the covectors defined by

$$\varepsilon^i(E_j) = \delta^i_j$$

where δ^i_j is the Kronecker delta. Then it is easy to see that $(\varepsilon^1, \dots, \varepsilon^n)$ forms a basis for V^*, called the dual basis to $(E_1, \dots, E_n)$. For example, denote $(e^1, \dots, e^n)$ as the dual basis to the standard basis $(e_1, \dots, e_n)$ in $\mathbb{R}^n$ then for $v \in \mathbb{R}^n$, they can be given as

$$e^i(v) = e^i(v^1, \dots, v^n) = v^1 e^i(e_1) + \dots + v^n e^i(e_n) = v^i.$$

Thus, in general, if (E_j) is a basis for V and (ε^i) its dual basis then for any $v = \sum_j v^j E_j := v^j E_j$, we have

$$\varepsilon^i(v) := \varepsilon^i(v^j E_j) = \varepsilon^i \left(\sum_j v^j E_j \right) = \sum_j v^j \varepsilon^i(E_j) = \sum_j v^j \delta^i_j = v^i.$$

Furthermore, an arbitrary covector $\omega \in V^*$ can be expressed in terms of the dual basis as

$$\omega = \omega(E_i)\varepsilon^i = \omega_i \varepsilon^i$$

and

$$\omega(v) = \omega(v^j E_j) = \omega_i \varepsilon^i(v^j E_j) = \omega_i v^i.$$

Let M be a manifold. For each $p \in M$, define the *cotangent space at* p to the tangent space T_pM to be $T_p^*M = (T_pM)^*$. Elements of T_p^*M are called *tangent covectors at* p. Similar to the tangent bundle, denote the *cotangent bundle of* M as

$$T^*M = \bigsqcup_{p \in M} T_p^*M.$$

It has a natural projection map $\pi : T^*M \to M$ sending $\omega \in T_p^*M$ to $p \in M$. Given any smooth local coordinates (x^i) on $U \subseteq M$, for each $p \in U$ we denote the basis for T_p^*M dual to $(\partial/\partial x^i|_p)$ (the basis of T_pM) by $(\lambda^i|_p)$. This defines n maps $\lambda^i, \dots, \lambda^n : U \to T^*M$, called *coordinate covector fields.* Given $p \in U$, $\lambda^i|_p \in T_p^*M$ picks out the i-th component of a tangent vector at p.

Definition 7.6.2. (Section) If $\pi : M \to N$ is any continuous map, a section of π is a continuous right inverse for π, i.e, a continuous map $\sigma : N \to M$ such that $\pi \circ \sigma = \mathrm{Id}_N$.

A section of T^*M is called a *covector field.* Similarly, a section of the map $\pi : TM \to M$ is called a *vector field on M.* A vector field is called *smooth* if it is smooth as map from M to TM and the same definition applies for covector fields but from M to T^*M. A (co)vector field that is not necessarily smooth is called a rough (co)vector field.

In any smooth local coordinates on an open subset $U \subseteq M$, a (rough) covector field ω can be written in terms of the coordinate covector fields (λ^i) as $\omega = \omega_i \lambda^i$ for n functions $\omega_i : U \to \mathbb{R}$ called the *component functions of* ω characterized by

$$\omega_i(p) = \omega_p \left(\left. \frac{\partial}{\partial x^i} \right|_p \right) = \omega(p) \left(\left. \frac{\partial}{\partial x^i} \right|_p \right)$$

where $w_p \in T_p^*M$.

If ω is a (rough) covector field and X is a vector field on M, then we can form a function $\omega(X) : M \to \mathbb{R}$ by

$$\omega(X)(p) = \omega_p(X_p)$$

for $p \in M$.

The most important application of covector fields is providing a way to interpret partial derivatives as their components.

Let f be a smooth real-valued function on a manifold M. Define a covector field df, called the *differential of f* by

$$df_p(v) = v(f) \quad \text{for } v \in T_pM.$$

Let (x^i) be smooth coordinates on an open subset $U \subseteq M$, and let $(\lambda^i|_p)$ be the basis for T_p^*M. Write df in coordinates as

$$df_p = A_i(p)\lambda^i|_p$$

for some component functions $A_i : U \to \mathbb{R}$, then we have

$$df_p \left(\frac{\partial}{\partial x^i}\bigg|_p \right) = A_i(p)\lambda^i|_p \left(\frac{\partial}{\partial x^i}\bigg|_p \right) = A_i(p).$$

By definition of df, it follows that

$$A_i(p) = df_p \left(\frac{\partial}{\partial x^i}\bigg|_p \right) = \frac{\partial f}{\partial x^i}(p)$$

and

$$df_p = \frac{\partial f}{\partial x^i}(p)\lambda^i|_p.$$

Thus, the component functions of df in any smooth coordinate chart are partial derivatives of f w.r.t those coordinates. Because of this, we can think of df as an analogue of the classical gradient in $\mathbb{R}^n$ in a coordinate-independent way. Let $f = x^j$ be one of the coordinate functions then we obtain

$$dx^j|_p = \frac{\partial x^j}{\partial x^i}(p)\lambda^i|_p = \delta_i^j \lambda^i|_p = \lambda^j|_p.$$

Hence, the coordinate covector field is the differential of the coordinate functions, dx^j, and

$$df = \frac{df}{dx}dx.$$

It can be shown that df satisfies all of the usual properties of the ordinary derivative. Furthermore, let

$$\Delta f = f(p+v) - f(p)$$

for $v \in \mathbb{R}^n$ then by Taylor's theorem,

$$\Delta f \approx \frac{\partial f}{\partial x^i}(p)v^i = df_p(v).$$

This shows that df_p is a linear functional that best approximates Δf near p.

In section 7.4, we defined df_p as a linear map from T_pM to $T_{f(p)}\mathbb{R}$ but in this section, we defined it as a covector on T_pM, a linear map from T_pM to $\mathbb{R}$ but they are the same because one can easily identify $T_p\mathbb{R}$ with $\mathbb{R}$.

