YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SOFTWARE DEVELOPMENT FOR TRANSITIONS OF GRAPHS FROM DISCRETE STATE INTO THE CONTINUOUS STATE

Çağatay YÜCEL

Thesis Advisor: Assoc. Prof. Ahmet Hasan KOLTUKSUZ, Ph.D.

Department of Computer Engineering

Bornova-İZMİR
June 2012

ÖZET

KESİKLİ GRAFLARIN SÜREKLİ HALE DÖNÜŞTÜRÜLMESİ İÇİN YAZILIM GELİŞTİRME

YÜCEL, Çağatay

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Bölümü
Tez Danışmanı: Doç. Dr. Ahmet Hasan KOLTUKSUZ

Haziran 2012, 75 sayfa

Güncel bilgi modelleri bilgiyi oluşturan kelimelerin ve harflerin frekansları, kelime uzunlukları ve bilginin sıkıştırılması gibi bilginin sözdizimsel özelliklerin incelenmesiyle ilgilenmektedir. Bilginin analizini ve elde edinimini geliştirmek için semantik özellikler üzerinde çalışan yeni hesaplama modelleri tanımlanmalıdır.

Bu çalışmada bilginin ve yeni hesaplama modellerinin tanımlanmasına elverişli yapılar olarak türevlenebilir manifoldlara yer verilmiştir. Tanımları gereği manifoldlar global ölçekte bakıldığında Öklidyen olmayan özellikler gösterirken lokal ölçeklerde öklidyen uzaylara benzemektedirler. Bu özellikleri sayesinde öngörülen yeni modellerin Öklidyen modeller üzerinde çalışan güncel modelleri de kapsaması söz konusudur.

Bilginin bilgisayar bilimlerindeki en yaygın modellerinden biri graf yapılarıdır. Graf yapıları tanımları itibariyle ayrık ve hesaplanabilirlerdir. Bu tezin temel amacı graflardan manifoldlara bağıntılar kurulmasını araştırarak graf olarak tanımlanan bilginin yeni ve sürekli modellere taşınabilirliğini sınamaktır. Bu amaç dahilinde bilginin geometrik özelliklerinin tanımlanmasına bir adım daha yaklaşılmış olacaktır.

Anahtar Kelimeler: Bilgi, Bilginin modellenmesi, Laplacian, Laplace - Beltrami Operatorü, Graf, Manifold, Türevlenebilir Geometri, Öklidyen olmayan geometri.

ABSTRACT

SOFTWARE DEVELOPMENT FOR TRANSITIONS OF GRAPHS FROM DISCRETE STATE INTO THE CONTINOUS STATE

YÜCEL, Çağatay

MSc in Computer Engineering

Supervisor: Assoc. Prof. Ahmet Hasan KOLTUKSUZ, Ph.D.

June 2012, 75 pages

The contemporary information model deals only with syntactics of information, such as frequency of the occurances of characters, length of words and compression amount of documents. Computable models targeting semantic properties of information, such as relations between words, should be defined and studied in order to improve the analysis and the retrieval of information.

Manifolds are suitable differentiable mathematical objects for information to be defined on. By their very definition they are non-euclidean in the global view but in local scales they resemble euclidean spaces. This property provides that the contemporary models can also be defined within the previsioned new models of information models.

One of the most basic representation of information is through graphs. They are discrete and highly computable mathematical bojects. In this thesis, the main aim is to investigate methods of embedding this simple piece of information onto manifolds. This aim is supposed to lead us to defining the geometrical aspects of information.

Keywords: Information, Information Modeling, Laplacian, Laplace - Beltrami Operator, Graph, Manifold, Differential Geometry, Non - Euclidean Geometry.

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor Ahmet Koltuksuz, Ph.D., Head of the Computer Engineering Department of Yaşar University, without his vision and wisdom this thesis wouldn't have been possible. I could never have done all of this studies through my academic life without his teachings and supervision in computer science, physics and mathematics. He always introduces me to unknown areas. Thank you.

I would like to express my sincere gratitude to Selma Tekir, Ph.D. for her guidance, continuous support, patience and motivation. Her wide knowledge and her logical way of thinking have been of great value for me.

My sincere thanks goes to Hüseyin Hışıl, Ph.D. for his detailed and constructive comments, and for his important support throughout this work. He was also in my thesis committee. His truly scientist intutition have helped a lot and I am grateful in every possible ways.

I owe my most sincere gratitude to Mehmet Terziler, Ph.D. Head of the Mathematics Department of Yaşar University, official committee member for his constructive and motivating comments at the defence.

I would like to express my sincere thanks to Serap Atay, Ph.D. for her kindness, loving support and for all the sources she had shared with me. Her encouragement helped me in all the time of my research.

I would like to thank The Scientific and Technological Research Council of TURKEY for the financial support of Master of Science Fellowship Programme.

I gratefully thank Burak Ekici, Erdem Sarılı, Görkem Kılınç and Onur Erbaş for their constructive comments and helps in the development of this thesis. Their kind support have been of great value in this study. I gratefully thank Selen Bodur for her continuous support throughout this study.

I would like to record my gratitude to my roommates and collegues in Yaşar University. I would like to thank Burcu Külahçıoğlu, M.Sc. for her constructive comments during the writing of this study. I owe my sincere thanks to the professors of Computer and Software Engineering departments of Yaşar University.

Finally, I would like to thank my father Hilmi Yücel, M.D. for raising me as a researcher, to him I dedicate this thesis. I would like to thank my brother Çağan Selçuk Yücel, M.Sc. and my mother Muazzez Yücel for their loving support.

TEXT OF OATH

I declare and honestly confirm that my study titled “SOFTWARE DEVELOPMENT FOR TRANSITIONS OF GRAPHS FROM DISCRETE STATE INTO THE CONTINUOUS STATE”, and presented as Master’s Thesis has been written without applying to any assistance inconsistent with scientific ethics and traditions and all sources I have benefited from are listed in bibliography and I have benefited from these sources by means of making references.

20 / 06 / 2012

Çağatay YÜCEL

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

The information model is the representation of information in a way that it can be analyzed, measured, processed and transferred. The contemporary information model can deal only with the syntactics of information, such as frequency of the occurances of characters, length of words and compression percentage of plain texts. The model was introduced by Claude E. Shannon in his 1948 famous paper “A Mathematical Theory of Communication” [5].

In this information model, the definition of information is based on probability theory and statistics. *The Shannon Entropy*, the most striking concept within this model, is given by the quantification of the expected value of information contained in a message. This model contains nothing about the semantics of information. For the semantic properties to be modeled, ontology based semi-automatic information retrieval models have been proposed. These models rely mostly on the human interaction to define the relations between words, in order to derive their meanings [6].

Information Retrieval (IR) is the process of searching specific information either as

- text, sound, image, video, data or metadata in some document, or
- specific documents within a collection.

IR systems are designed with the objective of providing, in response to a query, references to documents that would contain the information desired by the user [7].

In IR systems, documents and queries are represented in a mathematical model where an operation regarding to the closeness of documents are formally defined. There must be a conversion of documents and queries into the element set of the system to retrieve which documents the user should read with respect to the query user provided.

The process begins when user enters a query into the system. The system converts this query into an element in the model and relates it with some other elements with the closeness function of the system. Closeness function f is defined as:

$$f\colon V \times Q \to U \tag{1.1}$$

where V is the mathematical model of document collection, Q is the set of queries for the information needs of the user and U is the subset of V relevant to the query of the user.

The Vector Space Models (VSM's) has been the standard model for information retrieval since 1975. In this model, each unique word or some subset of unique words within document collection represents a dimension in space and called terms. Choosing the terms depends on the application. Each document and query represents a vector within that multi-dimensional space [8].

1.1 Motivation and Aims

VSM terms are assumed to be orthogonal. This assumption, leaves out the semantic relationship between terms. The terms which represents the coordinate system of the document space, can be related and the angles between them can vary depending on the relation instead of being orthogonal. This problem is called “The Problem of Dimensionality” [9]. Regarding the coordinate system as constant is yet another problem in addition to the problem of dimensionality. The angles between terms can vary depending on the document. This variation among documents leads to new document spaces defined by different sets of basis vectors.

The aforementioned problems lead to the assumption that the structure of information is non-linear, and should be defined in continuous mathematical objects instead of vector spaces. Therefore the models related to the manifolds are studied in this research. Manifolds are suitable differentiable mathematical objects for information to be defined on. By their very definition they are non-euclidean in the global view but in local scales they resemble euclidean spaces. AS a consequence, the contemporary models can also be defined within the previsioned new models of information models.

One of the most basic representation of information is through graphs. Graphs are discrete and highly computable. In this thesis, the main aim is to investigate methods of embedding information onto manifolds using graohs. The methodology is constructed as follows;

- The graph should be constructed from points which are believed to be samples from a manifold, so that the geometry of information is preserved.
- The relation between the properties of the graph and the manifold should be defined.
- And finally, the embedding map should be constructed.

Transition of graphs onto manifolds enables a series of applications such as graph matching and dimensionality reduction to be done using graphs along with

the manifold properties. Image, text and sound analysis examples can be found at [3], [2], [4].

For the aim of examining graph embedding methods, python script programming language based software are developed in this thesis. It is important to state that the transition methods can be useful after the non-linear information properties are inputted.

1.2 Outline

The rest of this thesis is structured as follows.

Chapter 2 consists of the definitions of mathematical structures. In this chapter, manifolds and graphs are defined and their properties are presented.

Chapter 3 defines the relation between manifolds and graphs using the Laplacian Operator.

Chapter 4 and Chapter 5 present the graph embedding methods.

Chapter 6, the final chapter, concludes the thesis and summarize future works in the direction of this research.

Chapter 2

Mathematical Background

In this chapter the necessary definitions including manifolds and graphs are given. The structure of this chapter is as follows:

- Vectors, basis vectors, tensors and transformation law is explained briefly.
- The notion of maps, its properties, and more importantly the notion of continuity are stated.
- Definition of coordinate charts, manifolds, and their properties are presented.
- Definition of graphs and properties of graphs are provided.

The notations used in this chapter is from the “Einstein’s Summation Notation” [10].

2.1 Vectors, Basis Vectors, Tensors and Transformation Law

2.1.1 Vectors, Vector Spaces and Vector Fields

In euclidean spaces, vectors are the line elements equipped with a direction. Each vector has a magnitude and a definite direction. A vector can be represented as a

graphical arrow which has an initial and terminal point.

- A vector may possess a constant initial point and terminal point. Such a vector is called a bound vector.
- When only the magnitude and direction of the vector matter and the vector is called a free vector.

Definition 2.1. Let $v_1, v_2, v3$ be vectors and $n_1, n_2, s \in \mathbb{R}$. A vector space over a field F is a set with two binary operations (+,*) satisfying

- $v_1 + (v_2 + v_3) = (v_1 + v_2) + v_3$(Associativity)
- $v_1 + v_2 = v_2 + v_1$ (Commutativity)
- There exists an element $0 \in V$, s.t. $v + 0 = v$ for all $v \in V$ (Identity)
- $s.(v_1 + v_2) = s.v_1 + s.v_2$
- $(n_1 + n_2)v = n_1v + n_2v$
- $n_1.(n_2.s) = (n_1.n_2).s$
- For all $v \in V$, there exists $-v$ s.t. $v + (-v) = 0$ (Inverse)
- For all $s \in F$, $1s = s$, $1 \in F$ is the multiplicative identity

Definition 2.2. Although the terms “scalar field” and “vector field” contains the term “field”, the definitions below should not be mixed up with the algebraic definition of fields.

- A scalar field is an assignment of a scalar to each point in the euclidean subspace.
- A vector field is an assignment of a vector to each point in the euclidean subspace.

2.1.2 Basis Vectors and Vector Expansion on Basis

Definition 2.3. A basis of a vector space is the set of linearly independent vectors which can be used to generate every vector in that space. When the angles between them are not perpendicular, they are called skew-angular basis. Orthogonal otherwise.

Definition 2.4. A coordinate system is a basis complemented with a fixed point called origin.

