Inference In Latent Random Geometric Graphs

Răzvan - Dumitru Meriniuc[1]

Machine Learning MSc

Peter Orbanz

Submission date: 11th of September 2020

[1]**Disclaimer:** This report is submitted as part requirement for the Machine Learning MSc at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

The graph is a ubiquitous data structure, arising in numerous real-world scenarios, ranging from social networks to drug design. While relatively simple, it is difficult to exploit it in the context of machine learning, which prompts one to find ways of encoding it in a geometric space with a well-defined metric. Our work aims to study the feasibility of treating the embeddings of the nodes of a graph as latent variables, in the context of random geometric graphs (RGG) and Bayesian statistics. By placing a prior on the embeddings and drawing an edge between two points randomly, based on the distance between them, we intend to approximate the posterior of the embeddings, given the graph, which would allow us to better describe the geometry capturing the properties of the graph.

Contents

Chapter 1

Introduction

Learning representations of symbolic data such as graphs, multi-relational data and text is an essential paradigm in artificial intelligence and machine learning. For example, in the field of Natural Language Processing, embeddings of words, such as `WORD2VEC` [1], `GLOVE` [2] and `FASTTEXT` [3] have proven crucial for advancing the field, leading to the excellent results in tasks such as machine translation and sentiment analysis. Similarly, embeddings of graphs, such as `NODE2VEC` [4], `DEEPWALK` [5] and latent space embeddings [6]. `RESCAL` [7], `TRANSE` [8] and Universal Schema [9] are examples of embeddings of multi-relational data that are used in knowledge graphs and information extraction.

The graph is a ubiquitous data structure, which arises in numerous real-world scenarios, ranging from social networks [10] to drug design [11]. Its relative simplicity allows one to attach a myriad of semantics to the nodes and edges, hence its prevalence. However, its format limits the scope of machine learning techniques one can use on such data, but embedding its vertices in a geometry, be it euclidean or with a non-zero curvature, allows us to extend the range of methods we can use while modelling the data. This idea arises naturally, as many networks coming from physical considerations are governed by an un-

derlying geometry, such as the road network in a country. Moreover, it enables us to work in a space which is endowed with more modelling approaches and ties well into the concept of two similar entities being *close* to each other.

We are going to focus on data given as a simple homogeneous graph, i.e unweighted, undirected graph that contains no self-loops and no multiple edges [12]. The homogeneity refers to the edges representing the same concept. Mathematically, we define it as a pair $G = (V, E)$, where V is the set of vertices, also called nodes, and E is the set of edges. We will represent the vertices as consecutive natural numbers, so $V = \{1, 2, \dots N\}$, and the edges as unordered pairs $\{i, j\}$, $i \neq j$, $i, j \in V$. Such a graph can be completely defined by its adjacency matrix, A, $A_{ij} = 1$ if $\{i, j\} \in E$ and 0 otherwise, as we are working with unweighted edges, so we are interested whether an edge exists or not. Since the graph is undirected, $A_{ij} = A_{ji}$.

The aim of this project is to study the feasibility of performing inference on the latent geometric embeddings, in a Bayesian setting. The literature generally focuses on empirical embeddings, targeting the individual nodes, while we are more interested in their underlying distribution. A random geometric graph (RGG) is usually generated by sampling points from some space, which represent the nodes, and connecting each pair by an edge if the distance between them is less than some fixed threshold. In order to avoid this step function and the extra hyperparameter, we generate an edge stochastically, again relying on the distance between the points. The closer two points are, the greater the chance of them being connected.

In the literature, embeddings of nodes [1, 13, 14] , edges [15, 16] and even whole subgraphs [17, 18] have been studied. We are going to focus on embedding of nodes only.

Random geometric graphs are generally studied from a graph theoretical or algorithmic point of view. We aim, however, to describe in probabilistic terms the distribution of the embeddings and perform inference, which has not been studied so far, to our knowledge.

Chapter 2

Background And Related Work

2.1 Bayesian Statistics And Variational Inference

We are going to work in the framework of Bayesian statistics, which relies on the Bayesian interpretation of probabilities, where the probability conveys the degree of belief in an event, which may rely on prior knowledge about the event, whether it is a personal belief or based on results of previous experiments. This is in contrast to the frequentist interpretation, which views probability as the limit of the relative frequency of an event after a large number of trials [19].

The cornerstone of Bayesian statistics is Bayes' formula. If X is some data generated by a process with parameter θ, then

DEFINITION 1 (Bayes' Formula, Posterior, Likelihood, Prior, Evidence)

$$\mathbb{P}(\theta \mid X) = \frac{\mathbb{P}(X \mid \theta)\mathbb{P}(\theta)}{\mathbb{P}(X)}$$

1. *The **posterior** distribution, $\mathbb{P}(\theta \mid X)$, represents our updated belief about the param-*

eter θ after observing the data X.

2. *The **likelihood** distribution, $\mathbb{P}(X \mid \theta)$, denotes the probability of the data being generated by the given the parameter θ.*

3. *The **prior** distribution, $\mathbb{P}(\theta)$, quantifies our assumption about the parameter θ.*

4. *The **evidence** distribution, $\mathbb{P}(X)$, represents the probability of the data occurring, which is computed by marginalising out the parameter θ.*

The central Bayesian principle consists of placing a probability over the parameters, thus giving rise to the prior. It is asymptotically equivalent to the frequentist method, the parameter converging to the true one [20], but the downside is that there is no mathematical decision theory one can rely on when choosing the prior. It is totally up to the practitioner, which allows us to insert our belief into the model, thus forcing us to be honest about the assumptions we make regarding the data and learning process.