7.7 Tensors

Definition 7.7.1. (Multilinearity) Suppose $V_1, \dots, V_k$ and W are vector spaces. A map $F : V_1 \times \dots V_k \to W$ is said to be multilinear if it is linear as a function of each variable separately when the others are fixed.

Define $L(V_1, \dots, V_k; W)$ as the set of all multilinear maps from $V_1 \times \cdots \times V_k$ to W, this is a vector space under pointwise addition and scalar multiplication. Let $F_1, \dots, F_l \in L(V_1, \dots, V_k; W)$ depending on $n_1, \dots, n_l$ variables then their *tensor product* $F_1 \otimes \cdots \otimes F_l$ is a multilinear function of $n = n_1 + \cdots + n_l$ variables such that

$$F_1 \otimes \cdots \otimes F_l(v_1^{n_1}, \dots, v_{n_1}^{n_1}; v_1^{n_2}, \dots, v_{n_2}^{n_2}; \dots; \dots, v_{n_l}^{n_l}) = F_1(v_1^{n_1}, \dots, v_{n_1}^{n_1}) \dots F_l(v_1^{n_l}, \dots, v_{n_l}^{n_l}).$$

Let V be a finite dimensional vector space. If k is a positive integer, a *covariant k-tensor on V*, α, is an element of $L(V1, \dots, V_k; \mathbb{R})$, which can be thought of as a real-valued multilinear function of k elements of V:

$$\alpha : \underbrace{V \times \cdots \times V}_{k \text{ times}} \to \mathbb{R}.$$

We will denote $T^k(V^*)$ as $L(V1, \dots, V_k; \mathbb{R})$. A covariant k-tensor α on V is said to be *symmetric* if its value is unchanged by interchanging any pair of arguments:

$$\alpha(v_1, \dots, v_i, \dots, v_j, \dots, v_k) = \alpha(v_1, \dots, v_j, \dots, v_i, \dots, v_k)$$

whenever $1 \leq i < j \leq k$. Denote the subspace of all symmetric k-tensors as $\Sigma^k(V^*)$. Let S_k be the symmetric group on k elements, α a k-tensor, we then define a new k-tensor σ_α by

$$\sigma_\alpha(v_1, \dots, v_k) = \alpha(v_{\sigma(1)}, \dots, v_{\sigma(k)}).$$

We define a projection Sym $: T^k(V^*) \to \Sigma^k(V^*)$ called symmetrization by

$$\text{Sym } \alpha = (\text{Sym } \alpha)(v_1, \dots, v_k) = \frac{1}{k!} \sum_{\sigma \in S_k} \alpha(v_{\sigma(1)}, \dots, v_{\sigma(k)}) = \frac{1}{k!} \sum_{\sigma \in S_k} \sigma_\alpha.$$

If $\alpha \in \Sigma^k(V^*)$ and $\beta \in \Sigma^l(V^*)$, we define their symmetric product to be the $(k+l)$-tensor $\alpha\beta$ given by

$$\alpha\beta = \text{Sym } (\alpha \otimes \beta) = \frac{1}{(k+l)!} \sum_{\sigma \in S_{k+l}} \alpha(v_{\sigma(1)}, \dots v_{\sigma(k)}) \beta(v_{\sigma(k+1)}, \dots, v_{\sigma(k+l)}).$$

Proposition 7.7.1. *(a) The symmetric product is symmetric and billinear: for all symmetric tensors α, β, γ and all $a, b \in \mathbb{R}$,*

$$\alpha\beta = \beta\alpha,$$

$$(a\alpha + b\beta)\gamma = a\alpha\gamma + b\beta\gamma = \gamma(a\alpha + b\beta).$$

(b) If α and β are covectors, then

$$\alpha\beta = \frac{1}{2}(\alpha \otimes \beta + \beta \otimes \alpha).$$

Alternatively, a covariant k-tensor α on V is said to be *alternating* if it changes sign whenever two of its arguments are interchanged:

$$\alpha(v_1, \dots, v_i, \dots, v_j, \dots, v_k) = -\alpha(v_1, \dots, v_j, \dots, v_i, \dots, v_k).$$

The subspace of all alternating covariant k-tensors on V is denoted as $\Lambda^k(V^*) \subset T^k(V^*)$.

Let M be a manifold, we then define the bundle of covariant k-tensors on M by

$$T^kT^*M = \bigsqcup_{p \in M} T^k(T_p^*M)$$

and a tensor field is a smooth section of the canonical projection of the bundle. Let $\alpha : M \to T^kT^*M$ be a k-tensor field then $\alpha|_p \in T^k(T_p^*M) = L(\underbrace{T_pM, \dots, T_pM}_{k \text{ times}}; \mathbb{R})$.

7.8 Riemannian Manifold

To be able to define geometric concepts such as lengths, angles, and distances on smooth manifolds, it is essential to introduce the structure of Riemannian metric.

Definition 7.8.1. (Riemannian manifold) Let M be a manifold, a Riemannian metric on M is a smooth symmetric covariant 2-tensor field on M that is positive definite at each point. A Riemannian manifold is a pair (M, g), where M is a manifold and g a Riemannian metric on M.

If g is a Riemannian metric on M, then for each $p \in M$, the 2-tensor g_p is an inner product on T_pM. Because of this, we shall use the notation $\langle v, w \rangle_g$ to denote $g_p(v, w)$ for $v, w \in T_pM$.

In any smooth local coordinates (x^i) of an open subset U of M, a Riemannian metric g can be written

$$g = g_{ij} dx^i \otimes dx^j$$

where $g_{ij} = g\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) : U \to \mathbb{R}$. The symmetry of g allows us to write g as

$$\begin{aligned} g &= g_{ij} dx^i \otimes dx^j \\ &= \frac{1}{2}(g_{ij} dx^i \otimes dx^j + g_{ji} dx^i \otimes dx^j) \\ &= g_{ij} dx^i dx^j. \end{aligned}$$

Example 7.8.1. (Euclidean metric) Let $g = \delta_{ij} dx^i dx^j$, $p \in U \subseteq \mathbb{R}^n$ and $v, w \in T_pM$ then

$$g_p(v, w) = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_{ij} dx^i|_p dx^j|_p(v, w) = \sum_{i=1}^{n} dx^i|_p(v) dx^i|_p(w) = \sum_{i=1}^{n} v^i w^i = v \cdot w$$

which is the Euclidean dot product.