When our vectors reaching to infinity and perpendicular to each other, the space is called a *Cartesian Coordinate System*. Whenever the angles different than perpendicular, then the space is still called *Euclidean Coordinate System* but the basis is no more orthogonal. If the angles between basis vectors are changing at every point, more precisely if instead of lines as basis vectors, there are curves then the space is said to be in *curvilinear coordinate* system.

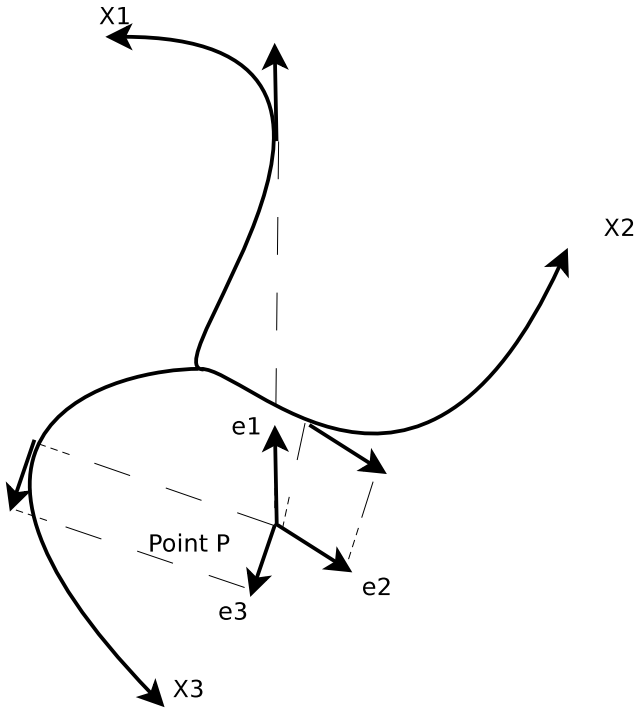
A vector in curvilinear coordinates is not curved as it can be incorrectly interpretted. Instead we have different basis vectors at each point, determined by partial derivatives of the curves at the point. In that case, at every point there exists a vector space called tangent space. Tangent spaces will be detailed in the properties of manifolds in section 2.2.3. The Figure 2.1 illustrates the definition of basis vectors.

Let $e_1, e_2, \ldots, e_n$ be the basis vectors and $a_1, a_2, \ldots, a_n$ be the coefficients of the components of a vector. Once we have the basis vectors, any vector within the space that the basis vectors span can be represented as

$$a = a^1 e_1 + a^2 e_2 + a^3 e_3 + \ldots + a^n e_n = a^i e_i \tag{2.1}$$

where n is the dimension of the space. This notation is called vector expansion over the basis e.

FIGURE 2.1: The basis vectors of the tangent space at the point p.



2.1.3 Basis Transformations

Every vector has a unique vector expansion on any basis. Let say we have three basis vectors e_1, e_2 and e_3 in $\mathbb{R}^3$. These three basis vectors define all the three dimensional vectors in the space $\mathbb{R}^3$ in the form of $a^i e_i$.

In order to have simple coefficients for your vectors in your vector space, it is needed to change the basis. Changing the basis is the same as expanding a vector on a basis.

Let's define new basis vectors as $\hat{e}_1, \hat{e}_2$ and $\hat{e}_3$. The old basis vectors can be defined on the new space that is constructed by the new basis vectors. Let's take one of the old basis vectors e_1.

$$e_1 = s_1^1 \hat{e}_1 + s_1^2 \hat{e}_2 + s_1^3 \hat{e}_3$$

The second and third vector can be expanded as well;

$$e_2 = s_2^1 \hat{e}_1 + s_2^2 \hat{e}_2 + s_2^3 \hat{e}_3$$

$$e_3 = s_3^1\hat{e}_1 + s_3^2\hat{e}_2 + s_3^3\hat{e}_3$$

When considered jointly, these three formulas called transition formulas. They can be grouped and called as *transition matrix* or *direct transition matrix* [11];

$$S = \begin{Vmatrix} s_1^1 & s_1^2 & s_1^3 \\ s_2^1 & s_2^2 & s_2^3 \\ s_3^1 & s_3^2 & s_3^3 \end{Vmatrix}$$

We can also define a transition from the new basis to the old one.

$$\hat{e}_1 = t_1^1 e_1 + t_1^2 e_2 + t_1^3 e_3$$

$$\hat{e}_2 = t_2^1 e_1 + t_2^2 e_2 + t_2^3 e_3 \tag{2.2}$$

$$\hat{e}_3 = t_3^1 e_1 + t_3^2 e_2 + t_3^3 e_3 \tag{2.3}$$

This time the matrix is called *inverse transition matrix* [11].

$$T = \begin{Vmatrix} t_1^1 & t_1^2 & t_1^3 \\ t_2^1 & t_2^2 & t_2^3 \\ t_3^1 & t_3^2 & t_3^3 \end{Vmatrix}$$

Theorem 2.5. *The inverse transition matrix T is the inverse of the direct transition matrix S.*

2.1.4 Vectors - Covectors or Contravariant - Covariant Vectors

A vector does not change when the basis of the vector changed but their coordinates change according to the change of the basis [11].

Suppose we have a vector a expanded on the basis set e_i and let's try to change the basis.

$$a = a^1 e_1 + a^2 e_2 + a^3 e_3 + \ldots + a^n e_n = a^i e_i \tag{2.4}$$

Basis is changed according to our previous formula **??**, written again, this time stating the Einstein summation indices also.

$$e_i = T_i^j \hat{e}_j \tag{2.5}$$

Substituting 2.5 into 2.4 yields:

$$a^i e_i = a^i (T_i^j \hat{e}_j) = (a^i T_i^j)\hat{e}_j = \hat{a}^i \hat{e}_j$$

Hence, *the direct vector transition formula* is as below[11]:

$$\hat{a}^i = a^i T_i^j$$

As it can be seen easily, *the inverse vector transition formula* is:

$$a^i = \hat{a}^i S_i^j$$

Mathematically, we can construct a vectorial object in two ways: one that transforms as (vectors) and one that transforms oppositely as (covectors) aforementioned transformations.

For a vector to be coordinate system invariant, the coordinates of the vector must contravary under a change of basis. That is, the coordinates must vary in the opposite way (with the inverse transformation) as the change of basis. For this being so, they are also called contravariant vectors. Note that, In Einstein's notation, contravariant components are stated as upper indices.

Definition 2.6. A geometric object a in each basis by a set of coordinates $a_1, a_2, \ldots, a_n$ and such that its coordinates obey the below transformation rules under a change of basis is called a *vector (contravariant vector)*[11]:

$$\hat{a}^i = a^i T_i^j$$

and

$$a^i = \hat{a}^i S_i^j$$

For a covector, (such as a gradient) to be coordinate system invariant, the coordinates of the vector must covary under a change of basis to maintain. That is, the coordinates must vary by the same transformation as the change of basis. For this being so, they are also called covariant vectors. In Einstein's notation, covariant components are stated as lower indices.

Definition 2.7. A geometric object a in each basis by a set of coordinates $a_1, a_2, \ldots, a_n$ and such that its coordinates obey the below transformation rules under a change of basis is called a *covector (covariant vector)*[11]:

$$\hat{a}^i = a^i S_i^j$$

and

$$a^i = \hat{a}^i T_i^j$$

2.1.5 Tensors and Their Properties

Before giving the general definition of tensors, it is important to give the definition of the linear operators for understanding the concept.

Definition 2.8. A geometric object F in each basis represented by a square matrix F_j^i and such that components of its matrix obeys the below transformation rules under a change of basis is called a *linear operator*[11]:

$$\hat{F}_j^i = T_p^i \cdot S_j^q \cdot F_q^p$$

$$F_j^i = S_p^i \cdot T_j^q \cdot \hat{F}_q^p$$

As stated in the definition, there is one covariant index and for that being so, there is one inverse transition matrix in the transformation law and the same applies to the contravariant index.

Generalizing that idea will lead through the tensor definition.

Definition 2.9. A geometric object X in each basis represented by a $(r+s)$ dimensional array $X^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s}$ and such that components of its multidimensional array obeys the below transformation rules under a change of basis is called a *tensor* of rank (r,s)[11]:

$$X^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} = S^{i_1,i_2,\ldots,i_r}_{h_1,h_2,\ldots,h_r} T^{k_1,k_2,\ldots,k_s}_{j_1,j_2,\ldots,j_s} \hat{X}^{h_1,h_2,\ldots,h_r}_{k_1,k_2,\ldots,k_s}$$

$$\hat{X}^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} = T^{i_1,i_2,\ldots,i_r}_{h_1,h_2,\ldots,h_r} S^{k_1,k_2,\ldots,k_s}_{j_1,j_2,\ldots,j_s} X^{h_1,h_2,\ldots,h_r}_{k_1,k_2,\ldots,k_s}$$

2.1.5.1 Tensor Addition and Multiplication by a Scalar

Tensor addition and multiplication by a scalar are the most primitive operations. The addition formula is as below:

$$Z^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} = X^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} + Y^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s}$$

As it can be seen from the formula that tensors must be of the same rank in order to perform an addition. The tensor multiplication by a scalar is given by the formula:

$$X^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} = \alpha Y^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s}$$

Scalar multiplication doesn't change the rank of the tensor.

2.1.5.2 Tensor Product

Tensor product is given by the formula:

$$Z^{i_1,i_2,\ldots,i_r+p}_{j_1,j_2,\ldots,j_s+q} = X^{i_1,i_2,\ldots,i_r}_{j_1,j_2,\ldots,j_s} \otimes Y^{i_r+1,i_r+2,\ldots,i_r+p}_{j_s+1,j_s+2,\ldots,j_s+q}$$

This formula is denoted by the symbol $\otimes$. As can be seen from the formula, it takes two tensors with rank respectively $(r, s), (p, q)$ and generates a new tensor with rank $(r + p, q + s)$. This operation increases the rank of the tensors. [11]

2.1.5.3 Contraction

This operation reduces the rank of a tensor of rank (r, s) to $(r-1, s-1)$. *Contraction* is performed by summing over one contravariant and one covariant index. So the formula is:

$$Z^{i_1,i_2,\ldots,i_r-1}_{j_1,j_2,\ldots,j_s-1} = X^{i_1,i_2,\ldots,k,\ldots,i_r}_{j_1,j_2,\ldots,k,\ldots,j_s}$$

Replacing an upper and a lower index with the summation index k let us sum all free indices and reduce the summation index.

2.1.5.4 Raising and Lowering Indices

Raising and lowering indices includes two operations: tensor product and contraction. Before explaining these two concepts, it is important to understand what the metric tensor is.

The metric g_{pq} is the tensor that defines the inner geometry of the space. The metric is used when calculation of the shortest path between two vectors or points needed and also it allows the computation of the shortest path between two points in a certain geometry. This concept will be considered in detail in the Section 2.2.5.

The raising procedure is as below, the first tensor product by the metric is taken and then the second index and the index to be raised is contracted. For that operation being so, the covariant indices are increased by and the contravariant indices decreased by one.

$$Y^{\ldots,p,q,\ldots}_{\ldots,k,\ldots} = g^{pq} \otimes X^{\ldots}_{\ldots,k,\ldots}$$

$$X^{\ldots,p,\ldots}_{\ldots} = g^{pk} Y^{\ldots}_{\ldots,k,\ldots}$$

The inverse operation is called *the lowering procedure* and it is using the inverse metric.

$$Y^{\ldots,k,\ldots}_{\ldots,p,q,\ldots} = g_{pq} \otimes X^{\ldots,k,\ldots}_{\ldots}$$

$$X^{\ldots}_{\ldots,p,\ldots} = g_{pk} Y^{\ldots,k,\ldots}_{\ldots}$$

More information about tensors and tensor operations can be found at [11], and in the first two chapters of [12]. Tensor's properties and their differentiation will be given after the definition of manifold and the smoothness of manifolds are understood. The following section constructs the definition of manifolds.