Statistical inference is the process of using data analysis to deduce properties of an underlying probability distribution [21]. For example, if we model some data according to a Gaussian distribution with a known variance, we can look at the data and infer what the mean is, and then use it to make predictions on new input points. When predicting, we could work with the parameter for which the data is most likely to occur, an estimate known as MAP (Maximum a posteriori), or average over the parameters in a Bayesian setting, by marginalisation, such that each element of the average has a weight represented by the posterior:

$$\mathbb{P}(\mathbf{X^*} \mid \mathbf{X}) = \int_{\Theta} \mathbb{P}(\mathbf{X^*}, \theta \mid \mathbf{X}) d\theta \tag{2.1}$$

$$= \int_{\Theta} \mathbb{P}(\mathbf{X^*} \mid \theta, \mathbf{X}) \mathbb{P}(\theta \mid \mathbf{X}) d\theta \tag{2.2}$$

$$= \int_{\Theta} \mathbb{P}(\mathbf{X^*} \mid \theta) \mathbb{P}(\theta \mid \mathbf{X}) d\theta \tag{2.3}$$

$$= \mathbb{E}_{\mathbb{P}(\theta|\boldsymbol{X})}[\mathbb{P}(\boldsymbol{X}^* \mid \theta)],$$

where $\mathbf{X}$ is the observed data, $\mathbf{X^*}$ are the points for which we are trying to predict a quantity of interest, Θ is the parameter space and θ is an arbitrary parameter. $\mathbb{P}(\mathbf{X^*} \mid \theta, \mathbf{X})$ becomes $\mathbb{P}(\mathbf{X^*} \mid \theta)$ because the model is fully specified given the parameters, the known points $\boldsymbol{X}$ adding no information.

This integral requires a closed form of the posterior, which is often intractable because we cannot calculate the evidence present in the denominator in **Definition 1**, $\mathbb{P}(\boldsymbol{X})$. Even if it can be calculated, the posterior is not usually the probability density function a known distribution, which would make it more difficult to study and use. A possible solution would be to approximate it with the average $\frac{1}{N} \sum_{i=1}^{N} \mathbb{P}(\boldsymbol{X}^* \mid \theta_i)$, where $\{\theta_i\}_{i=1}^{N}$ are the samples from $\mathbb{P}(\theta \mid \boldsymbol{X})$. The most popular class of sampling algorithms is Markov Chain Monte Carlo (MCMC) [22]. For example, the Metropolis – Hastings algorithm [23] allows us to circumvent the evidence, which cancels out. One of the drawbacks of this method is that it is non-deterministic and, in addition, we would not be able to measure how good our approximation is. Furthermore, there is the problem of choosing the hyperparameters, parameters of the model that are not governed by a prior, as they are just scalars whose value we need to determine. The classical approach aims to maximise the probability of the observed data, $\mathbb{P}(\boldsymbol{X})$, which is the term that generates the intractable integral preventing

us from calculating the posterior in the first place. We could again employ sampling to address this issue, adding an extra layer of randomness to our implementation, which is undesirable. The technique we are about to present deals with the approximation of the posterior with a new variational distribution, offering a measure of *distance* between the distributions, thus allowing us to determine how good our approximation is, which means that there is some quantity that tells us if we are working in the right direction. The setting of this approach offers a lower bound of the evidence as a by-product, while we are making the approximation better, which is where the beauty of this method lies. While it does not provide a hard maximum for the evidence, we get the next best thing, a lower bound.

The measure between distributions is called the Kullback – Leibler (KL) divergence [24], which is defined for any two distributions that share the support of the random variable.

DEFINITION 2 (Kullback – Liebler Divergence)

$$KL(q(u) \,||\, \mathbb{P}(u)) = \int q(u) \log \frac{q(u)}{\mathbb{P}(u)} du$$

We previously referred to this divergence as a way to measure the *distance* between distributions. It is not exactly accurate, since a distance must be symmetrical, while our divergence is not, but it is still useful as a measure of how much one distribution differs from another. By Gibbs' inequality [25], we get that it is non-negative, becoming 0 only when the two distributions are exactly the same.

In our case, we would like to approximate the posterior $\mathbb{P}(\theta \mid \boldsymbol{X})$ with a new distribution $q(\theta)$, having free parameters, that would behave as if it has seen the data. The advantage is that we can pick whatever variational distribution we want, allowing us to insert extra

assumptions that would make calculations easier. This means that we can keep the model *pure* and add simplifications in the approximation. We would thus like to minimise the KL divergence between the two distributions. Because calculating it involves working with the original posterior, which posed problems from the beginning, we rely on the following result, which rephrases the problem as maximising a different term, directly implying the minimisation of our divergence.

$$\log \mathbb{P}(\mathbf{X}) = KL[q(\theta) \,||\, \mathbb{P}(\theta \mid \mathbf{X})] + \mathcal{F}(\theta) \tag{2.4}$$

$$\mathcal{F}(\theta) = \mathbb{E}_{q(\theta)}[\log \mathbb{P}(\mathbf{X}, \theta)] - H(q(\theta)) \tag{2.5}$$

where $H(q(\theta))$ is the entropy of the variational distribution and $\mathcal{F}$ stands for free energy, a concept originally introduced by Karl Friston as an explanation for embodied perception in neuroscience [26].

We can see how maximising $\mathcal{F}$ implies minimising the KL term. Because the divergence is non-negative, we get that

$$\log \mathbb{P}(\boldsymbol{X}) \geq \mathcal{F}(\theta) \tag{2.6}$$

which acts as a lower bound for our evidence. Alternatively, we can deduce the same result, bypassing the KL term, using Jensen's inequality [27] applied to the expected value.