The length or norm of $v \in T_pM$ is defined to be

$$|v|_g = \langle v, v\rangle_g^{1/2}$$

and the angle between two nonzero tangent vectors $v, w \in T_pM$ is a unique $\theta \in [0, \pi]$ satisfying

$$\cos\theta = \frac{\langle v, w\rangle_g}{|v|_g |w|_g}.$$

One of the most important tools a Riemannian metric gives us is the ability to define lengths of curves. If M is a manifold, we define a curve in M to be a continuous map $\gamma : J \to M$, where $J \subseteq \mathbb{R}$ is an interval. Let $t_0 \in J$, define the velocity of γ at t_0, denoted by $\gamma'(t_0)$, to be the vector

$$\gamma'(t_0) = d\gamma \left(\left. \frac{d}{dt} \right|_{t_0} \right) \in T_{\gamma(t_0)}M,$$

where $d/dt|_{t_0}$ is the standard coordinate basis vector in $T_{t_0}\mathbb{R}$.

Let (U, φ) be a smooth chart with coordinate functions (x^i). If $\gamma(y_0) \in U$, we can write the coordinate representation of γ as $(\gamma^1(t), \ldots, \gamma^n(t))$ then we have

$$\gamma'(t_0) = \frac{d\gamma^i}{dt}(t_0) \left. \frac{\partial}{\partial x^i} \right|_{\gamma(t_0)}$$

which is essentially the same formula as it would be in Euclidean space.

Given a curve $\gamma : [a, b] \to M$, the length of γ is

$$\mathrm{L}_g(\gamma) = \int_a^b |\gamma'(t)|_g dt.$$

The integral is well-defined because $|\gamma'(t)|_g$ is continuous for all but finitely many values of t, and has well defined limits from left, right end points.

Proposition 7.8.1. *Let (M, g) be a Riemannian manifold, and let $\gamma : [a, b] \to M$ be a piecewise smooth curve segment. If $\tilde{\gamma}$ is a reparametrization of γ, then $L_g(\tilde{\gamma}) = L_g(\gamma)$.*

Using curve segments as "measuring tapes", we can define distance between two points on a (connected) Riemannian manifold by letting it be the infinum of the lengths of all piece-wise smooth curve segments between two points.

Definition 7.8.2. (Gradient) Define the gradient of a smooth real-valued function f on (M, g) as a unique vector field that satisfies

$$\langle \nabla f, X\rangle_g = X(f) \quad \text{for every vector field } X$$

or equivalently,

$$\langle \nabla f, \cdot\rangle_g = df.$$

8 Manifold Regularization and Laplacian SVM

Regularization is a technique used to constraint a model to prevent overfitting. Manifold regularization is a somewhat similar process where you apply the regularization constraints with respect to the underlying manifold. We will first discuss the idea and theory behind some regularization techniques such as L_2 regularization, commonly known as ridge regression, then connect it to manifold regularization and Laplacian SVM.

8.1 Regularization

Given an $m \times n$ data matrix A with y a vector of labels associated with each row of A. In a usual Machine Learning setting, one would like to find a weight vector w such that $Aw = y$ as a model to predict new data. But unfortunately, more often than not, the linear system $Aw = y$ is ill-posed so we can not solve for an exact w. The obvious approach to this problem is to find w such that the (squared) error $\|Aw - b\|_2^2$ is minimized.

Let $f = w^T x$ be a linear hypothesis function for a certain linear regression problem. Note that if w is large then f would be sensitive to small perturbations in the input and thus, become undesirable as a classifier. To mitigate this problem, one would want to control the size of w by adding a regularization term, such as $\|w\|^2$, to the objective $\|Aw - b\|_2^2$. Our objective problem, often called ridge regression, is

$$\text{minimize} \quad \|Aw - y\|_2^2 + K \|w\|_2^2 \, ,$$

where $K \in \mathbb{R}$ controls the complexity of the regularizer. As $K \to \infty$, the more important the regularization term is and vice versa. The motivation and full details of the inner workings of this regularization method is the trade off between bias and variance which can be further studied in [30].

8.2 Manifold Regularization

Expanding the idea in the previous chapter to take into account the intrinsic geometry of the data, we arrive at manifold regularization framework. Recall that in Machine Learning, the manifold assumption is an assumption that the data from an input space X, usually $\mathbb{R}^n$, only lies within a low dimensional manifold $M \subset X$ which geometry will be used as a regularization term.

Let $X = \{x_1, \ldots, x_n\}$ be a dataset of n samples with each x_i drawn from a marginal distribution P_X with only l of which are labeled and drawn from the conditional distribution $P(y \mid x)$. The $n - l = u$ unlabeled samples give additional information about the marginal distribution. Thus, there need to be an identifiable relation between the conditional and marginal for the model to work. To that end, we make a specific assumption that if two points $x_1, x_2 \in X$ are close in the intrinsic geometry of $P(X)$, then the conditional distribution $P(y \mid x_1)$ and $P(y \mid x_2)$ are similar. We now proceed to a concrete formulation of the regularization problem given by Belkin *et al* [31].

For a Mercer kernel $K : X \times X \to \mathbb{R}$, there is an associated RKHS $\mathcal{H}$ of functions $f : X \to \mathbb{R}$ with the corresponding norm $\| \, \|_{\mathcal{H}}$. Given a set of labeled examples (x_i, y_i), $i = 1, \ldots, l$ the standard framework estimated an unknown hypothesis function by minimizing

$$f^* = \operatorname{argmin} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma \, \|f\|_{\mathcal{H}}^2 ,$$

where V is some loss function i.e a multivariate function that maps events or values to $\mathbb{R}$ representing the cost associated with that event. The regularization term $\|f\|_{\mathcal{H}}^2$ can be thought of as an analogue to $\|w\|_2^2$ in L_2 regularization.