2.2 Manifolds

2.2.1 Maps and Continuity

To construct the definition of the manifold and its properties of being smooth and locally euclidean, some preliminary definitions are required. One of the most basic definitions is the definition of map notion.

Definition 2.10. Given two sets M and N, a map ϕ:M $\rightarrow$ N is a relationship that assigns each element of M to exactly one element of N.

The composition of given two maps ϕ, ψ is defined below:

Definition 2.11. Given two maps ϕ:M $\rightarrow$ N , ψ: N $\rightarrow$ K, the composition ($\psi \circ \phi$): M $\rightarrow$ K is defined by the operation ($\psi \circ \phi$)(a) = (ψ (ϕ))(a).

A map ϕ is called one-to-one or injective if each element of N has at most one element of M mapped into it and a map is called onto or surjective if each element of N has at least one element of M mapped into it.

In the case of the map ϕ the set M is called **domain** and the set N is called **image**.

The notion of *continuity* of a map given here is the notion of continuity in ordinary functions which are maps defined from $\mathbb{R}$ to $\mathbb{R}$. One can extend the idea to the higher dimensional euclidean spaces, $\mathbb{R}^m$.

Definition 2.12. A map ϕ in $\mathbb{R}$ is continuous at x = a if and only if;

1. ϕ(a) is defined.
2. $\lim_{x\to a}\phi(x)$ exists.
3. $\lim_{x\to a}\phi(x) = \phi(a)$

The left hand derivative of ϕ is given by $\lim_{h\to 0^-}\frac{\phi(a+h)-\phi(a)}{h}$ provided that this limit exists and the right hand derivative $\lim_{h\to 0^+}\frac{\phi(a+h)-\phi(a)}{h}$, again, provided that this limit exists. We say that a map ϕ is *differentiable* at x = a if the left hand derivative equals the right hand derivative. Any calculus book can be checked in order to understand this notions therefore no references will be provided for this notions.

To extend these notions towards more general euclidean spaces, linear map notion must be given.

Definition 2.13. A linear map $\phi : \mathbb{R}^m \to \mathbb{R}^n$ takes a point $(x^1, x^2, \ldots, x^m)$ in $\mathbb{R}^m$ to a point $(y^1, y^2, \ldots, y^n)$ in $\mathbb{R}^n$ while preserving the operations of addition and scalar multiplication. The map $\phi : \mathbb{R}^m \to \mathbb{R}^n$ can be thought as collection of following maps [12]:

$$y^1 = \phi^1(x^1, x^2, \ldots, x^m)$$

$$y^2 = \phi^1(x^1, x^2, \ldots, x^m)$$

$$\vdots$$

$$y^n = \phi^1(x^1, x^2, \ldots, x^m)$$

If p^{th} derivative of a map exists and is continuous, that map is called C^p. A linear map is called C^p if all of its component's p^{th} derivative exists and is continuous. A C^0 map is continuous but not differentiable and a C^∞ map is continuous and can be differentiated infinitely. C^∞ maps are called *smooth* [12].

With the definition of smoothness, we can now define *diffeomorphisms*.

Definition 2.14. Two sets M and N are called diffeomorphic if there exists a C^∞ map $\phi : M \to N$ with an inverse $\phi^{-1} : N \to M$ which is also C^∞. Here, the map ϕ is called *diffeomorphism* [12].

The notion of diffeomorphisms is useful when considering the equivalence of manifolds.

2.2.2 Coordinate charts and manifold definition

Definition 2.15. An open ball is a set of all points x in $\mathbb{R}^n$ such that $|x - y| < r$ for some fixed $y \in \mathbb{R}^n$ and $r \in \mathbb{R}$, where $|x - y|$ is euclidean distance.

In other words, an open ball is the interior of an n-sphere with a radius r centered at y. This definition directly inherits the meaning of a metric space. Here, the metric is the euclidean distance.

Definition 2.16. A set V is called an open set if for any $y \in V$, there is an open ball centered at y such that $y \in$ V.

An open set can be thought as an interior of some $(n-1)$ dimensional closed surface [12]. Along with a map onto an open set in $\mathbb{R}^n$ leads to a definition of charts.

Definition 2.17. A *chart* or *coordinate system* is a one-to-one map

$$\phi : U \to V \tag{2.6}$$

where U is a subset of M and V is an open set in R^n.

Since any map is onto its image, U is an also open set in M. Finally, with these ingredients in hand, manifold definition can be given.

Definition 2.18. An atlas for a set M is an indexed collection (U_α, ϕ_α) of charts on M such that $\bigcup U_\alpha = M$. If the images of charts are n-dimensional Euclidean spaces, then M is said to be an n-dimensional manifold[12].

The manifold definition comprises two important properties. The first one is being locally euclidean. The images of charts are euclidean spaces and since all the charts are consisting of an open set and a map, the chart resembles the euclidean space of the same dimension. This property is called being locally euclidean.

The other important property among charts is being smoothly sewn together. The meaning of this property is smooth maps can be defined between the intersectioned parts between the euclidean spaces that the local parts of the manifold resembles.

2.2.3 Directional Derivatives and Tangent Spaces

A tangent space at point p can be imagined as the collection of vectors that is tangent to all the curves passing p. A derivative definition of manifolds on curves should be given next in order to define the concept of "being tangent on manifolds".

Definition 2.19. Let F be the space of all curves through a point p on a manifold. For each differentiable curve f in F, there is an operator called *directional derivative* such that:

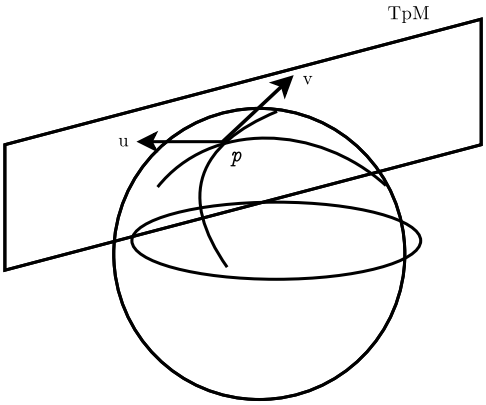
$$f \to df/d\lambda$$

where λ is the parameter along the curve.

Being differentiable for a curve on a manifold means that the curve is differentiable at every chart of the manifold. With the definition of a derivative on manifolds, we can claim that a tangent space is the space of directional derivative operators along the curves through p [12]. The tangent space definition is as the following:

Definition 2.20. Tangent space is a real vector space $\mathbb{R}^n$ tangentially attached to a point p of a differentiable n-manifold M, denoted by T_pM. If γ is a curve passing through p then the derivative of γ at p is a vector in T_pM.

FIGURE 2.2: Tangent space of a manifold M



2.2.4 Riemannian Manifolds and The Metric Tensor

At every point of a manifold, there is a tangent space that defines the tangent vectors of that point. The tangent space at a point p has the same dimensionality as the manifold. There are two properties for a manifold to be *Riemannian*: it should have an inner product defined in every tangent space of the manifold such that one can compute the norm of a vector and the distance between two vectors from that space. The other property is that the inner product should vary smoothly and inner product of two tangent spaces should specify a smooth function on M. This inner product property is allowed by *the metric tensor*.

Since the basis vectors of the tangent space can be constructed using the partial derivatives of the manifold at a point p, the metric can also be different at every point on the manifold and the metric should vary smoothly from point to point on the manifold as the coordinate system changes. That means precisely, given any open subset U on manifold M, at each point p in U, the metric tensor assigns a metric $g_{\mu,\vartheta}$ and this assignment is a smooth mapping on M. Furthermore, it can be seen as a bilinear operator on vectors V^{μ}, U^{ϑ} and also denoted as $g_p(V^{\mu}, U^{\vartheta})$.

The properties of the metric are provided as follows:

- The metric is symmetric. Where U and V are vectors in a tangent space.

$$g_{\mu\vartheta}V^{\mu}U^{\vartheta} = g_{\vartheta\mu}U^{\vartheta}V^{\mu}$$

- The metric is bilinear. Where a, b are scalars,

$$g_{\mu\vartheta}(aV^{\mu} + bU^{\vartheta})W^{\alpha} = a \cdot g_{\mu\vartheta}V^{\mu}W^{\alpha} + b \cdot g_{\mu\vartheta}U^{\vartheta}W^{\alpha}$$

$$g_{\mu\vartheta}W^{\alpha}(aV^{\mu} + bU^{\vartheta}) = ag_{\mu\vartheta}W^{\alpha}V^{\mu} + bg_{\alpha\vartheta}W^{\alpha}U^{\vartheta}$$

- The metric is non-degenarate. That means the determinant of the metric does not vanish, therefore we can calculate the inverse metric by the formula:

$$g_{\mu\vartheta}g^{\vartheta\sigma} = g^{\lambda\sigma}g_{\lambda\mu} = \delta^{\mu}_{\sigma} = \delta^{\sigma}_{\mu}$$

Further reference can be found at [12], [13].

2.2.5 Length of Curves on a manifold and Geodesics

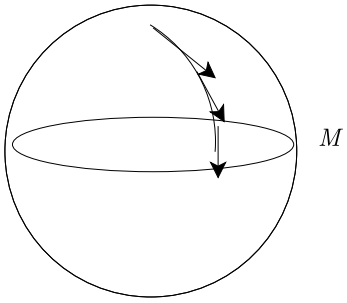
Assume that there exists a curve $\gamma(t) : [0, 1] \to M$. On each point p on the curve γ, there exists a tangent vector $\frac{d\gamma(t)}{dt}$. Since we have the metric in each tangent space, we can calculate each tangent vectors norm. Moving around the curve by infinitesimal steps and summing up this vectors as in figure 2.3 gives us the length of the curve. We can denote the length of the curve γ as $L(\gamma)$.

$$L(\gamma) = \int_0^1 ||\frac{d\gamma(t)}{dt}||dt \tag{2.7}$$

Although the geometry is curved, the notion of the straight line remains. The generalization of straight line is called *geodesics*. A Riemannian manifold is geodesicaly complete. This means that for every point a, b on manifold μ, there exists a geodesic joining them. This theorem is called *Hopf - Rinow theorem*. The details on this theorem can be found at [14].

Geodesic distances are shortest paths between two points on a manifold. To give the mathematical definition of the geodesic, *covariant derivative* should be defined first.

FIGURE 2.3: Tangent vectors of a curve on a manifold M.



2.2.6 Affine Connection, Covariant Derivative and Geodesics

Covariant derivatives are important in this study since the definition of geodesic depends on this notion. Given a parametric curve $\gamma(t)$ on M, as $\gamma(t)$ moves on M, the tangent space $T_{\gamma(t)}M$ changes. This change can be defined with the notion of *covariant derivatives* [15].

Definition 2.21. Let (M, g) be a Riemannian Manifold M equipped with a smooth metric g and let V be the set of all vector fields in M and let $f : M \to R$ is any smooth function.

A connection on M is an operator $\nabla : V \times V \to V$ that satisfies the following conditions:

- $\nabla_{X_1+X_2}Y = \nabla_{X_1}Y + \nabla_{X_2}Y$
- $\nabla_X Y_1 + Y_2 = \nabla_X Y_1 + \nabla_X Y_2$
- $\nabla_{fX}Y = f\nabla_X Y$
- $\nabla_X fY = X(f)Y + f\nabla_X Y$

In addition to those properties, if a connection satisfied the properties below, it becomes *connection with respect to the metric* g.

- $X(g(Y, Z)) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z)$ for any $X, Y, Z \in V$
- $\nabla_X Y - \nabla_Y X = [X, Y]$, the lie bracket of X, Y

Theorem 2.22 (The Fundamental Theorem of Riemannian Geometry)**.** *For any smooth manifold M with a smooth Riemannian metric g there exists a unique Riemannian connection on M corresponding to g. This connection is named* Levi - Civita Connection.

For the proof of this theorem, see [14].