It is worth noting that we have not yet touched the hyperparameters of either distribution. Assuming we could somehow compute $KL[q(\theta) \,||\, \mathbb{P}(\theta \mid \boldsymbol{X})]$ and pick the parameters that minimise it, it is not exactly obvious how doing this would provide a better chance for our data, expressed as a lower bound in our case, since it looks like this just gets the distributions closer disregarding the other terms. And this is the beauty of this technique. That

striving for a better approximation generates a better lower bound for the data, addressing both aspects of the model that raised problems, mentioned at the beginning of this section. In practice, we optimise both sets of hyperparameters at the same time to maximise $\mathcal{F}$, which we should be able to compute effectively. We choose the variational distribution q, so we ought to get a closed form of its entropy, while $\log \mathbb{P}(\boldsymbol{X}, \theta)$ is the model itself, and if we cannot express it, then we know nothing. Thus, we transform this Bayesian inference problem into an optimisation one.

In some cases, we run into terms of the free energy we cannot calculate and have to rely on approximations through sampling. While not ideal, this is done in the same setting and we can still measure how good our approximation of the posterior is, albeit this *distance* is just an estimate as well.

Prediction is performed by approximating $\mathbb{P}(\theta \mid \boldsymbol{X})$ with $q(\theta)$ in the integral at 2.3, thus

$$\mathbb{E}_{q(\theta)}[\mathbb{P}(\boldsymbol{X}* \mid \theta)] \approx \mathbb{E}_{\mathbb{P}(\theta|\boldsymbol{X})}[\mathbb{P}(\boldsymbol{X}* \mid \theta)] \tag{2.7}$$

2.2 Random Geometric Graphs

The beginning of random graph theory is usually attributed to the publication of the seminal papers of Erdős and Rényi [28, 29, 30] introducing the standard random graph model, which studies the behaviour of a random graph with n nodes and N edges, placing an uniform prior on all graphs that respect this requirement. However, around the same time, Gilbert [31] proposed a different random graph model, which relies on a geometric layout underpinning the nodes, edges being determined based on the relative position of vertices. We are going to refer to the latter model by *random geometric graph*.

A Poisson process was employed to model the geometric points, with the idea that they ought to be spread uniformly around the plane with a positive density.

DEFINITION 3 *A Poisson process $\mathcal{P}$ with density one in $\mathbb{R}^2$ is a random subset of $\mathbb{R}^2$ such that:*

1. *The number of points in a measurable set A is governed by a Poisson distribution with the mean equal to the Lebesgue measure of A.*

2. *For two disjoint, measurable sets $A, B \subset \mathbb{R}^2$, the number of points they contain are independent from each other.*

Any spatial process can serve for generating the random points, but in the original model, points are picked in $\mathbb{R}^2$ according to a Poisson process with density one, connecting pairs of points if the distance between them is at most R. So, according to the first property of the definition above, the expected number of neighbours a point has is $N = \pi R^2$. The Poisson process is extensively discussed in [32]. A final version of this model can be constructed by restricting the space to a square of area n, ensuring the number of points, which is a Poisson random variable, has average n. More details on the topic of random geometric graphs can be found in [33]. These stochastic structures are generally studied from an algorithmic point of view or in order to understand their behaviour asymptotically, in terms of the number of various graph statistics, such as connected components. Extensions on this model have been made. Dynamic random geometric graphs, where nodes move in random directions, are analysed in [34], the behaviour for an RGG with a general connection function is studied in [35] and [36] analyses the properties for the case where the points are uniformly distributed on a sphere.

Another level of randomness can be introduced, producing a *soft random geometric graph*, which does not rely on a threshold value, instead using a connection function, ϕ, which takes as a parameter the distance between two points, the output becoming the mean of a Bernoulli random variable which determines the existence of an edge. This was introduced by Waxman [37], using a stretched exponential, $\phi(d) = \beta e^{-\frac{d}{d_0}}$. It can be further generalised by introducing decay in the distance, $\phi(d) = \beta e^{-\left(\frac{d}{d_0}\right)^{\eta}}$.

We aim to use this random structre as the underpinning for a latent space model, more akin to machine learning than algorithmic or graph theory, where it is usually studied. Performing inference in this setting has not been attempted so far, to our knowledge.

2.3 Exchangeability

One can view statistical inference as the procedure of extracting a pattern, which can be interpreted as the *parameter* underlying the model, from observed data. This leads to an understanding of the randomness in the data source as the underlying pattern combined with the sample randomness. When such a common pattern exists and the data is not completely beclouded by the sample randomness, exchangeability properties supply criteria for when extracting the underlying patterns is possible.

In practice, when working with a model in a Bayesian manner, we usually make use of Bayes' formula in the following form

DEFINITION 4 (Bayes' Formula, Posterior, Likelihood, Prior, Evidence)

$$\mathbb{P}(d\theta \mid X_{1:N}) \overset{a.s.}{=} \frac{\prod_{i=1}^{N} p_\theta(x_i)}{\int_{\boldsymbol{T}} \prod_{i=1}^{N} p_{\theta'}(x_i)\mathbb{P}(d\theta')}\mathbb{P}(d\theta)$$

where $\mathbb{P}$ is a prior distribution, $\boldsymbol{T}$ is the parameter space of θ, p_θ is a probability density function and $X_{1:N}$ represents N observations.

We make considerable assumptions, which consist not only of the choice of likelihood density and prior distribution, but also the fact that, given a parameter θ from the random variable Θ, the joint likelihood of the observations factorises, such that

$$p_\theta(x_1, \ldots, x_N) = \prod_{i=1}^{N} p_\theta(x_i) \tag{2.8}$$

Our assumption is that the observations $X_1, \ldots, X_N$ are independently and identically distributed given Θ, which makes them **conditionally independent**. We can mathematically express that as

$$\mathbb{P}(X_{1:N} \in dx_1 \times \cdots \times dx_N \mid \Theta) \overset{a.s.}{=} \prod_{i=1}^{N} \mathbb{P}(X_i \in dx_i \mid \Theta) \tag{2.9}$$

This assumption of conditional independence sits at the core of Bayesian modelling, implying that, given Θ, the randomness attributed to the observations decouples entirely, so all the joint information in the sampled data is enclosed in Θ, which becomes our quantity of interest we would like to extract from the observations. Exchangeability then seeks to provide the context in which we may assume that conditional independence, given some random quantity, is met.