The goal is to extend this framework by incorporating additional information about the geometric structure of P_X. We would like to ensure that the solution is smooth with respect to both the ambient space and marginal P_X. To achieve that, we introduce the intrinsic regularizer:

$$f^* = \operatorname{argmin} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \, \|f\|_{\mathcal{H}}^2 + \gamma_I \, \|f\|_I^2 ,$$

where $\|f\|_I^2$ is an appropriate penalty term that should reflect the intrinsic structure of P_X. The two constants γ_A and γ_I control the complexity of the regularizers in the ambient space and the intrinsic structure respectively.

In most cases, P_X is unknown so we must get an empirical estimates of P_X and $\| \, \|_I$. Note that it is sufficient to have unlabeled examples for the estimates.

Remark. The RKHS norm $\| \, \|_{\mathcal{H}}$ encodes different notions of smoothness depends on the kernels. For example, let $f(x) = wx$ be a line in $\mathbb{R}^2$, and the kernel $K(x, y) = \langle x, y \rangle$ be the standard inner product, it is clear that K is a symmetric, positive definite kernel with an associated RKHS then we have

$$\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} = w^2.$$

Imagine we have 2 classes of data separated by $f(x)$, as the slope w increases, the differences between them becomes finer which is prone to overfitting and vice versa.

Definition 8.2.1. (Support of a probability distribution) Without going too much into measure theory, we simply define the support of a distribution P as the smallest closed set R such that the probability of an event in R is not zero.

Assume that the support of P_X is a compact Riemannian submanifold $\mathcal{M} \subset \mathbb{R}^n$. One natural choice for $\|f\|_I$ is the Dirichlet energy

$$\int_{x \in \mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 \, dP_X(x),$$

which uses the gradient $\nabla_{\mathcal{M}}$ to measure how varied a function is so it makes sense as a candidate to minimize overfitting.

Remark. We include both the intrinsic and ambient regularizer to trade off when the manifold assumption does not hold.

By Stokes' theorem, we have

$$\int_{x \in \mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 \, dP_X(x) = \int_{x \in \mathcal{M}} f(x) \Delta_{\mathcal{M}} f(x) dP_X(x)$$

where $\Delta_{\mathcal{M}}$ is the Laplace-Beltrami operator which generalizes the Laplacian on the Euclidean space. But as mentioned before, we can not compute the intrinsic smoothness penalty because we do not know the manifold $\mathcal{M}$ and the embedding of it onto Euclidean space. Thus, we must, somehow, estimate it. It turns out that the graph Laplacian can be used as a discrete counterpart of the Laplacian $\Delta_{\mathcal{M}}$.

Definition 8.2.2. (Graph Laplacian) Given a simple weighted graph $G = (V, E, W)$ where V, E are the sets of vertices and edges respectively, and W the set of weights associated to each edge. Define the graph Laplacian L as

$$L_{ij} = D_{ii} - W_{ij},$$

where $D_{ii} = \sum_j W_{ij}$.

Given a dataset X, we can construct the graph Laplacian on it using the algorithm given by Belkin *et al* [32]:

1. Construct an adjacency graph: Node i and j are connected by an edge if $\|x_i - x_j\|^2 < \epsilon$ with the Euclidean norm.
2. For each edge ij, assign to it a weight

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\epsilon}}$$

3. Construct the graph Laplacian $L = D - W$.

The graph Laplacian can be used as an operator on a function $f : V \to \mathbb{R}$

$$L(f)(v_i) = \sum_{v_j \sim v_i} (f(v_j) - f(v_i)) W_{ij},$$

where $v_j \sim v_i$ means that they are connected; and as quadratic form

$$\mathbf{f}^T L \mathbf{f} = \sum_{i,j} (\mathbf{f}_i - \mathbf{f}_j)^2 W_{ij},$$

where $\mathbf{f} = [f(v_1)\ f(v_2)\ \dots\ f(v_n)]$. As mentioned before, the relationship between graph Laplacian and the Laplace-Beltrami operator is expressed in the following theorem.

Theorem 8.2.1. *(Belkin-Niyogi [33]) Let the n data points $\{x_1, \dots, x_n\}$ be sampled from the uniform distribution over the embedded Riemannian d-manifold $\mathcal{M}$. Let $\epsilon = n^\alpha$, with $0 < \alpha < \frac{1}{2+d}$. Then for all $f \in C^\infty(\mathcal{M})$, $x \in X$, there exists a constant C, s.t in probability ,*

$$\lim_{n\to\infty} C \frac{\epsilon^{-\frac{d+2}{2}}}{n} L(f)(x) = \Delta_{\mathcal{M}} f(x).$$

Corollary 8.2.1.

$$\mathbf{f}^T L \mathbf{f} = \sum_{i,j} (\mathbf{f}_i - \mathbf{f}_j)^2 W_{ij} \approx \int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 \, dP_X(x).$$

The optimization problem becomes

$$f^* = \operatorname{argmin} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_{\mathcal{H}}^2 + \frac{\gamma_I}{n^2} \mathbf{f}^T L \mathbf{f}. \tag{8.1}$$

Remark. The original theorem involves the volume term vol($\mathcal{M}$) but since $\mathcal{M}$ is compact, the volume is finite and thus, can be replaced with C. The exact details about the volume form on a Riemannian manifold is beyond the scope of this paper, interested readers can find more at [34].

Remark. Define $\text{dist}_{\mathcal{M}}(x, y)$ to be the distance between two points on the Riemannian manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$, called the geodesic. One curious property between the Euclidean distance and the geodesic distance is that for a function $f \in C^\infty(\mathcal{M}) : \mathcal{M} \to \mathbb{R}$, we have [32]

$$\|f(x) - f(y)\| \leq \text{dist}_{\mathcal{M}}(x, y) \, \|\nabla_{\mathcal{M}} f(x)\| .$$

Under the assumption that two points close in the intrinsic structure should have a high probability of being the same class, we then would like to minimize $\|\nabla_{\mathcal{M}} f(x)\|$ since it provides us with an estimate of how far apart f maps nearby points. Thus, we would like find

$$\operatorname{argmin} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f(x)\|^2$$

which returns to our original problem.