The unique connection given above can be constructed from the metric, and it is encapsulated in an object called the *Christoffel Symbol*, given by

$$\Gamma^{\lambda}_{\mu\upsilon} = \frac{1}{2} g^{\lambda\sigma}(\partial_{\mu} g_{\upsilon\sigma} + \partial_{\upsilon} g_{\sigma\mu} + \partial_{\sigma} g_{\mu\upsilon})$$

The use of this symbol is fundamentally for taking covariant derivatives ∇_{μ}. The covariant derivative of a vector field V^{υ} is given by [12]:

$$\nabla_{\mu} V^{\upsilon} = \partial_{\mu} V^{\upsilon} + \Gamma^{\upsilon}_{\mu\sigma} V^{\sigma}$$

This notion is the generalization of the partial derivatives on manifolds. The formula can be interpreted as the partial derivative plus a correction specified by a set of n matrices $\Gamma^{\rho}_{\mu\sigma}$. The covariant derivative of a tensor of rank (k, l) is given by the formula [12]:

$$\begin{aligned}
\nabla_{\sigma} T^{\mu_1\mu_2\ldots\mu_k}_{\upsilon_1\upsilon_2\ldots\upsilon_l} &= \partial_{\sigma} T^{\mu_1\mu_2\ldots\mu_k}_{\upsilon_1\upsilon_2\ldots\upsilon_l} \\
&\quad + \Gamma^{\mu_1}_{\sigma\lambda} T^{\lambda\mu_2\ldots\mu_k}_{\upsilon_1\upsilon_2\ldots\upsilon_l} + \Gamma^{\mu_2}_{\sigma\lambda} T^{\mu_1\lambda\ldots\mu_k}_{\upsilon_1\upsilon_2\ldots\upsilon_l} + \ldots \qquad (2.8) \\
&\quad - \Gamma^{\lambda}_{\sigma\upsilon_1} T^{\mu_1\mu_2\ldots\mu_k}_{\lambda\upsilon_2\ldots\upsilon_l} - \Gamma^{\lambda}_{\sigma\upsilon_2} T^{\mu_1\mu_2\ldots\mu_k}_{\upsilon_1\lambda\ldots\upsilon_l} - \ldots \qquad (2.9)
\end{aligned}$$

The concept of *parallel transport* is moving a vector or tensor along a path while keeping it constant. In the flat space, there is no need to consider the point that the vector or tensor to be moved on. However, In a curved space, the result of parallel transport depends on the underlying path between points that the vector or tensor to be moved.

For a tensor to be constant on a given curve $\gamma(\lambda)$ is given by the formula:

$$\frac{D}{d\lambda}T^{\mu_1\mu_2\ldots\mu_k}_{\upsilon_1\upsilon_2\ldots\upsilon_l} = \frac{d\gamma^\sigma}{d\lambda}\nabla_\sigma T^{\mu_1\mu_2\ldots\mu_k}_{\upsilon_1\upsilon_2\ldots\upsilon_l} = 0$$

Specifying this formula for vectors yields [12]:

$$\frac{d}{d\lambda}V^\mu + \Gamma^\mu_{\sigma\rho}\frac{d\gamma^\sigma}{d\lambda}V^\rho = 0$$

As stated in the previous section, *geodesics* are the generalized notion of straight line in the curved space. A straight line is the path of the shortest distance between two points. Also, a straight line can be seen as a path that parallel transports its own tangent vector [12].

The tangent vector to a path $\gamma(\lambda)$ is:

$$\frac{dx^\mu}{d\lambda}.$$

The condition that it is parallel transported is as below and this equation is called *geodesic equation* [12]:

$$\frac{d^2\gamma^\mu}{d\lambda^2} + \Gamma^\mu_{\rho\sigma}\frac{d\gamma^\rho}{d\lambda}\frac{d\gamma^\sigma}{d\lambda} = 0$$

2.2.7 Gradient and Exponential Map

The gradient of a scalar function on M is the vector directed at the greatest rate of change and has magnitude of the greatest rate of change at the point p.

$$\mathrm{grad}(f_p) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$$

Gradients can also be applied to tensor fields. Applying gradient to a tensor field with rank (k, l) yields a tensor with rank $(k, l + 1)$

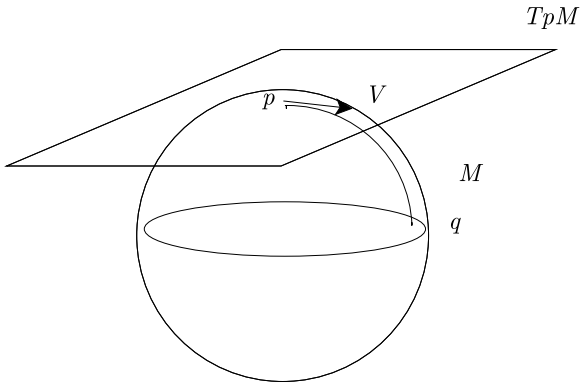
$$Y^{i_1,i_2,\ldots,i_k}_{q,j_1,j_2,\ldots,j_l} = \mathrm{grad}_q(X^{i_1,i_2,\ldots,i_k}_{j_1,j_2,\ldots,j_l})$$

Another definition should be given in order to define *Laplace - Beltrami Operator* which is the main object of study in this thesis. With the use of the definition of geodesics we can define the exponential map of a vector in a tangent space of a manifold.

Definition 2.23. The *exponential map* Exp_p at a point p in M maps the tangent space T_pM into M by sending a vector v in T_pM to the point in M a distance $|v|$ along the geodesic from p in the direction of v [16].

The exponential map takes a vector from the tangent space and map it onto another point on the manifold using the geodesic along the direction fo the vector. Figure 2.4 depicts the map from the tangent space at p onto the point q.

FIGURE 2.4: Exponential map Exp_p of a vector v at point p



2.2.8 Laplace-Beltrami Operator

The Laplace Operator, named after Pierre Simon Laplace and Eugene Beltrami, is the operator on surfaces that maps functions to functions. It can be defined as exponential map of the gradient of a scalar function defined on some manifold M.

In euclidean spaces, this operator can geometrically be interpreted as the map from a point p to another point q so that from the point p, the direction of the greatest rate of change with a magnitude of the greatest rate of change is the point q.

Exponential maps are defined on tangent spaces. From the scalar function f at point p the tangent vector is defined naturally by the *grad* operator. After obtaining this tangent vector, we can apply exponential map and move along with the geodesic in the direction of this tangent vector.

Definition 2.24. The Laplace-Beltrami operator is denoted as $\triangle$ and defined in euclidean spaces as

$$\triangle_M f(p) = \sum_i \frac{\partial^2 f(exp_p(v))}{\partial x_i^2}$$

and on any manifold as

$$\triangle_M f(p) = \frac{1}{\sqrt{det(g)}} \cdot \sum_j \frac{\partial}{\partial x^j} \left(\sqrt{det(g)} \cdot \sum_i g^{ij} \cdot \frac{\partial f}{\partial x_i^2} \right)$$

where $f : M \to R$ is a scalar function, g^{ij} is the metric of the manifold.

2.2.9 Curvature and Sectional Curvature

The curvature of a manifold is defined by the *Riemann Curvature Tensor*. Parallel transportation of a vector defined in a tangent space of the manifold, will linearly transform the vector. The Riemann curvature tensor directly measures the transformation in a general Riemannian manifold. This transportation is known as the *holonomy* of the manifold. [14]

Assume that we have vector v, a and b, a and b are direction vectors and v is the vector that we want to calculate the curvature of. Parallel transport it in the direction of a and then in the direction of b. When the vector v comes back to its original point, there will be a linear transformation reflecting the curvature around a and b of the vector v. For that being so, the curvature tensor should be represented by a tensor of rank $(1, 3)$ [12].

The Riemann Curvature Tensor is given by the formula:

$$R^{\rho}_{\sigma\mu\upsilon} = \partial_{\mu}\Gamma^{\rho}_{\upsilon\sigma} - \partial_{\upsilon}\Gamma^{\rho}_{\mu\sigma} + \Gamma^{\rho}_{\mu\lambda}\Gamma^{\lambda}_{\upsilon\sigma} - \Gamma^{\rho}_{\upsilon\lambda}\Gamma^{\lambda}_{\mu\sigma}$$

The sectional curvature can be defined as the deviation in curving of the geodesic to the euclidean distance between these two points. The sectional of a surface can be defined using the Riemann Curvature Tensor and two vectors. These two vectors are for constructing the surface. Sectional curvature is denoted with K [17].

$$K(S) = K(u_a, v_b) = \frac{R_{\mu\rho\upsilon\sigma} \cdot u^{\mu}_a \cdot v^{\rho}_a \cdot u^{\upsilon}_a \cdot v^{\sigma}_a}{G_{pqrs} \cdot u^{p}_a \cdot v^{q}_a \cdot u^{r}_a \cdot v^{s}_a}$$

where $G_{pqrs} = g_{pr}g_{qs} - g_{ps}g_{qr}$.

2.3 Graphs and Their properties

2.3.1 Graphs

Definition 2.25. A graph G is a finite nonempty set of objects called *vertices* together with a set of unordered pairs of distinct vertices of G called *edges*. The vertex set is denoted by V and the edge set is denoted by E.

The edge $e = u, v$ of a graph is said to join the vertices u and v and they are called *adjacent* if they are joined by an edge.

A *weighted graph* is a graph where each edge has a real number associated to it. A *directed graph* is a graph where each edge has a direction.

Degree of a vertex is the number of vertices that it connects and being *incident* to an edge means that vertex is connected to the edge. Two vertices that is connected by an edge is called *adjacent*[18].

2.3.2 Matrix Structures of Graphs

Another way of representing a graph is *adjacency matrix*. The definition is as follows:

Definition 2.26. Let n be the number of vertices. Adjacency matrix is an $n \times n$ matrix where

$$a_ij = \begin{cases} 1 & \text{if } v_iv_j \in E \\ 0 & \text{if } v_iv_j \notin E \end{cases}$$

Also one can define the Incidence matrix.

Definition 2.27. Let n be the number of vertices and m be the number of edges. Incidence matrix is an $n \times m$ matrix such that:

$$b_ij = \begin{cases} 1 & \text{if } v_ie_j \text{ are incident} \\ 0 & \text{otherwise} \end{cases}$$

Weight matrix is similar to the adjacency matrix but instead of 1's the value of the matrix is decided by the weight of the edges.

Definition 2.28. Let n be the number of vertices. Weight matrix is an $n \times n$ matrix where

$$w_ij = \begin{cases} W(e_{ij}) & \text{if } v_iv_j \in E \\ 0 & \text{if } v_iv_j \notin E \end{cases}$$

Diagonal Weight Matrix of a graph is a matrix whose sums are row-sums of W.

$$D_{ii} = \sigma_j w_{ij} \tag{2.10}$$

Degree matrix is a diagonal matrix where the diagonal represents the degrees of vertices.

$$d_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

2.3.3 Graph Laplacian

Laplacian of a graph is another matrix representation of graphs, mainly used in spectral graph theory.

The Laplacian can be defined as $L = D - W$:

Definition 2.29.

$$L(u,v) = \begin{cases} d_v - w_{uv} & \text{if } u = v \\ -w_{uv} & \text{if } a_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

In this study, the Laplacian carries an important role for the transitions of them, which is explained in detail in the next section.

Chapter 3

Convergence of Graph Laplacian to Laplace-Beltrami Operator

In this chapter, the convergence and relation between Laplacian and Laplace-Beltrami operator is inspected. This intuition will be the key concept in the process of transition of the graphs to the manifolds. The theorems and concepts given in this chapter forms a solid ground to the applications and algorithms implemented in this study. Mentioned theorems and proofs are provided by the studies of Mikhail Belkin and Partha Niyogi [19]. Briefly, in this chapter:

- The Heat Kernel which is a solution for Heat Equation is given in terms of Laplacian.
- The convergence for the uniform distribution is provided.
- The convergence for an arbitrary probability distribution is provided.

3.1 Heat Equation

The Heat Equation is a partial differential equation which describes the distribution of heat in a given region or surface over time.