DEFINITION 5 (Exchangeability) *A random sequence $X_{1:\infty} = (X_1, X_2, \dots)$ is said to be **exchangeable** if the order in which the values X_i are observed is irrelevant to their joint distribution. Mathematically, we write this as*

$$(X_1, X_2, \dots) \stackrel{d}{=} (X_{\pi(1)}, X_{\pi(2)} \dots), \quad \forall \textit{bijections } \pi : \mathbb{N} \to \mathbb{N} \tag{2.10}$$

We can see that an i.i.d. sequence is obviously exchangeable, since their distribution is a product, which commutes. The same holds for a conditionally i.i.d. sequence, given some Θ, since it is a product like in 2.8, so the set of conditionally i.i.d. sequences is contained in the set of exchangeable sequences. The seminal result by de Finetti surprisingly proves that these two sets are actually equal, so a sequence $X_{1:\infty}$ is exchangeable if and only if it is conditionally i.i.d, for some Θ.

Defining $P_\theta(\bullet) := \mathbb{P}(X_i \in \bullet \mid \Theta = \theta)$, we have a family of measures

$$M := \{P_\theta \mid \theta \in \boldsymbol{T}\} \tag{2.11}$$

Because Θ is random, P_Θ is a random variable in the set of all probability measures on the sample space $\boldsymbol{X}$, $\mathbf{PM(X)}$, thus a random probability measure. Instead of thinking about Θ as a random parameter for the distribution governing our sequence, we may view it as a random probability measure, so $\mathbf{T} = \mathbf{PM}(\mathbf{X})$, so that $P_\theta = \theta$. Abbreviating the factorial distribution as

$$P_\theta^\infty(dx_1 \times dx_2 \dots) = \prod_{i \in \mathbb{N}} P_\theta(dx_i) \tag{2.12}$$

Let us now introduce de Finetti's Theorem.

THEOREM 6 (de Finetti) *An infinite random sequence $X_{1:\infty}$ is exchangeable if and only if there exists a random probability measure Θ on $\boldsymbol{X}$, such that*

$$\mathbb{P}(X_{1:\infty} \in \bullet \mid \Theta) \overset{a.s.}{=} \Theta^{\infty}(\bullet) \tag{2.13}$$

The theorem usually appears in the form at (2.14), which is a direct consequence of the above, obtained by marginalising out the random probability measure Θ, leading to equality in expectation only. We have that exchangeability implies the equation below, but not the other way around. The right-hand side of the equation below is a mixture, which means that it can be sampled in two stages, by first generating $\Theta \sim \eta$ and then sampling $X_{1:N} \mid \Theta$ from Θ^{∞}, making $X_{1:\infty}$ condtionally i.i.d., which are already exchangeable.

COROLLARY 7 *A random sequence X is exchangeable if and only if*

$$\mathbb{P}(X \in \bullet) = \int_{\boldsymbol{PM}(\boldsymbol{X})} \theta^{\infty}(\bullet)\eta(d\theta) \tag{2.14}$$

for some distribution η on $\boldsymbol{PM(X)}$.

We can alternatively present de Finetti's theorem using random variables instead of distributions. For $\theta \in \mathbf{PM}(\mathbf{X})$ a probability measure on $\mathbf{X}$, we denote the i.i.d. random sequence sampled from θ as

$$X_{\theta}^{0} := (X_1, X_2, \dots), \quad \text{where } X_1, X_2, \dots \overset{i.i.d}{\sim} \theta$$

so $\mathbb{P}(X_{\theta}^{0}) = \theta^{\infty}$. Exchangeability is obtained by randomising θ, i.e. if Θ is a random probability measure on $\mathbf{X}$, then X_{Θ}^{0} is an exchangeable sequence. de Finetti's theorem is

then restated as

$$X_{1:\infty} \text{ exchangeable } \Leftrightarrow X_{1:\infty} \overset{a.s.}{=} X^0_\Theta \quad \text{for some } \Theta \in \boldsymbol{RV}(\boldsymbol{PM}(\boldsymbol{X})) \tag{2.15}$$

This approach will be useful in expressing the more advanced representation theorems presented below, which can be presented more elegantly using random variables instead of distributions.

2.3.1 Exchangeable Arrays

A d-**array** is a type of structure defined as a potentially infinite collection of variables indexed by d indices,

$$x := (x_{i_1,\ldots,i_d})_{i_1,\ldots,i_d \in \mathbb{N}}, \quad \text{where } x_{i_1,\ldots,i_d} \in \boldsymbol{X}_0 \tag{2.16}$$

The sequences previously discussed are 1-arrays and matrices can be seen as 2-arrays, with $\boldsymbol{X}_0$ an algebraic field, so that operations are well-defined. A simple graph, one with no multiple edges, can be represented by a 2-array, having $\boldsymbol{X}_0 = \{0, 1\}$, by its adjacency matrix. An undirected graph implies a symmetric matrix. We will continue to discuss only 2-arrays, but generalisations to generic d-arrays can be made as well [38].

Let X be a random 2-array. We need to fix the components of X we are going to permute in order to define exchangeability. Since the data is structured in rows and columns, it means that there is some semantic attributed to these sub-structures, otherwise we would just express the data as a regular sequence. Therefore, permuting should preserve the rows and columns, i.e. if two data points are located on the same column, they should still share

that column after permuting them, regardless of the new order in which they appear, and similarly for rows. We thus reduce permutations of entries of X to permutations of its rows and columns, by either applying some permutation π to both rows and columns, or use separate permutations, π_r for rows and π_c for columns.