It turns out that the optimal solution f^* is a linear combination of the kernel K, which is shown by

Theorem 8.2.2. *(Representer Theorem [35]) Given a nonempty set X, a positive definite real-valued kernel $K : X \times X \to \mathbb{R}$, a training sample $(x_1, y_1), \ldots, (x_m, y_m) \in X \times \mathbb{R}$, a strictly monotonically increasing real-valued function g on $[0, \infty]$, an arbitrary cost function $c : (X \times \mathbb{R}^2)^m \to R \cup \{\infty\}$, and a class of functions*

$$\mathcal{F} = \left\{ f \in \mathbb{R}^X \middle| f(.) = \sum_{i=1}^{\infty} \beta_i K(\cdot, z_i), \beta_i \in \mathbb{R}, z_i \in X, \|f\| < \infty \right\},$$

where $\|\cdot\|$ is the RKHS norm of $\mathcal{H}$ associated with K. Then for any $f \in \mathcal{F}$ minimizing the regularized risk functional

$$c((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))) + g(\|f\|)$$

admits a representation of the form

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i).$$

This theorem is very important because it allows us to discretize the objective problem which will be we will talk about in the next section.

8.3 Laplacian SVM

Recall the primal problem from soft margin SVM is

$$\min_{w, b, \xi} \quad \frac{1}{2}\langle w, w\rangle + C \sum_{i=1}^{n} \xi_i^2$$

which can be extended to a more general form [36] in the RKHS $\mathcal{H}$

$$\min_{f \in \mathcal{H}} \quad \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f(x_i)) + \gamma \|f\|_{\mathcal{H}}^2,$$

where $V(x_i, y_i, f(x_i))$ is the hinge loss defined by

$$V(x_i, y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)),$$

and the labels $y_i \in \{-1, +1\}$. Let $\xi_i = V_i$ and include the intrinsic regularizer to obtain the primal

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^n, \xi \in \mathbb{R}^l} \quad & \frac{1}{l}\sum_{i=1}^{l} \xi_i + \gamma_A \alpha^T K \alpha + \frac{\gamma_I}{n^2} \alpha^T KLK\alpha \\ \text{s.t.} \quad & y_i \left(\sum_{j=1}^{n} \alpha_j K(x_i, x_j) \right) \geq 1 - \xi_i, \quad i = 1, \ldots, l, \\ & \xi_i \geq 0, \quad i = 1, \ldots, l, \end{aligned}$$

where K is the Gram matrix over the labeled points. We then have the Lagrangian

$$\begin{aligned} L(\alpha, \xi, b, \beta, \zeta) =& \frac{1}{l}\sum_{i=1}^{l} \xi_i + \frac{1}{2}\alpha^T \left(2\gamma_A K + 2\frac{\gamma_I}{n^2} KLK \right) \alpha \\ & - \sum_{i=1}^{l} \beta_i \left(y_i \left(\sum_{j=1}^{n} \alpha_j K(x_i, x_j) + b \right) - 1 + \xi_i \right) - \sum_{i=1}^{l} \zeta_i \xi_i \end{aligned}$$

with β, ζ the Lagrange multipliers. It follows that

$$\alpha = \left(2\gamma_A I + \frac{2\gamma_I}{n^2} LK \right)^{-1} J^T Y \beta^*$$

and the dual problem is

$$\begin{aligned} \beta^* = \max_{\beta \in \mathbb{R}^l} \quad & \sum_{i=1}^{l} \beta_i - \frac{1}{2}\beta^T Q \beta \\ \text{s.t.} \quad & \sum_{i=1}^{l} \beta_i y_i = 0, \\ & 0 \leq \beta_i \leq \frac{1}{l} \quad i = 1, \ldots, l \end{aligned}$$

where $J = [I \quad 0]$ is an $l \times n$ matrix with I as the $l \times l$ identity matrix, $Y = \text{diag}(y_1, y_2, \ldots, y_l)$, and

$$Q = YJK \left(2\gamma_A I + \frac{2\gamma_I}{n^2} LK \right)^{-1} J^T Y.$$

The condition $0 \leq \beta_i \leq 1/l$ can be observed by letting $\partial L / \partial \xi_i = 0$ with $\xi_i \geq 0$. Thus we have transformed the abstract optimization problem 8.1 into a computable one using PCG algorithm described in [37].

Remark. There exist other variations of the Laplacian SVM formulation such as SS-LapSVM [38] where the authors create two graph Laplacians to take into account both the spatial and spectral neighborhood since hyperspectral data complies with both spectral and spatial homogeneity.

9 Experiments, Results, and Discussion

9.1 Preprocessing

The experiments were conducted on the Drenthe dataset provided by the NLR. The dataset is composed of multiple raster images usually of dimension $500 \times 500 \times 420$ with 420 spectral bands. Each of these images only contain, on average, one or two roofs that reportedly has asbestos so for each of those roofs, we take a small window that fully contain the area of the roof plus some of the surroundings. Let w, h be the width and height of the new windows to be extracted from the full image, we can obtain the new image shape $(w + 1) \times (h + 1) \times 420$ for each of the original full image.

After windowing the data, the Maximum Noise Fraction (MNF) method was used to reduce the dimension of the data from 420 to 60. The most optimal number of retained spectral bands is still unclear but after several experimentations, it should be above around 50 to 80, depending on the available resources. The MNF was applied by converting all the (windowed) images which were three-dimensional to two dimensional, and later transformed them back for training.