Definition 3.1. Let $x_1, x_2, \ldots, x_n$ be the spatial variables and t is time variable. The *heat equation* for $\mathbb{R}^n$ is:

$$\frac{\partial u}{\partial t} - (\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \ldots + \frac{\partial^2 u}{\partial x_n^2}) = 0$$

or alternatively:

$$\frac{\partial u}{\partial t} - \triangle u = 0$$

where $\triangle$ is the Laplace-Beltrami operator for $\mathbb{R}^n$ and $u(x_1, x_2, \ldots, x_n, t)$ is the heat function in $\mathbb{R}^n$.

The Laplace-Beltrami operator as can be seen in the definition closely related to the heat flow over a space.

Let $f : M \to \mathbb{R}$ be the initial heat distribution. The value $u(x, t)$ is the heat distribution at the time t. In this case, $u(x, o) = f(x)$. *The heat kernel solution* (H_t) is one of the main solution to the heat equation problem. The solution is given by the formula:

$$u(x, t) = \int_M H_t(x, y) f(y)$$

and in a local coordinate system on a manifold, the solution H_t is approximately the Gaussian [19].

$$H_t(x, y) = (4\pi t)^{\frac{n}{2}} e^{-\frac{|x-y|^2}{4t}} (f(x, y) + O(t)) \tag{3.1}$$

where $f(x, y)$ is a smooth function on manifold with $f(x, x) = 1$ and $O(t)$ is the error value. When x, y are close, i.e. in the same neighbourhood, and t is small, H_t is approximately [2]:

$$H_t(x, y) \approx (4\pi t)^{\frac{n}{2}} e^{-\frac{|x-y|^2}{4t}}.$$

So for euclidean spaces, the heat kernel is typically given by:

$$H^t f = (4\pi t)^{\frac{n}{2}} \int_{R^n} e^{-\frac{|x-y|^2}{4t}} f(y) dy$$

where the limit of $H^t f$ when $t \to 0$ is given by

$$f(x) = \lim_{t \to 0} H^t f(x).$$

We know that this equation satisfies the heat equation $\frac{\partial u}{\partial t} - \triangle u = 0$, leaving the Laplace-Beltrami alone yields:

$$\triangle\, u(x,t) = -\frac{\partial u(x,t)}{\partial t}$$

At $t = 0$

$$\begin{aligned} \triangle f(x) &= -\frac{\partial}{\partial t} u(x,t)\Big|_{t=0} \\ &= -\frac{\partial}{\partial t} H^t f(x)\Big|_{t=0} \\ &= \lim_{t \to 0} \frac{1}{t}(f(x) - H^t f(x)) \end{aligned}$$

The Heat Kernel is Gaussian and integrates to 1

$$= \lim_{t \to 0} -\frac{1}{t}\big((4\pi t)^{-\frac{n}{2}} \int_{R^n} e^{-\frac{|x-y|^2}{4t}} f(y) dy - f(x)(4\pi t)^{-\frac{n}{2}} \int_{R^n} e^{-\frac{|x-y|^2}{4t}} dy\big)$$

The integrals can be approximated using summations over a set of points $(x_1, x_2, \ldots, x_k)$ which are assumed to be sampled on a manifold, then the Laplace-Beltrami operator becomes:

$$\triangle\, f(x) = \frac{1}{t} \frac{(4\pi t)^{\frac{n}{2}}}{k} \big(f(x) \sum_i e^{-\frac{|x_i - x|^2}{4t}} - \sum_i e^{-\frac{|x_i - x|^2}{4t}} f(x_i)\big)$$

If the weights of the graph which is constructed from sample points are chosen to be $w_{ij} = e^{-\frac{|x_i - x_j|^2}{4t}}$, then the above expression simplifies to:

$$\frac{1}{t(4\pi t)^{\frac{n}{2}}} L_n^t f(x)$$

where L is the Graph Laplacian of identical points [19]. These set of equations and convergence construct the mathematical basis for the graph embeddings to manifolds. The heat kernel provides us a smooth approximation of edges between sampled discrete points of manifolds.

3.2 Convergence Theorems

3.2.1 Convergence for Points from a Uniform Distribution

Consider a manifold embedded in $\mathbb{R}^n$. Given data points $S_n = x_1, x_2, \ldots, x_n$ sampled i.i.d. from a uniform distribution. The Laplacian can be constructed from this sample point by taking $x_1, x_2, \ldots, x_n$ as vertices and taking edges by the formula $w_{ij} = e^{-\frac{|x_i - x_j|^2}{4t}}$. The below theorem shows that for a fixed function $f \in C^\infty(M)$ and for a fixed point $p \in M$, after appropriate scaling (according to the heat equation, explained in the previous section) L converges to Laplace-Beltrami Operator ($\triangle$).

Theorem 3.2. *Let data points $x_1, \ldots, x_n$ be sampled from a uniform distribution on a manifold $M \subset R^n$. Put $t_n = n^{-\frac{1}{k+2+\alpha}}$, where $\alpha > 0$ and let $f \in C^\infty(M)$. Then the following equation holds:*

$$\lim_{n \to \infty} \frac{1}{t(4\pi t)^{\frac{n}{2}}} L_n^{t_n} f(x) = \frac{1}{vol(M)} \triangle_M f(x)$$

where the limit is taken in probability and $vol(M)$ is the volume of the manifold with respect to the canonical measure.

The proof of this theorem is in [19].

3.2.2 Convergence for Points from an Arbitrary Probability Distribution

Above theorem for an arbitrary probability distribution P of a set of sampled points can be stated as follows:

Theorem 3.3. *Let $P : M \to R$ be a probability distribution function on M according to which data points $x_1, \ldots, x_n$ are drawn in independent and identically distrubuted fashion. Then for $t_n = n^{-\frac{1}{k+2+\alpha}}$, $\alpha > 0$, we have*

$$\lim_{n\to\infty} \frac{1}{t(4\pi t)^{\frac{n}{2}}} L_n^{t_n} f(x) = \frac{1}{vol(M)} P(x) \triangle_{P^2} f(x)$$

where $\triangle_{P^2}$ is the weighted Laplacian.

In the algorithms in this study, the intuition is always that the graph is a proxy to the manifold. Therefore, to justify this intuition, these theorems are provided in this section. For further reference about Laplacian and Laplace - Beltrami operator, see [19], [20], [21], [22].

Chapter 4

Constructing Graphs from Point Clouds

This chapter aims to describe the methods used to construct graphs from n dimensional data. In this thesis, two methods are used for the construction:

- k-Nearest Neighbours
- $\in$-Neighbourhoods

This chapter contains the analysis of these two methods, their ramifications and advantages in the process. At the end of this chapter, 3D visualizations of the graphs constructed using these methods from random datasets are provided.

4.1 k-Nearest Neighbours Method ($k - nn$)

This method has been studied and widely used in the fields of pattern recognition, statistical classification, computer vision and machine learning. As the name suggests, this method produces a graph in which every point is connected to its k nearest neighbors. The distance function used in this study is Euclidean Distances of the data points.

Algorithm 1 Computation of $k-nn$ Graphs

Input: X: Dataset of n dimensions, k: The parameter of $k-nn$
Output: Undirected graph in which k-nearest neighbours are connected

```
Euc ← [n][n]                                   ▷ Calculate Euclidean Distances
for i ← 1 to n do
    for j ← 1 to n do
        Euc[i][j] ← Distance(X[i], X[j])
    end for
end for

for i ← 1 to n do
    for j ← 1 to k do                          ▷ Find k minimum for each node in X
        minindex = min{Euc[i]}
        Adj[i][minindex] = 1
        Euc[i][minindex] = maxint
    end for
end for
```

The Algorithm is given below:

This is the brute force version of this algorithm and its asymptotic tight bound is $O(kn^2)$. There are many optimizations and parallel implementations that can be applied on this algorithm. When $k = 1$, the nearest neighbor for each data point is connected. This particular case is called the all nearest neighbors problem. The optimization for the $1-nn$ problem can be found in the reference numbered [23]. Furthermore, relaxation based versions of this algorithm can be inspected in order to approximate $k-nn$. The optimizations and parallel implementations are not included in this research. For further reading for optimizations refer to [24], [25], [26].

The $k-nn$ algorithm does not make any geometrical assumptions on the data. The only assumption is that the data lies on a metric space.

4.1.1 Parameter Selection

The parameter of the $k-nn$ guarantees that there will be k edges for each node in the graph. Therefore, wrong choice of the parameter does not lead to significant

geometrical mistakes in this algorithm. The best choice of the parameter generally depends on the data. However, smaller values generate sparse graphs.

4.1.2 Visualization

This chapter includes visualizations of the $k - nn$ algorithm with respect to the different choices of k in the random datasets for 20, 30 and 40 nodes. The generated random numbers are within the open interval of $(0, 1)$. These visualizations intends to give intuitive notion about constructed graphs.

FIGURE 4.1: Graph constructed from 20 nodes and with a parameter $k = 3$.

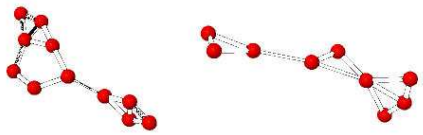
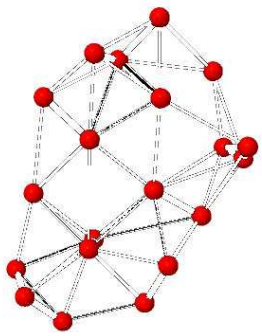


FIGURE 4.2: Graph constructed from 20 nodes and with a parameter $k = 5$.



Even though it is a small possibility to construct separated graphs with this method, as can be seen in the Figure 4.1.2, two discrete graphs are constructed as a result of this algorithm with the parameter choice of $k = 3$.

FIGURE 4.3: Graph constructed from 20 nodes and with a parameter $k = 7$.

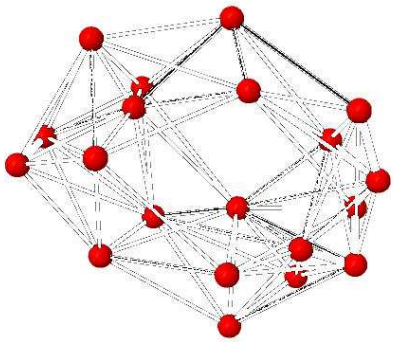
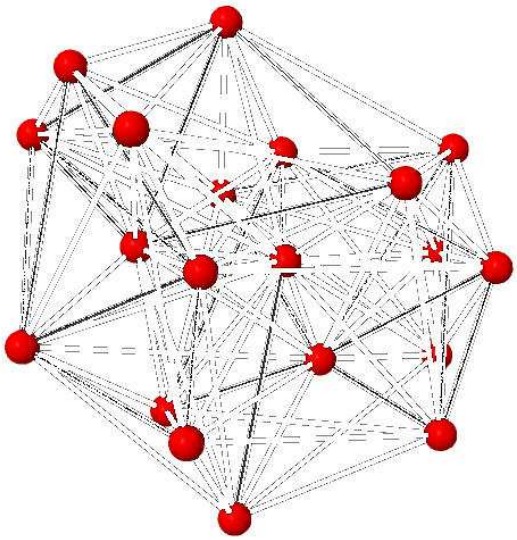


FIGURE 4.4: Graph constructed from 20 nodes and with a parameter $k = 10$.



4.2 $\in$ - neighbourhoods

$\in$-graph is a graph where pairwise nodes are connected if the distance in between is less than a predefined parameter $\in$. The $\in$-graph is more geometrically motivated than the $k - nn$ algorithm since the choice of the parameter is more geometrically dependent on the data set.

The $\in$ - graph algorithm with wrong choice of parameter $\in$ with respect to the data may yield to disconnected graphs [2]. However, if chosen wisely, this algorithm yields to graphs that are geometrically symmetric.

Algorithm 2 Computation of $\in$ - Graphs

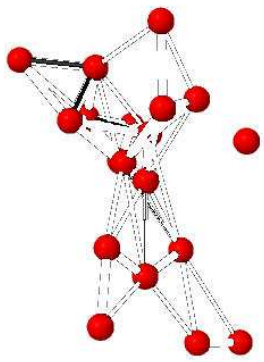
Input: X: Dataset of n dimensions, $\in$: The parameter of $k - nn$
Output: Undirected graph in which pairwise points are connected if the distance in between less than or equals to $\in$.

for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to n **do**
    if Distance(X[i], X[j]) $\leq \in$ **then** ▷ Calculate Euclidean Distances and connect
      Adj[i][j] $= 1$
    end if
  end for
end for

The $\in$-graph method is studied extensively in the literature. For further optimizations and literature points, see [27], [28].