DEFINITION 8 *A random 2-array $X = (X_{ij})_{i,j\in\mathbb{N}}$ is said to be* ***jointly exchangeable*** *if*

$$(X_{ij}) \overset{d}{=} (X_{\pi(i),\pi(j)}), \quad \forall \textit{ bijections } \pi : \mathbb{N} \to \mathbb{N} \tag{2.17}$$

X is said to be ***separately exchangeable*** *if*

$$(X_{ij}) \overset{d}{=} (X_{\pi_r(i),\pi_c(j)}), \quad \forall \textit{ bijections } \pi_r,\ \pi_c : \mathbb{N} \to \mathbb{N} \tag{2.18}$$

We can simply generate a random matrix by defining a function f with its image equal to $\boldsymbol{X}_0$ and takes two arguments. Sampling two random sequences $(U_1, U_2, \dots)$ and $(V_1, V_2, \dots)$ and setting $X_{ij} := f(U_i, V_j)$ generates our random matrix. If the sequences (U_i) and (V_i) contain independent elements and are independent of each other, then (X_{ij}) is separately exchangeable. However, setting $X_{ij} := f(U_i, U_j)$, we generate a jointly exchangeable matrix. However, these are not all of the existing exchangeable arrays. We can, for example, add another argument to the function f, a random variable U_{ij}, without breaking the exchangeability, provided that the sequence (U_{ij}) is made of independent elements. The distribution governing the random variable we made use of is irrelevant, as they do not add any expressive power, so we are free to choose any convenient, simple distribution, such as the uniform distribution on [0, 1].

DEFINITION 9 *Let $\boldsymbol{F}(\boldsymbol{X}_0)$ be the space of measurable functions $\theta : [0, 1]^3 \to \boldsymbol{X}_0$, (U_i) and (V_i) two i.i.d. sequences and (U_{ij}) an i.i.d. 2-array, all elements being Uniform[0, 1]*

random variables. We define two random arrays, J^0_θ and S^0_θ, for any $\theta \in \boldsymbol{F}$, as

$$J^0_\theta := \theta(U_i, U_j, U_{ij}) \tag{2.19}$$

$$S^0_\theta := \theta(U_i, V_j, U_{ij}) \tag{2.20}$$

Remarkably, these two random arrays play an analogous role to that of i.i.d. sequences, in that any exchageable array can be generated by making θ random, result proven by Aldous and Hoover.

THEOREM 10 (Aldous, Hoover) *A random 2-array $X = (X_{ij})$ with entries in a Polish space $\boldsymbol{X}_0$ is jointly exchangeable if and only if*

$$X \stackrel{d}{=} J^0_\Theta, \quad \textit{for some } \Theta \in \boldsymbol{RV}(\boldsymbol{F}(\boldsymbol{X}_0)) \tag{2.21}$$

and separately exchangeable if and only if

$$X \stackrel{d}{=} S^0_\Theta, \quad \textit{for some } \Theta \in \boldsymbol{RV}(\boldsymbol{F}(\boldsymbol{X}_0)) \tag{2.22}$$

DEFINITION 11 *A **graphon** is a measurable function $w : [0,\, 1]^2 \to [0, 1]$.*

Its connection to exchangeable graphs arises when considering a symmetric adjacency matrix, thus working with simple, undirected graphs, as we can replace the three-argument function θ in J^0_θ with our graphon w, which only takes two arguments. For $\theta \in \Theta(\{0, 1\})$ and a uniform random variable U, $\theta(x, y, U) \in \{0,\, 1\}$, so we can define w as

$$w(x, y) := \mathbb{P}[\theta(x, y, U) = 1] \tag{2.23}$$

which makes it a measurable function $[0,\, 1]^2 \to [0,\, 1]$. For a fixed w, we can sample a

random graph G_w^0 as follows:

$$\begin{aligned} U_1, U_2, \ldots &\overset{iid}{\sim} Uniform[0,1] \\ X_{ij} &\sim Bernoulli(w(U_i, U_j)) \end{aligned} \tag{2.24}$$

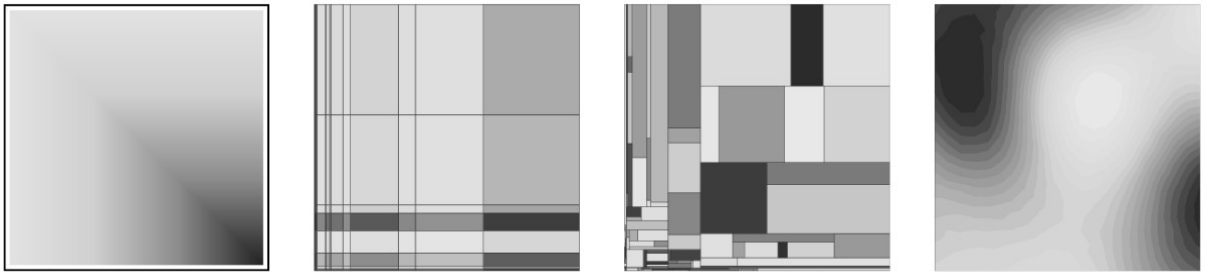


Figure 2.1: Graphons representing different types of random graph models. Left to right: Undirected graph with linear edge density [39], nonparametric block model for separately exchangeable data [40], Mondrian process model for separately exchangeable data [41], graphon with Gaussian process distribution for undirected graph [42]. Figure taken from [38].