Figure 10: Example hyperspectral image in 3 bands

9.2 Results

All the training data are windowed with a square of width (= height) equals 30 and 50. The (soft margin) support vector machine model was implemented with the Gaussian or Radial Basis Kernel of the form

$$K(x, y) = \exp -\frac{\|x - y\|^2}{2\sigma^2} \quad \text{for } \sigma \in \mathbb{R},$$

with complexity constant C, in 5.3, varies from 1 to 10. It was found that 1 produces the best result. The deep learning model HybridSN were trained using Adam [39] optimizer with learning rate of 0.001, decay of 10^{-6}. It was trained for 10 epochs with batch size of 128, and binary cross entropy

$$J(\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i}),$$

for $\hat{y_i}$ the predicted probability of y_i, as a continuously differentiable loss function. The results from there training are shown in Table 9.1. Because of the varying nature of the data, we let the model train 10 times on m training images randomly sampled from the dataset, then take the average to get the most reliable assessment. The four metrics that are used are Overall Accuracy (OA), Average Accuracy (AA), Cohen's Kappa, and Jaccard Similarity (mIoU). Out of four of them, the Jaccard similarity is the most important one because we are dealing with a class imbalance situation which OA and AA are vulnerable to.

Window size	Training samples	Methods	OA	AA	Kappa	mIoU
30	10%	SVM	0.75	0.76	0.46	0.51
		HybridSN	**0.88**	**0.86**	**0.70**	**0.65**
	30%	SVM	**0.83**	**0.83**	**0.65**	**0.66**
		HybridSN	0.82	0.82	0.63	0.65
50	10%	SVM	0.81	0.69	0.35	0.30
		HybridSN	**0.87**	**0.71**	**0.48**	**0.38**
	30%	SVM	**0.85**	**0.74**	**0.50**	**0.42**
		HybridSN	0.82	0.68	0.37	0.32

Table 9.1: Results of SVM and HybridSN

It can be readily seen that by increase the window size, the Jaccard Similarity (mIoU) reduces significantly because by doing so, the effects of class imbalance are amplified, and other pixels that may or may not be similar to the asbestos ones, are included but are labeled as non-asbestos thus confuses the model. Furthermore, both the SVM and HybridSN models are still prone to classifying non-asbestos roof as asbestos when given a picture with (allegedly) non-asbestos roofs.

The LapSVM method implementation was taken from [40] with a few modifications. The best parameters for γ_A and γ_I were found to be 10^{-5} and 10 respectively. The best number of points to be randomly unlabeled for each class is roughly 20. The model is evaluated with respect to the Euclidean and Cosine metrics with binary, and heat kernel weights

$$W_{ij} = \exp\frac{-\|x_i - x_j\|^2}{2\sigma^2}.$$

Because of the limitations of time and resources, we will only train LapSVM with data of window size 30.

Training samples	Metrics	Weights	OA	AA	Kappa	mIoU
10%	Euclidean	Binary	0.58	0.57	0.14	0.31
		Heat Kernel	**0.63**	**0.64**	**0.27**	**0.40**
	Cosine	Binary	**0.61**	**0.63**	**0.25**	**0.42**
		Heat Kernel	0.59	0.58	0.16	0.32
30%	Euclidean	Binary	0.49	0.50	0.2	0.29
		Heat Kernel	**0.67**	**0.66**	**0.32**	**0.43**
	Cosine	Binary	**0.62**	**0.63**	**0.24**	**0.40**
		Heat Kernel	0.52	0.54	0.08	0.34

For concrete pictorial results on test images, please consult the Appendix 12.

10 Conclusion

Performance wise, SVM and HybridSN are somewhat comparable but HybridSN training speed is much faster than both SVM and LapSVM since it utilizes GPU. LapSVM's performance was relatively underwhelming but predictable since it is in its basic form; with best metrics and weights for LapSVM seems to be Cosine and binary weight or Euclidean and heat kernel. As the window size increases, the performance for all the models decay because of the class-imbalance and the unknown nature of the non-asbestos pixels. Also, it seems that for pictures of asbestos roof in the middle of a farm or vegetable-based surrounding, the model performs better than urban surrounding. This makes sense because the signature level of vegetables are more distinct than cement and asbestos. Because of the lack of non-asbestos roofs, the three models results indicate that they are only detecting roofs, not necessarily asbestos, at the moment.

Since the signal to noise ratio (SNR) of the data, using the squared of mean over standard deviation, is quite low (≈ 3), this poses a problem for the SVM method because of its sensitivity to noise [41] and class-imbalance data [42]. Other noise resistant methods such as Logistics Regression or Decision Tree may be preferred.

11 Further Research

The implementation of Laplacian SVM is still in its most basic form i.e has not been adapted for hyperspectral data. Thus, it would be important to improve the existing model or implement any existing models such as Gu *et al* [43] that utilizes different metrics. Furthermore, since the data is noisy, using different noise estimation methods outlined in [44] or denoising method such as wavelet denoise [45] could have a positive impact on the outcome of the model. Also, the current model can be easily extended to include non-asbestos roof and improve its capability.

The theories of supervised and semi-supervised is rich and almost endless, in this paper, we have only explained a small fraction of the theories behind Riemannian manifold, Support Vector Machine, PCA, etc. In particular, the SVM method has strong ties to Vapnik-Chervonenkis (VC) theory which had not been mentioned in the paper. Furthermore, PCA, like SVM, can be generalized to include a kernel thus boosting its efficacy.

Bibliography

[1] Marty S. Kanarek. *Mesothelioma from Chrysotile Asbestos: Update.* 2011. DOI: 10.1016/j.annepidem.2011.05.010.

[2] Nick Ubels. “Classifying asbestos roofs in the Dutch province of Drenthe using hyperspectral imagery and deep learning”. 2019.

[3] Fiona Heuff et al. *Handleiding verwachtingskaart asbestdaken Provincie Drenthe.* 2017.

[4] Tanmay Chakraborty and Utkarsh Trehan. *SpectralNET: Exploring Spatial-Spectral WaveletCNN for Hyperspectral Image Classification.* 2021. DOI: 10.48550/ARXIV.2104.00341. URL: https://arxiv.org/abs/2104.00341.

[5] Swalpa Kumar Roy et al. “HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classification”. In: *IEEE Geoscience and Remote Sensing Letters* 17.2 (Feb. 2020), pp. 277–281. DOI: 10.1109/lgrs.2019.2918719. URL: https://doi.org/10.1109%2Flgrs.2019.2918719.