4.2.1 Visualizations

FIGURE 4.5: Graph constructed from 20 nodes and with a parameter $\in = 0.5$.



In the Figure 4.2.1, there is a dangling node which is not connected to any other node in the graph.

FIGURE 4.6: Graph constructed from 20 nodes and with a parameter $\in= 0.6$.

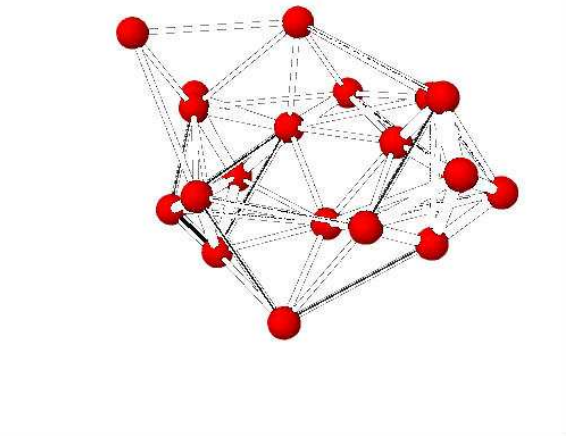
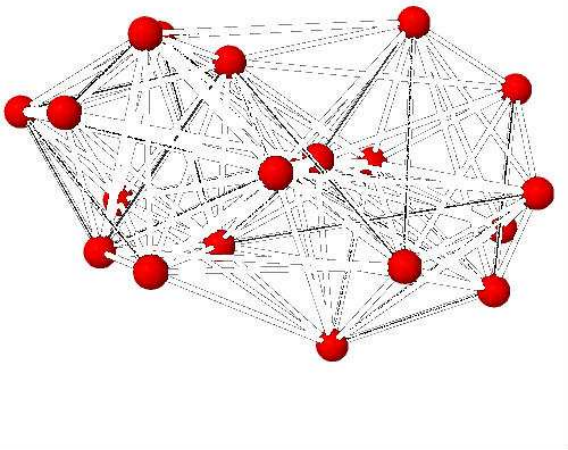


FIGURE 4.7: Graph constructed from 20 nodes and with a parameter $\in= 0.7$.



Chapter 5

Transition to Manifolds

The justification of the relation between graph Laplacian and Laplace-Beltrami operator is given in Chapter 3. The methods of calculating the Laplacian of a graph is given in Chapter 4. This chapter introduces the methods of transitions of graphs onto manifolds. With this aim, there are 4 methods to be described next.

1. ISOMAP (Tenenbaum, de Silva, Langford, 2001)

2. Locally Linear Embeddings (Roweis, Saul, 2001)

3. Laplacian Eigenmaps (Belkin, Niyogi, 2002)

4. Riemannian Approach (Antonio Robles-Kelly, 2007)

Each of these methods are based on different key ideas. *Isomap* tries to implement the shortest path algorithm for calculating the distances and it does not depend on the Laplacian matrix to transit the nodes of the graph. *Laplacian Eigenmaps* method is making use of the heat equation method and the method of *Locally Linear Embedding* method also comprises a relation with the Laplacian [2]. In the *Riemannian Approach* method, the distances between nodes are calculated with the predefined constant curvature and points are mapped according to these distances [4].

The first three algorithm aims to reduce the dimensionality of the data lies on a nonlinear manifold. Yet, the relation of these algorithms and this study is about the graph mappings of these algorithms. These algorithms create mappings from graphs onto manifolds in the process of reducing the dimensionality. Therefore, these algorithms constitutes a framework for the aim of representing data on manifolds.

5.1 Software Development and Technologies Used

The following part of this thesis contains information about the methods of transmission and the visualizations of the aforementioned methods in 3-dimensional space.

In this study, these methods are coded in the programming language of *Python version 2.7*. *Python* language is chosen because of the fast n-dimensional matrix manipulation library *NumPy* and the scientific library of Python *SciPy*. The versions of *NumPy* and *SciPy* are respectively 1.6.1 and 0.9.0.

The integrability of the open source mathematical software *SAGE* is also one of the reasons of choosing the Python language. Graph visualizations of this study is from the graph library of the *SAGE*. The version of *SAGE* used in this thesis is version 4.8. The 3-dimensional manifold visualizations are from the surface interpolation library of *SAGE*. All the manifold visualizations in this study have the aim of providing a geometrical idea of these methods.

5.2 Graph Embedding Methods

5.2.1 ISOMAP

Isomap algorithm, as mentioned in the introduction of this chapter, uses shortest path algorithm to compute the distances between nodes. The main aim of this algorithm to reduce the dimensionality of the data on non-linear manifold. The algorithm tries to find a low dimensional representation covering the geometrical aspects of the data. Isomap tries to combine the major algorithmic features of Principal Component Analysis (PCA) and Multi-Dimensional Scaling (MDS) with the flexibility to learn a broad class of nonlinear manifolds. PCA finds a low-dimensional embedding of the data with respect to the variance of the data set while MDS tries to find a appropriate embedding with respect to the interpoint euclidean distances. PCA and MDS, are simple to implement, efficiently computable, and guaranteed to discover the true structure of data lying on or near a linear subspace of the high-dimensional input space [29].

As explained in the introduction of this chapter, the algorithm creates a graph from the data set and maps it onto manifolds. The interpoint distances are calculated as euclidean distances and the shortest paths netween nodes constitutes the embedding.

The first part of the algorithm is the construction of graphs in one of the two methods explained in Chapter 4. After generating the graph, the graph structure for the embedding is constructed. The initialization is done by defining $d_g(i,j) = d_g(j,i)$ and if node i and node j are linked, $d_g(i,j) = \infty$.

The second phase is to define the shortest paths. For each value of k in the interval of $0, \ldots, N$, where N the number of nodes, replace all entries $d_g(i,j)$ by $min(d_g(i,j), d_g(i,k)+d_g(k,j))$. The matrix of final values will contain the shortest paths in the graph. Those values are regarded as the geodesics of two points on the manifold.

The final phase the algorithm is to compute the embeddings on a manifold. Let λ_p be the p^{th} eigenvalue of the matrix $\tau(d_g)$ where $\tau(D) = -HSH/2$ where S is the

square of the matrix d_g and H is the centering matrix defined as $H_{ij} = \delta_{ij} - 1/N$. Let v_p^i be the i^{th} component of the p^{th} eigenvector. The p^{th} component of the d dimensional data vector y_i is computed as $\sqrt{\lambda_p v_p^i}$.

Algorithm 3 ISOMAP

Input: X: Dataset of n dimensions.

1. Compute the graph using one of the methods in Chapter 4.
2. Compute the shortest path distances between all the nodes in graph.
3. Returned data points y_i on manifold computed as $\sqrt{\lambda_p v_p^i}$.

The Isomap Method may not be stable according to the geometry of the underlying data since the curvature and the metric of the manifold is not regarded in this method. However, this method is very efficient. For that reason, this algorithm is mentioned in this thesis as one of the methods that provides an isometric transmission of graphs onto manifolds. Yet, the distances between nodes are calculated as shortest path in the graph and these distances are regarded as geodesic. However, the shortest path distance concept is not equivalent of geodesic definition on a smooth manifolds. Therefore, the link between geodesic and shortest path is weak in this method of transmission.

5.2.2 Laplacian Eigenmaps Method

Laplacian Eigenmaps method considers the construction of geometric representation of data on a low dimensional manifold. The geometrical intuition behind this method is inspired by the convention of heat in the nature. This method constructs a natural link between the Graph Laplacian and the Laplace Beltrami Operator by the heat equation.

In this method, locality of the nodes with respect to their euclidean distances are preserved. Locality property means that the embedding keeps the local points near on the manifold. The neighbourhood information also plays a key role in the construction of the graph from datasets. The graph is constructed by one of the two methods described in Chapter 4, which are $k - nn$ or $\in$-neighbourhood. In either

case the locality is tried to be preserved and the near points are tried to be connected, which ensures the neighbourhood information also to be preserved.

Algorithm 4 explains the method explicitly. The heat kernel weight selec-

Algorithm 4 Laplacian Eigenmaps [2].

1. Constructing the adjacency graph using
   - $k - NN$ or
   - $\in -$ Neighbourhood.
2. After constructing the adjacency graph. The graphs weights should be chosen. Two ways defined in the Laplacian Eigenmaps method. These are:
   - Simple minded weight selection:

$$w_{ij} = \begin{cases} 1 & \text{if node i and j are connected} \\ 0 & \text{otherwise} \end{cases}$$

   - The heat kernel weight selection, which is:

$$w_{ij} = \begin{cases} e^{-\frac{|x_i - x_j|^2}{4t}} & \text{if node i and j are connected} \\ 0 & \text{otherwise} \end{cases}$$

3. Construct the Graph Laplacian and compute the eigenvalues and eigenvectors for the problem of:

$$L \cdot f = \lambda \cdot D \cdot f \tag{5.1}$$

   Let $f_0, f_1, \ldots, f_{k-1}$ be the solutions of the problem 5.1. The solutions are ordered according to their eigenvalues:

$$\begin{aligned} L \cdot f_0 &= \lambda_0 \cdot D \cdot f_0 \\ L \cdot f_1 &= \lambda_1 \cdot D \cdot f_1 \\ &\cdots \\ L \cdot f_{k-1} &= \lambda_{k-1} \cdot D \cdot f_{k-1} \\ 0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{k-1} & \end{aligned}$$

   The embedding is constructed by omitting the f_0 since it is the trivial solution of the problem 5.1. [2]

tion naturally provides us a smooth approximation of edges between the sampled discrete points of the manifolds. Heat kernel, as explained in the Chapter 3, is the smooth convention of heat between two discrete points by a geodesic. Consequently, the intuition of defining geodesic provides the approximation.

This method also related with the spectral clustering problem. Since, the Laplacian and its eigenvalues can be used to describe geometrical properties of graphs, they also bares information about connectedness and clusters of graphs. The justification of this relation is explained in [2].

The method of Laplacian Eigenmaps is also a reduction of the next method Locally Linear embedding (LLE). The problem that the LLE attemps to minimize is an equivalent of finding the eigenfunctions of the Graph Laplacian in return. The detailed justification is also given in [2].

5.2.3 Locally Linear Embedding (LLE)

LLE method is one of the dimensionality reduction methods with a different approach. LLE, instead of estimating pairwise distances, globally reconstructs the embedding using an error function on linear weights. This error function is used to keep local points near in the embeddings. The linear weights are computed as the minimal value of the following error function:

$$\varepsilon(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2 \tag{5.2}$$

The weights of the graph from the sample points are constructed by minimizing these least square problem in (5.2). In this computation, there are two constraints: only the connected points are accounted for the least square problem and sum of all edge weights of each node is always 1. By these two constraints, the constructed graph presents invariant information about the underlying geometry [1].

The method is provided in Algorithm 5 and Figure 5.1 depicts the LLE method.