These exchangeability theorems have a similar format: There is some random exchangeable structure X, whether it is a sequence or a graph. Then its respective representation theorem introduces some space $\boldsymbol{T}$ and a family of random variable X_θ^0, parametrised by $\theta \in \boldsymbol{T}$. X is then exchangeable if and only if

$$X \overset{d}{=} X_\Theta \quad \text{for some}\, \Theta \in \boldsymbol{RV}(\boldsymbol{T}) \tag{2.25}$$

In practice, the data consists of only a single, large graph. This observed graph with N vertices, $\mathcal{G}_N$, can be seen as a small subset of an infinite graph $\mathcal{G}$, just like N observation in a sequential data set can be interpreted as the first N elements of an infinite sequence. Assuming exchangeability, we know there is a random graphon W such that $\mathcal{G} \overset{d}{=} \mathcal{G}_W$, explaining our vertices as the first N vertices sampled using 2.24.

2.4 Quadratic Forms In Random Variables

We are interested in determining the probability density function of the squared distance between two random variables, X_1 and X_2, governed by the same Gaussian distribution. The squared distance is a random variable, $Y = (X_1 - X_2)^T(X_1 - X_2)$. Since the difference between two Gaussian random variables is itself a Gaussian, we are looking for the probability density function of $U = X^T X$, where $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

Theorem 4.2b.1 in [43] states that the density we are looking for is

$$f(u) = \sum_{k=0}^{\infty} (-1)^k c_k \frac{u^{\frac{n}{2}+k-1}}{\Gamma(\frac{n}{2}+k)}, \quad 0 < u < \infty \tag{2.26}$$

where

$$c_0 = \exp\left(-\frac{1}{2}\sum_{j=1}^{n} b_j^2\right) \prod_{j=1}^{n} (2\lambda_j)^{-\frac{1}{2}} \tag{2.27}$$

$$c_k = \frac{1}{k}\sum_{r=0}^{k-1} d_{k-1} c_r, \quad k \leq 1 \tag{2.28}$$

$$d_k = \frac{1}{2}\sum_{j=1}^{n} (1 - k b_j^2)(2\lambda_j)^{-k}, \quad k \leq 1 \tag{2.29}$$

$$\tag{2.30}$$

with λ_j being the j^{th} eigenvalue of Σ and $\boldsymbol{b} = P\Sigma^{-\frac{1}{2}}\boldsymbol{\mu}$, where P is the matrix with its columns equal to the eigenvectors corresponding to the eigenvalues $\{\lambda\}$, so

$$P^T \Sigma P = \text{diag}(\lambda_1, \ldots, \lambda_n) \tag{2.31}$$

We would ideally like to have the probability density function in 2.26 reduced from a

series. This is possible only if all the eigenvalues have to the same value and $\boldsymbol{b} = \mathbf{0}$, as the recursive interaction in the definition of the c_k terms leads to complicated expressions that do not reduce to a simple formula.

More on this topic can be found in [43], chapters 3 and 4.

Chapter 3

RGG Inference

We look to create a simple setting in which we explore the possibility of using a soft random geometric graph as the latent model. Let us denote the observed data by $\mathcal{G} = \{X_{ij}\}_{i,j\in[N]}$, where $X_{ij} \in \{0, 1\}$ represents the existence of an edge between nodes i and j, assuming nothing that would induce any structure in the given graph. Let $X = \{X_i\}_{i\leq N}$ denote the set of latent embeddings of the vertices, $X_i \in \mathbb{R}^n$, $\forall i \in [N]$. We then model the random variable representing an edge, X_{ij}, as a Bernoulli random variable, with mean $e^{-U_{ij}^2}$, where $U_{ij} = \text{dist}(X_i, X_j)$. This stochastic edge makes the model a soft random geometric graph, as referenced in section 2.2.

For some prior on the embeddings $\mathbb{P}(X)$, we aim to determine the posterior, after observing the data, $\mathbb{P}(X \mid \mathcal{G})$. This implies that we need to determine the probability density function of $U_{ij}^2 \mid X_i, X_j \sim \mathcal{D}$, where $\mathcal{D}$ represents some prior distribution on the embeddings. We look to approximate the posterior by some variational distribution $q(X) \approx \mathbb{P}(X \mid \mathcal{G})$ that we pick ourselves, to circumvent calculating the likelihood $\mathbb{P}(X)$ and be certain that we choose one which is more accessible to work with.

3.1 Spherical RGG

When modelling the embeddings as sitting on a hypersphere, we have two possible approaches to measuring the geodesic, the shortest path between two points:

1. In a spherical geometry, where the edges are curves on the sphere
2. In a Euclidean setting, so the edges go through the sphere

In both frameworks, we need to place a prior over the embedding points. Let us assume we are working with a unit-radius hypersphere, though nothing much changes when working with an arbitrary radius. The simplest distribution would be the uniform one, but that is not interesting enough, as it offers no flexibility, so we opt for the von Mises - Fisher distribution, which is akin to the normal distribution, except the points all have a fixed norm, thus they are located on the same hypersphere. Its probability density function is

$$f_n(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = C_n(\kappa) e^{\kappa \boldsymbol{\mu}^T \boldsymbol{x}} \tag{3.1}$$

where $\kappa > 0$ is the concentration parameter of the distribution around the mean direction $\boldsymbol{\mu}$, $\|\boldsymbol{\mu}\| = 1$. The greater κ is, the more concentrated the points are, with $\kappa = 0$ determining a uniform distribution. The normalisation constant $C_n(\kappa)$ is

$$C_n(\kappa) = \frac{\kappa^{\frac{n}{2}-1}}{(2\pi)^{\frac{n}{2}} I_{\frac{n}{2}-1}(\kappa)} \tag{3.2}$$

where I_u represents the modified Bessel function of the first kind at order u.