[6] Yanan Luo et al. “HSI-CNN: A Novel Convolution Neural Network for Hyperspectral Image”. In: *2018 International Conference on Audio, Language and Image Processing (ICALIP).* 2018, pp. 464–469. DOI: 10.1109/ICALIP.2018.8455251.

[7] Giuseppe Bonifazi, Giuseppe Capobianco, and Silvia Serranti. “Asbestos containing materials detection and classification by the use of hyperspectral imaging”. In: *Journal of Hazardous Materials* 344 (2018), pp. 981–993. ISSN: 0304-3894. DOI: https://doi.org/10.1016/j.jhazmat.2017.11.056. URL: https://www.sciencedirect.com/science/article/pii/S0304389417308786.

[8] Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe. *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3.* Sept. 2015. DOI: doi:/10.4231/R7RX991C. URL: https://purr.purdue.edu/publications/1947/1.

[9] Małgorzata Krówczyńska et al. “Mapping asbestos-cement roofing with the use of APEX hyperspectral airborne imagery: Karpacz area, Poland – a case study”. In: *Miscellanea Geographica* 20.1 (2016), pp. 41–46. DOI: doi:10.1515/mgrsd-2016-0007. URL: https://doi.org/10.1515/mgrsd-2016-0007.

[10] Xiaopan Wang, Li Ma, and Fujiang Liu. “Laplacian support vector machine for hyperspectral image classification by using manifold learning algorithms”. In: *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS.* 2013, pp. 1027–1030. DOI: 10.1109/IGARSS.2013.6721338.

[11] Andrew A. Green et al. "A transformation for ordering multispectral data in terms of image quality with implications for noise removal". In: *IEEE Transactions on Geoscience and Remote Sensing* 26 (1988), pp. 65–74.

[12] James Lee, A.Stephen Woodyatt, and Mark Berman. "Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform". In: *Geoscience and Remote Sensing, IEEE Transactions on* 28 (June 1990), pp. 295–304. DOI: `10.1109/36.54356`.

[13] Jeremy Watt. URL: `https://jermwatt.github.io/control-notes/posts/zca_sphereing/ZCA_Sphereing.html#PCA-sphereing`.

[14] David Ruiz Hidalgo, Bladimir Bacca Cortés, and Eduardo Caicedo Bravo. "Dimensionality reduction of hyperspectral images of vegetation and crops based on self-organized maps". In: *Information Processing in Agriculture* 8.2 (2021), pp. 310–327. ISSN: 2214-3173. DOI: `https://doi.org/10.1016/j.inpa.2020.07.002`. URL: `https://www.sciencedirect.com/science/article/pii/S221431732030189X`.

[15] Deepesha Burse. *Introduction to CNN.* URL: `https://dev.to/deepeshaburse/introduction-to-cnn-1i03`.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* `http://www.deeplearningbook.org`. MIT Press, 2016.

[17] URL: `https://paperswithcode.com/sota/hyperspectral-image-classification-on-indian?p=hybridsn-exploring-3d-2d-cnn-feature`.

[18] Dirk Jan Struik. *Joseph-Louis Lagrange, comte de l'Empire — French mathematician — Britannica.* en. URL: `https://www.britannica.com/biography/Joseph-Louis-Lagrange-comte-de-lEmpire` (visited on 05/11/2022).

[19] H. W. Kuhn and A. W. Tucker. "Nonlinear programming". In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950.* Berkeley and Los Angeles: University of California Press, 1951, pp. 481–492.

[20] Vladimir N. Vapnik. *An overview of statistical learning theory.* 1999. DOI: `10.1109/72.788640`.

[21] Vivek Salunkhe. *Support Vector Machine (SVM).* 2021. URL: `https://medium.com/@viveksalunkhe80/support-vector-machine-svm-88f360ff5f38`.

[22] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[23] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* 1st ed. Cambridge University Press, 2000. ISBN: 0521780195. URL: `http://www.amazon.com/Introduction-Support-Machines-Kernel-based-Learning/dp/0521780195/ref=sr_1_1?ie=UTF8&s=books&qid=1280243230&sr=8-1`.

[24] Jocelyn Quaintance Jean Gallier. *Algebra, Topology, Differential Calculus, and Optimization Theory For Computer Science and Machine Learning.* 2022. URL: `https://www.cis.upenn.edu/~jean/math-deep.pdf`.

[25] Chih-Chung Chang and Chih-Jen Lin Chih-Wei Hsu. “A Practical Guide to Support Vector Classification”. In: *BJU international* 101 (1 2008). ISSN: 1464-410X.

[26] *Normed Vector Space of Rational Numbers is not Banach Space.* URL: https://proofwiki.org/wiki/Normed_Vector_Space_of_Rational_Numbers_is_not_Banach_Space.

[27] Benyamin Ghojogh et al. *Reproducing Kernel Hilbert Space, Mercer’s Theorem, Eigenfunctions, Nyström Method, and Use of Kernels in Machine Learning: Tutorial and Survey.* 2021. DOI: 10.48550/ARXIV.2106.08443. URL: https://arxiv.org/abs/2106.08443.

[28] N. Aronszajn. “Theory of Reproducing Kernels”. In: *Transactions of the American Mathematical Society* 68 (3 1950). ISSN: 00029947. DOI: 10.2307/1990404.

[29] J.M. Lee and J.M. Lee. *Introduction to Smooth Manifolds.* Graduate Texts in Mathematics. Springer, 2003. ISBN: 9780387954486. URL: https://books.google.nl/books?id=eqfgZtjQceYC.

[30] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[31] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. “Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples”. In: *Journal of Machine Learning Research* 7.85 (2006), pp. 2399–2434. URL: http://jmlr.org/papers/v7/belkin06a.html.

[32] Mikhail Belkin and Partha Niyogi. “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”. In: *Neural Computation* 15.6 (2003), pp. 1373–1396. DOI: 10.1162/089976603321780317.