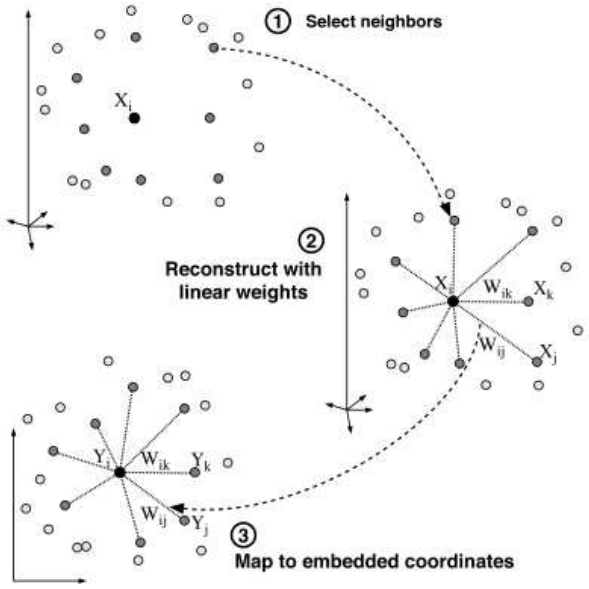
What makes this method different than other methods in this study is that LLE tries to assign each node a weight that fits best among its neighbours with respect to the cost function. The second important point of this method is that the embedding

Algorithm 5 Locally Linear Embedding [3]

1. For each node in the dataset, the edges are defined by either $k - NN$ or $\in -$ neighbourhood.
2. Each edge given a weight in the interval of $[0, 1]$ by minimizing the function $\sum_i |X_i - \sum_j W_{ij} X_j|^2$ such that the sum of all weights of each node is 1.
3. Embedding is computed by taking k lowest eigenvectors of the matrix:

$$E = (I - W)^T (I - W)$$

FIGURE 5.1: Steps of Locally Linear Embedding [1]



is invariant under linear operations such as rotating and scaling. The weights are calculated under the assumption that the sum of linear weights is 1 for each node. This assumption creates an equivalence between linearly modified versions of the data set. Therefore, any kind of linear operations do not change the embedding.

5.2.4 A Riemannian Approach for Graph Embedding

In this method the same relationship between Laplacian and Laplace - Beltrami operator is used. However, the edge weights are chosen as sectional curvatures of a manifold with constant curvature. This method uses the properties of Jacobi fields

to compute an edge-weight matrix in which the elements are connected by curved geodesics on the manifold between nodes [4].

In general, manifolds can have rather complex structures than the constant curved ones. However, the approach of this method is the most geometrically intuitive one. Finding a manifold which encapsulates the underlying geometry of information is the main aim of this method. The embedded manifold assumed to be of constant curvature. The curvature is represented by a parameter K, such that $K \in \mathbb{R}$. By altering this parameter, one can try to approach the geometry of underlying manifold of information. This method is proposed in [4] for the aim of graph matching.

The eigenvalue decomposition of Laplace - Beltrami operator provides many useful information about the underlying geometry such as the sectional curvature, volume or Euler characteristics of the geometry [4]. However, this method tries to find a corresponding manifold given a constant curvature and the Laplacian with the edge weights chosen to be that of the curvature.

The method first computes the edge weights between two nodes by the following formulation.

$$W_{ij} = \begin{cases} \int_0^1 (a(u,v)^2 + \kappa(sin(\sqrt{\kappa}a(u,v)t)^2))dt & \kappa > 0 \\ \int_0^1 a(u,v)^2 dt & \kappa = 0 \\ \int_0^1 (a(u,v)^2 - \kappa(sinh(\sqrt{-\kappa}a(u,v)t)^2))dt & \kappa < 0 \end{cases} \tag{5.3}$$

The formulation (5.3) is the representation of the geodesics on the manifold with the constant curvature κ. The function $a(u,v)$ is the Euclidean distance of the two nodes u and v. When $\kappa = 0$, that means the space is flat. On that ground, the edge weights are equal to the weights of an Euclidean space. If $\kappa \neq 0$ then the corrections which reflects the diversion from euclidean space is included in the formulation. This corrections are calculated as the Jacobian Field of a geodesic from a manifold of constant curvature [4].

The algorithm is summarized below:

Algorithm 6 A Riemannian Approach for Graph Embedding [4].

1. For each node in the dataset, the edges are defined by either $k-NN$ or $\in-$ neighbourhood.
2. Each edge given a weight by the function:

$$W_{ij} = \begin{cases} \int_0^1 (a(u,v)^2 + \kappa(sin(\sqrt{\kappa}a(u,v)t)^2))dt & \kappa > 0 \\ \int_0^1 a(u,v)^2 dt & \kappa = 0 \\ \int_0^1 (a(u,v)^2 - \kappa(sinh(\sqrt{-\kappa}a(u,v)t)^2))dt & \kappa < 0 \end{cases} \tag{5.4}$$

3. The embedding is calculated as the eigenvalues of the Graph Laplacian as explained in the third step of the Algorithm 4.

After the calculation of edge weights the procedure is very similar to the Laplacian Eigenmaps method. The embedding is calculated as the eigenvalues of the Graph Laplacian. However, this method constitutes a more geometrical intuition since the sectional curvature between the nodes of the graph is taken into account.

The graph matching applications and discussions of this method can be found in [4]. However this study is only interested in the procedure of transforming nodes of the graphs into the points of the manifold. In the Appendix A the visualizations of these methods in $3d$ is provided.

Chapter 6

Conclusion

This thesis aims to provide a framework for embedding graphs onto manifolds. As stated in the introduction, the author claims that this compilation of manifold methods will be useful when the regarded non-Euclidean information model and computable non-Euclidean properties of information are defined.

The point of origin of this thesis is that the information model should be smooth and nonlinear. To define a new information model, the properties and analogies between discrete and continuous worlds is inspected via this thesis.

In this thesis, the link between one of the main data structures of computation and smooth manifolds is investigated. Several methods are focused on for the purpose of finding the link and the methods in Chapter 5 are implemented to develop a software for this aim.

Graphs, the very common data structure of computation, are mapped onto manifolds in this thesis. Yet, it is important to state that to create a mapping of a graph, one should have the assumption that the data points which will be constituting the graph are sampled from a manifold. The link and the approximation to continuous states then becomes meaningful in this study.

The link between Discrete Laplacian and Continuous Laplace - Beltrami operator is studied in Chapter 3. This link is the main connection between two states.

Methods using this link are presented in Chapter 5. Though, the methods in Chapter 5 are borrowed from the areas of pattern recognition or manifold learning, the perception of these methods in the process of modeling information is novel. This study leads to further directions and they are discussed in the next section.

One hidden outcomes of this thesis is to gain knowledge about the theory of differential geometry. That is also one of the reasons that any application is not included in this study. Grasping the theory to find new theoretical basis for information is the key in this study. This key is important for the aims stated in the future directions.

6.1 Further Directions

One of the future directions emerged from this study is to develop a non - Euclidean information retrieval framework. The current Vector Space Models (VSM) can be expanded by the assumption that the vector space they are in is tangent space of a point on a manifold. This assumption leads to create geodesic distances between data points on manifold and with the geodesic distances being transformations on data points, there may be present optimizations on queries.

The second direction, which also makes this thesis meaningful, is to define the geometry of information with the rather complex structures than manifolds of constant curvature. The timeliness, validity and such properties of information cannot be modeled through static geometries. Geometry of information should evolve with time to model those properties. The current computational models which are all based on the Turing's model do not enclose the role of time in the information. Though the time and space based evolving geometries is within the enclosure of differential geometry, there is no computational models with such properties.

The third and final direction is to find different features of information which one cannot find when the information is in the discrete state. Again to find such features, the theory should be investigated to find similar links to that of Laplacian and Laplace - Beltrami.

Appendix A

Visualizations of the Graph Embedding Methods

In this part of the thesis, the visualizations of the methods in Chapter 5 are included. This visualizations are only surface interpolations for the data points generated by these methods and by no means they are representing the real geometry of manifolds. They are included in this thesis to provide an intuition about the methods explained in Chapter 5.

Datasets generated in this part is random data sets in Euclidean space. They are tried to be embedded into manifolds using the methods in Chapter 5. Sage version 4.8 uses GMP based random number generators for the functions used to create datasets.

A.1 Visualizations of Laplacian Eigenmaps

FIGURE A.1: Graph embedding using Laplacian Eigenmaps with 20 nodes

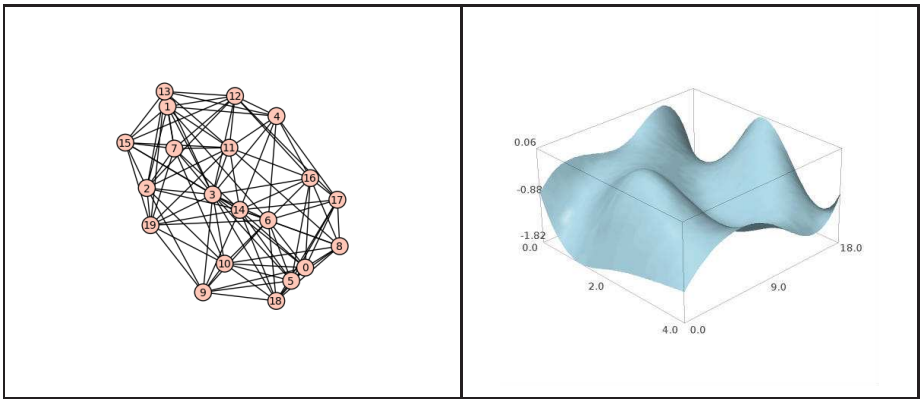


FIGURE A.2: Graph embedding using Laplacian Eigenmaps with 30 nodes

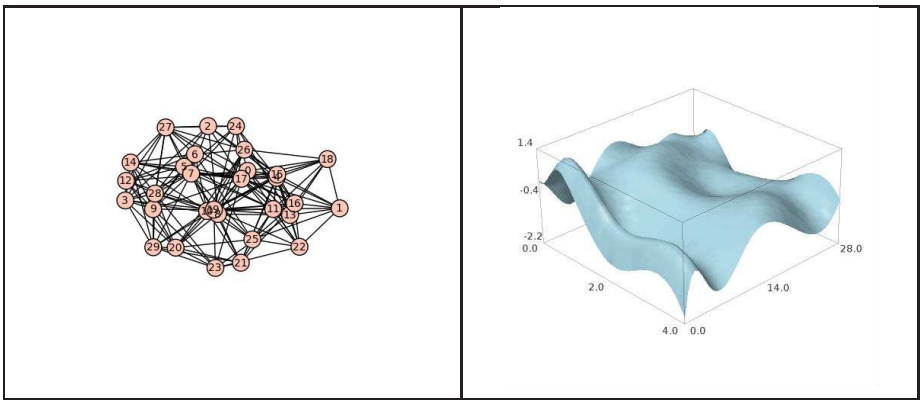
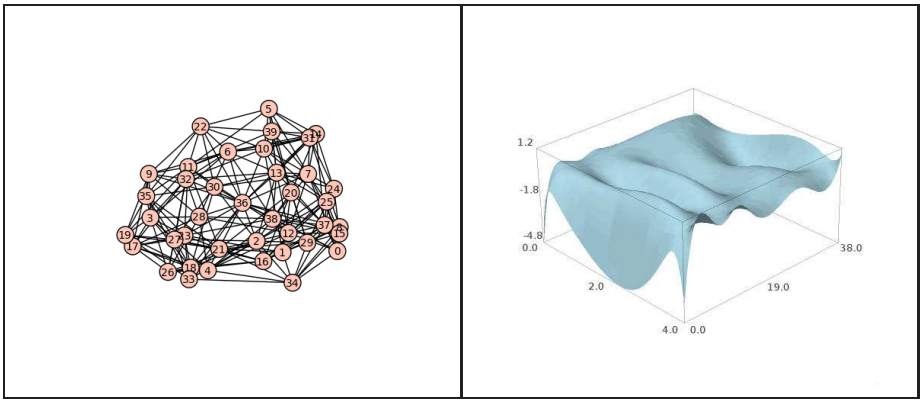