Since only the relative position of the points matters, we can pick any mean direction $\boldsymbol{\mu}$, so we can simplify our work by choosing $\boldsymbol{\mu} = (1, 0, \ldots, 0)$, leaving only κ to vary. We need to find the probability distribution function for the distance between two points

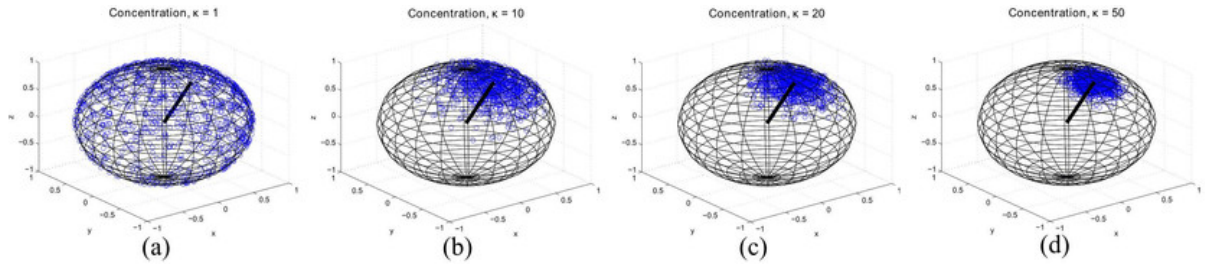


Figure 3.1: von Mises - Fisher pdf for different concentration parameters, in $\mathbb{S}_2$. Figure 3 in [44]

governed by a von Mises - Fisher distribution. In the first framework, the distance between two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is given by

$$d = \cos^{-1}(\boldsymbol{x}_1^T \boldsymbol{x}_2) \tag{3.3}$$

This distance function leads to an intractable integral when calculating the probability density function of the squared distance between two von Mises - Fisher random variables.

If we are to work with Euclidean distances, but the embeddings still lying on the hypersphere, the calculations are relatively more approachable, especially since we assume the mean direction to be a convenient one, namely $\boldsymbol{\mu} = (1, 0, \ldots, 0)$. Nevertheless, we again run into an integral we cannot solve analytically, rendering this avenue ineffective as well.

3.2 Hyperbolic RGG

Hyperbolic geometry is a non-Euclidean geometry referring to spaces of constant negative curvature. Its most famous representative instance is the Minkowski spacetime in special relativity. Hyperbolic spaces started receiving attention in the context of hierarchical

data representation due to the fact that they are naturally endowed to model exponential growth. A tree with branching factor f has $(f+1)f^{l-1}$ nodes at level l and $\frac{(f+1)f^l-2}{f-1}$ nodes on a level less or equal to l, thus the number of children grows exponentially with the distance to the root. This kind of behaviour can be reproduced in hyperbolic geometry using only two dimensions, by placing the nodes at level l on a sphere with radius $R \propto l$, in hyperbolic space. This can be done because hyperbolic spaces grow exponentially with distance, as opposed to their Euclidean counterparts, for which the circle area grows quadratically with the radius.

The work of Krioukov et al. in [45] paved the way for the usage of the hyperbolic space as a space to represent trees, proving that one can interpret them as a discrete version of a hyperbolic space. This served as a basis for [46], a paper that details a method of embedding the `WORDNET` [47] noun hierarchy, which is structured as a tree, hence the obvious latent hierarchical setting. The difficulty of representing spaces of constant negative curvature as subsets of Euclidean spaces leads to multiple equivalent models of hyperbolic spaces, each capturing different aspects of the geometry, but no single model encapsulating all of them. For example, the hyperboloid model is usually used in special relativity, and projecting it in two different ways to disks orthogonal to its main axis yields the Poincaré and Klein unit disks models. Krioukov uses the native representation in his analysis, such that all distance variables have their true hyperbolic values. While the first embeddings of Nickel and Kiela [46] lie in the Poincaré hyperbolic space, other papers use equivalent models [48, 49]. The field of NLP relies heavily on embeddings, so it was natural to extend some of the methods of embeddings words, namely `GLOVE` [2] and `SKIPGRAM` [1], to the hyperbolic spaces, as seen in [48, 50]. Adaptations to working with matrices in the Poincaré disk model have made it possible to use hyperbolic embeddings in deep learning [51, 52], although the field is still in its infancy.

The aforementioned work treats the embeddings algorithmically, without placing a prior over them, and, trying to do that, we can see where the difficulty lies. Working in the two dimensional hyperbolic space of curvature $-\xi^2$, $\mathbb{H}_\xi$, like Krioukov [45], and representing points using their polar coordinates, we have that the distance x between two points (r, θ) and (r', θ') is equal to

$$d = \frac{1}{\xi} \cosh^{-1}(\cosh(\xi r)\cosh(\xi r') - \sinh(\xi r)\sinh(\xi r')\cos(\Delta\theta)) \tag{3.4}$$

where $\Delta\theta = \pi - |\pi - |\theta - \theta'||$. This complicated formula makes it impossible to obtain a closed form for the probability density function corresponding to the distance between two hyperbolic points, for any sensible, non-trivial prior on the points. The same is true for the Poincaré ball of curvature -1, where the distance between two points x and y is

$$d = \cosh^{-1}\left(1 + \frac{2\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\right) \tag{3.5}$$

This space, while endowed with the ability to encode tree-like structures, as opposed to the Euclidean space, which cannot preserve the metric of arbitrary tree structures, even in infinite dimensions [53], comes with metrics that do not lead to a clean probability density function for non-trivial priors on the embeddings, deeming this avenue unfeasible.