[33] Mikhail Belkin and Partha Niyogi. “Towards a theoretical foundation for Laplacian-based manifold methods”. In: *Journal of Computer and System Sciences* 74.8 (2008). Learning Theory 2005, pp. 1289–1308. ISSN: 0022-0000. DOI: https://doi.org/10.1016/j.jcss.2007.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0022000007001274.

[34] J.M. Lee. *Riemannian Manifolds: An Introduction to Curvature.* Graduate Texts in Mathematics. Springer New York, 1997. ISBN: 9780387982717. URL: https://books.google.nl/books?id=ZRQgH7FQafgC.

[35] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. “A Generalized Representer Theorem”. In: *Computational Learning Theory.* Ed. by David Helmbold and Bob Williamson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 416–426. ISBN: 978-3-540-44581-4.

[36] Gregory E. Fasshauer, Fred J. Hickernell, and Qi Ye. “Solving support vector machines in reproducing kernel Banach spaces with positive definite functions”. In: *Applied and Computational Harmonic Analysis* 38.1 (2015), pp. 115–139. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2014.03.007. URL: https://www.sciencedirect.com/science/article/pii/S1063520314000475.

[37] Stefano Melacci and Mikhail Belkin. “Laplacian Support Vector Machines Trained in the Primal”. In: *Journal of Machine Learning Research* 12 (Mar. 2011), pp. 1149–1184. ISSN: 1532-4435.

[38] Lixia Yang et al. “Semi-Supervised Hyperspectral Image Classification Using Spatio-Spectral Laplacian Support Vector Machine”. In: *IEEE Geoscience and Remote Sensing Letters* 11.3 (2014), pp. 651–655. DOI: 10.1109/LGRS.2013.2273792.

[39] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[40] Gu Hong Yang. 2019. URL: https://github.com/GuHongyang/LapSVM-python.

[41] Abhinav Atla et al. “Sensitivity of Different Machine Learning Algorithms to Noise”. In: *J. Comput. Sci. Coll.* 26.5 (May 2011), pp. 96–103. ISSN: 1937-4771.

[42] Konstantinos Veropoulos, C. Campbell, and N. Cristianini. “Controlling the Sensitivity of Support Vector Machines”. In: *Proceedings of International Joint Conference Artificial Intelligence* (June 1999).

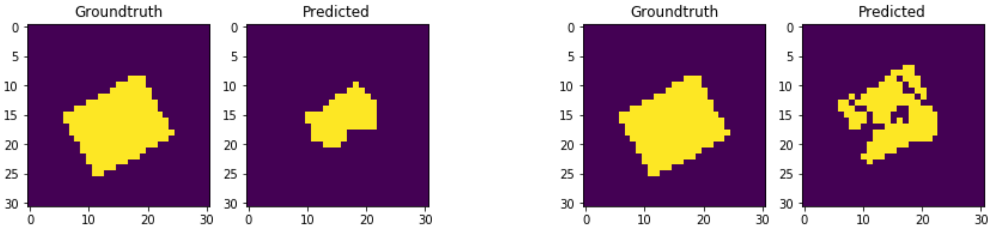
[43] Yanfeng Gu and Kai Feng. “Optimized Laplacian SVM With Distance Metric Learning for Hyperspectral Image Classification”. In: *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* 6 (June 2013), pp. 1109–1117. DOI: 10.1109/JSTARS.2013.2243112.

[44] Lianru Gao et al. “A comparative study on noise estimation for hyperspectral imagery”. In: *2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS).* 2012, pp. 1–4. DOI: 10.1109/WHISPERS.2012.6874262.

[45] Behnood Rasti et al. “Hyperspectral image denoising using 3D wavelets”. In: July 2012, pp. 1349–1352. ISBN: 978-1-4673-1160-1. DOI: 10.1109/IGARSS.2012.6351286.

12 Appendix

The results of HybridSN, LapSVM, and SVM on a test image:



(a) HybridSN Result

(b) SVM Result

Figure 11: Results of HybridSN and SVM on a test image.

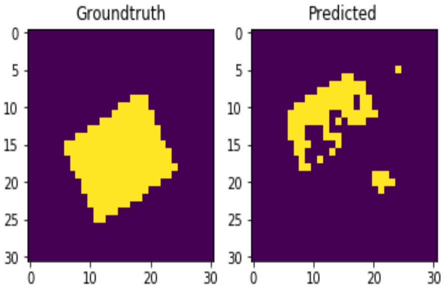
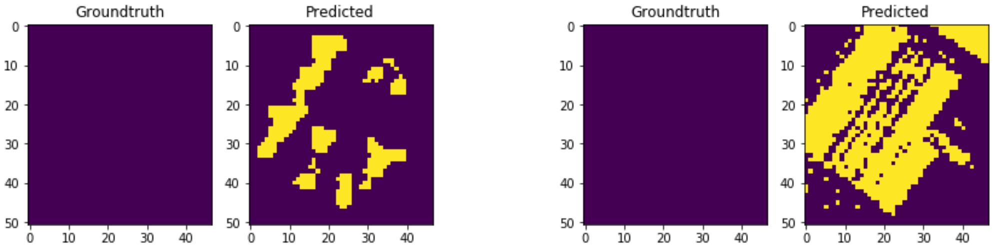


Figure 12: Result of LapSVM on a test image, with heat kernel weight and Euclidean metric.

The results of HybridSN, LapSVM, and SVM on a test image with no asbestos roof:



(a) HybridSN Result

(b) SVM Result

Figure 13: Results of HybridSN and SVM on a test image.

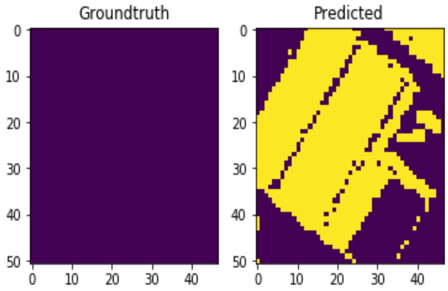


Figure 14: Result of LapSVM on a test image, with binary weight and Euclidean metric.