FIGURE A.3: Graph embedding using Laplacian Eigenmaps with 40 nodes



A.2 Visualizations of Locally Linear Embedding

FIGURE A.4: Graph embedding using LLE with 20 nodes

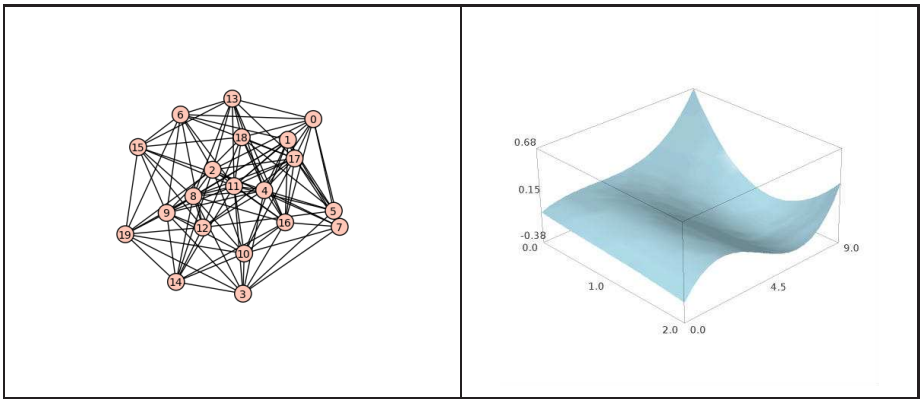


FIGURE A.5: Graph embedding using LLE with 30 nodes

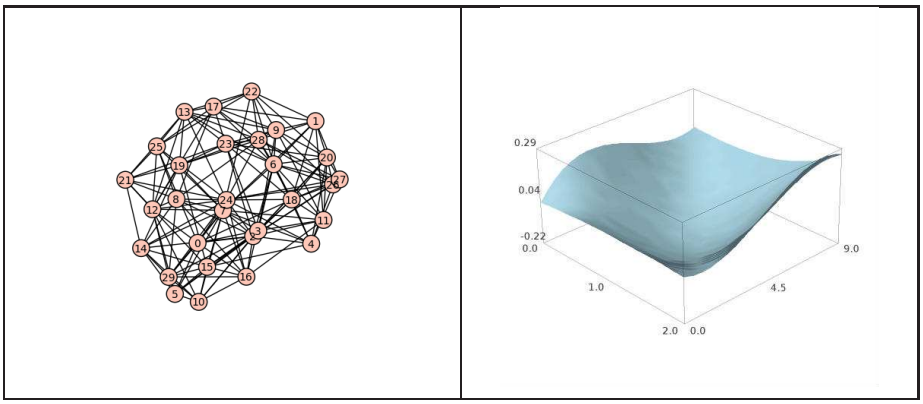
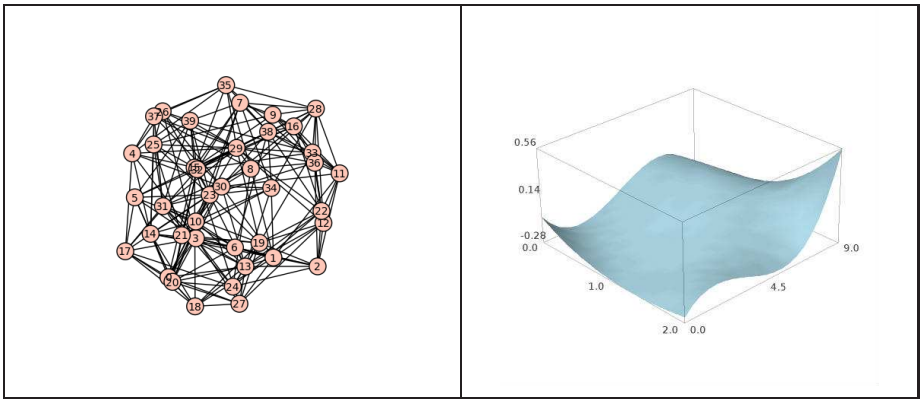


FIGURE A.6: Graph embedding using LLE with 40 nodes



A.3 Visualizations of Riemannian Approach

FIGURE A.7: Graph embedding using the Riemannian Approach with 20 nodes

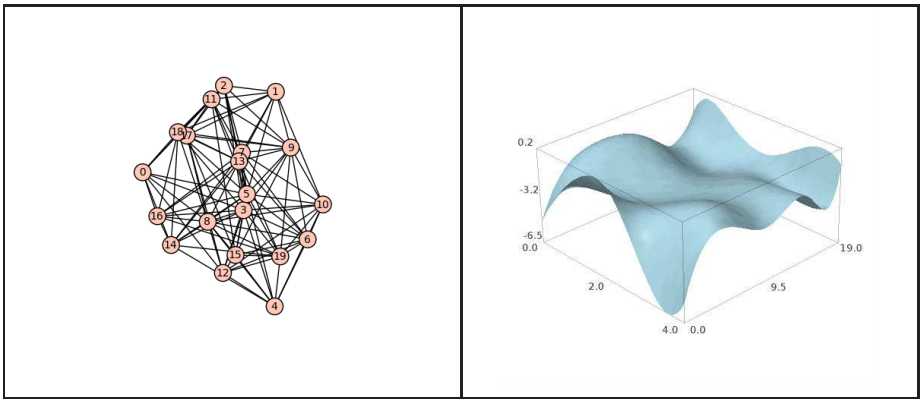


FIGURE A.8: Graph embedding using the Riemannian Approach with 30 nodes

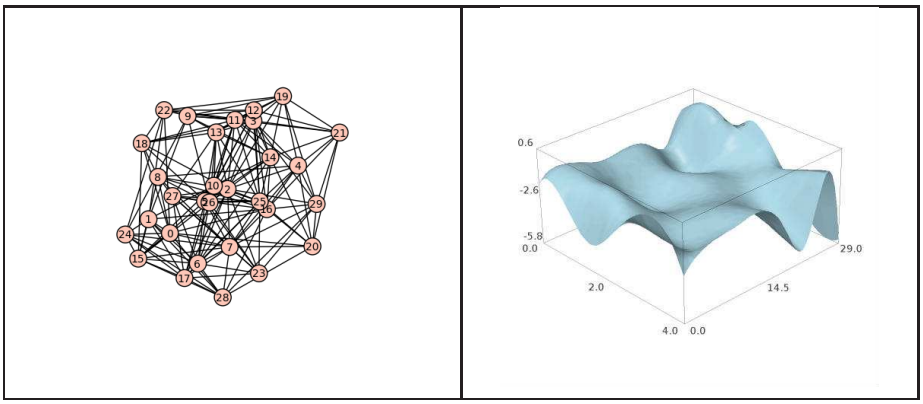
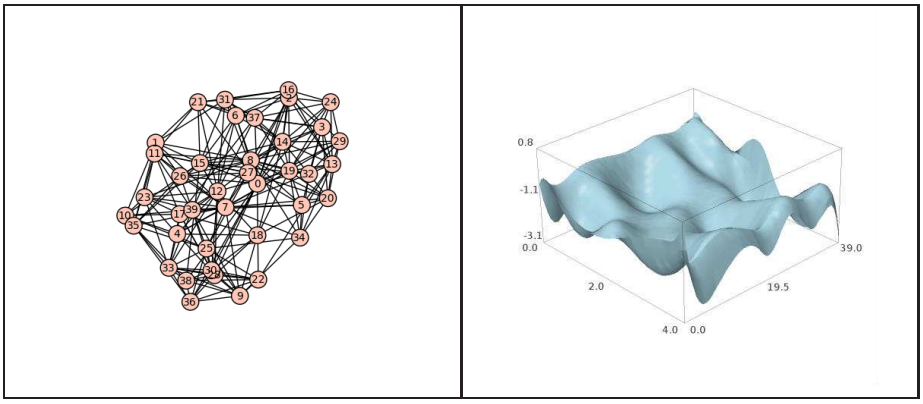


FIGURE A.9: Graph embedding using the Riemannian Approach with 40 nodes



Bibliography

[1] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003. ISSN 0899-7667. URL http://dx.doi.org/10.1162/089976603321780317.

[3] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[4] Antonio Robles-Kelly and Edwin R. Hancock. A riemannian approach to graph embedding. *Pattern Recognition*, 40(3):1042–1056, 2007.

[5] Claude E. Shannon. A mathematical theory of communication. *Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[6] David Vallet, Miriam Fernández, and Pablo Castells. The semantic web: Research and applications. In Jérôme Euzenat Asunción Gómez-Pérez, editor, *The Semantic Web: Research and Applications*, chapter An Ontology-Based Information Retrieval Model, pages 455–470. Springer Berlin / Heidelberg, 2005. URL http://dx.doi.org/10.1007/11431053_31.

[7] Sunny K. M. Wong, Wojciech P Ziarko, Vijay V. Raghavan, and Patrick CN Wong. On modeling of information retrieval concepts in vector spaces. *ACM Trans. Database Syst.*, 12(2):299–321, June 1987. ISSN 0362-5915.

[8] Gerard M. Salton, K. C. Andrew, and Chung Shu Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620,

November 1975. ISSN 0001-0782. doi: 10.1145/361219.361220. URL http://dx.doi.org/10.1145/361219.361220.

[9] Ahmet Koltuksuz and Selma Tekir. Intelligence analysis modeling. *Hybrid Information Technology, International Conference on*, 1:146–151, 2006. doi: http://doi.ieeecomputersociety.org/10.1109/ICHIT.2006.157.

[10] Albert Einstein. The foundation of the general theory of relativity. *Annalen der Physik*, 49(769-822):31, 1916.

[11] Ruslan Sharipov. Quick introduction to tensor analysis. *ArXiv Mathematics e-prints*, math:0403252, March 2004.

[12] Sean Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Benjamin Cummings, 2003. ISBN 0805387323.

[13] Charles W. Misner, Kip S. Thorne, and John. A. Wheeler. *Gravitation*. W. H. Freeman, 1973.

[14] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[15] Jean Gallier. *Advanced Geometric Methods in Computer Science Lecture Notes*. Springer-Verlag, 2011.

[16] Frank Morgan. *Riemannian Geometry: A Beginners Guide, Second Edition*. A K Peters/CRC Press, 2009.

[17] Selma Tekir. *Semantik Bilginin Analizi ve Modellenmesi*. PhD thesis, Ege University, 2010.

[18] Gary Chartrand and Linda M. Lesniak. *Graphs and digraphs (5. ed.)*. Wadsworth and Brooks - Cole Mathematics Series. Wadsworth, 2005. ISBN 978-0-534-06324-5.

[19] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *J. Comput. Syst. Sci.*, 74:1289–1308, December 2008. ISSN 0022-0000. doi: 10.1016/j.jcss.2007.08.006. URL http://dl.acm.org/citation.cfm?id=1460945.1461325.

[20] Bai Xiao and Edwin Hancock. Geometric characterisation of graphs. In Fabio Roli and Sergio Vitulano, editors, *Image Analysis and Processing – ICIAP 2005*, volume 3617 of *Lecture Notes in Computer Science*, pages 471–478. Springer Berlin, Heidelberg, 2005. ISBN 978-3-540-28869-5.

[21] Amit Singer. From graph to manifold Laplacian: The convergence rate. *Appl. Comput. Harmon. Anal*, 21:128–134, 2006.

[22] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. Graph Laplacians and their convergence on random neighborhood graphs. *J. Mach. Learn. Res.*, 8:1325–1370, June 2007. URL http://arxiv.org/abs/math/0608522.

[23] Jon Louis Bentley. Multidimensional divide-and-conquer. *Commun. ACM*, 23(4):214–229, April 1980. ISSN 0001-0782. doi: 10.1145/358841.358850. URL http://doi.acm.org/10.1145/358841.358850.

[24] Michael Connor and Piyush Kumar. Fast construction of k-nearest neighbor graphs for point clouds. *IEEE Trans. Vis. Comput. Graph.*, 16(4):599–608, 2010.

[25] Michael Connor and Piyush Kumar. Practical nearest neighbor search in the plane. In *SEA*, pages 501–512, 2010.

[26] Antal Bosch and Ko Sloot. Superlinear parallelization of k-nearest neighbor retrieval.

[27] Bernard Chazelle. An improved algorithm for the fixed-radius neighbor problem. *Inf. Process. Lett.*, 16(4):193–198, 1983.

[28] Matthew Dickerson and Robert L. (Scot) Drysdale III. Fixed-radius near neighbors search algorithms for points and segments. *Inf. Process. Lett.*, 35 (5):269–273, 1990.

[29] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000. URL http://web.mit.edu/cocosci/Papers/sci_reprint.pdf.