3.3 Euclidean RGG

We can start by assuming a simple multivariate Gaussian prior for the embeddings. Since their absolute position is irrelevant, we can simply choose the mean to be $\boldsymbol{\mu}_0 = \mathbf{0}$. As for the covariance matrix, let it be some generic Σ_0 for the time being. We aim to approximate the posterior $\mathbb{P}(\boldsymbol{X} \mid \mathcal{G})$, $\boldsymbol{X} = \{X_i\}_{i\in[N]}$ by a multivariate Gaussian, again with mean $\boldsymbol{\mu} = \mathbf{0}$

and some covariance matrix Σ, hence the approximating distribution q,

$$q(\boldsymbol{X}) \approx \mathbb{P}(\boldsymbol{X} \mid \mathcal{G})$$
$$\boldsymbol{X} \overset{q}{\sim} \mathcal{N}(\mathbf{0}, \Sigma)$$

Working in a variational inference setting, we can rewrite equation 2.5 for the free energy as

$$\mathcal{F}(\sigma) = \langle \log \mathbb{P}(\mathcal{G} \mid \boldsymbol{X}) \rangle_{q(\boldsymbol{X})} - KL[q(\boldsymbol{X}) \mid\mid \mathbb{P}(\boldsymbol{X})]$$

Le us focus on the first term for now, denoting the value of the square distance between two Gaussian embeddings by $u_{ij} = \text{dist}(X_i, X_j)^2$, and capitalising it to represent the random variable associated with it. We need to integrate over the distances between the pairs of embeddings, instead of the embeddings themselves, thus we rewrite

$$\begin{aligned}
\langle \log \mathbb{P}(\mathcal{G} \mid \boldsymbol{X}) \rangle_{q(\boldsymbol{X})} &= \int_{(\mathbb{R}^n)^N} q(\boldsymbol{X}) \log \mathbb{P}(\mathcal{G} \mid \boldsymbol{X}) d\boldsymbol{X} \\
&= \int_{(\mathbb{R}^n)^N} q(\boldsymbol{X}) \prod_{i<j} e^{-u_{ij} X_{ij}} (1 - e^{-u_{ij}})^{1 - X_{ij}} \\
&= \sum_{i<j} \int_{(\mathbb{R}_{>0})^{\frac{N(N-1)}{2}}} q(\boldsymbol{U})(-u_{ij} X_{ij} + (1 - X_{ij}) \log(1 - e^{-u_{ij}})) d\boldsymbol{U} \\
&= \sum_{i<j} \int_{\mathbb{R}_{>0}} q(u_{ij})(-u_{ij} X_{ij} + (1 - X_{ij}) \log(1 - e^{-u_{ij}})) du_{ij} \\
&= \sum_{i<j} \int_{\mathbb{R}_{>0}} q(u_{ij}) \left(-u_{ij} X_{ij} + (1 - X_{ij}) \log \frac{e^{u_{ij}} - 1}{e^{u_{ij}}} \right) du_{ij} \\
&= \sum_{i<j} \int_{\mathbb{R}_{>0}} q(u_{ij})(-u_{ij} X_{ij} - (1 - X_{ij}) u_{ij} + (1 - X_{ij}) \log(e^{u_{ij}} - 1)) du_{ij}
\end{aligned}$$

$$= \sum_{i<j} \int_{\mathbb{R}_{>0}} q(u_{ij})(-u_{ij} + (1 - X_{ij}) \log(e^{u_{ij}} - 1)) du_{ij}$$

$$= - \binom{N}{2} < u >_{q(u)} + \sum_{i<j} (1 - X_{ij}) \int_{\mathbb{R}_{>0}} q(u_{ij}) \log(e^{u_{ij}} - 1) du_{ij}$$

$$= - \binom{N}{2} < u >_{q(u)} + \left(\binom{N}{2} - \underbrace{E}_{\#\text{edges}} \right) \int_{\mathbb{R}_{>0}} q(u) \log(e^{u} - 1) du$$

We now have to calculate the probability density function of the squared distance between two random multivariate normal variables. As presented in section 2.4, in order to obtain a closed formula, we need to make some assumptions. We have already set $\boldsymbol{\mu} = \mathbf{0}$, so now we have to choose Σ such that the eigenvalues all have the same value, therefore we set $\Sigma = \sigma^2 \mathcal{I}$, leading to

$$q(u) = \frac{u^{m-1}(2\sigma^2)^{-m}}{(m-1)!} \exp(-(2\sigma^2)^{-1} u)$$

During the calculation we make an extra assumption, such that n is even, $n = 2m$, $m \in \mathbb{N}$. The full work can be found in appendix A. We then have

$$\langle \log \mathbb{P}(\mathcal{G} \mid \boldsymbol{X}) \rangle_{q(\boldsymbol{X})} = - \binom{N}{2} 2m\sigma^2 + \left(\binom{N}{2} - E \right) \frac{(2\sigma^2)^{-m}}{(m-1)!} \int_{\mathbb{R}_{>0}} u^{m-1} \exp(-(2\sigma^2)^{-1} u) \log(e^u - 1) du$$

The integral does not have a closed form, so we are forced to approximate it using Riemann sums. Since the function under the integral converges to 0 as u increases, we need to identify a large enough value of u such that the function is close enough to 0.

The KL divergence between two multivariate Gaussians of dimension n, $\mathcal{N}(\mu_1, \Sigma_1)$ and

$\mathcal{N}(\mu_2, \Sigma_2)$ equals

$$\frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - n + \mathrm{tr}[\Sigma_2^{-1}\Sigma_1] + (\mu_2 - \mu_1)^T\Sigma_2^{-1}(\mu_2 - \mu_1)\right] \tag{3.6}$$

The hyperparameters for the approximate distribution q and the prior $\mathbb{P}$ are usually optimised together. However, because the hyperparameters of the prior only appear in the KL divergence term, their optimal value equals the value of the hyperparameters of q. We therefore employ a constant prior and set $\Sigma_0 = \mathcal{I}$, the identity matrix. The free energy then becomes

$$\begin{aligned}\mathcal{F}(\sigma) = &-\binom{N}{2} 2m\sigma^2 + \left(\binom{N}{2} - E\right)\frac{(2\sigma^2)^{-m}}{(m-1)!}\int_{\mathbb{R}_{>0}} u^{m-1}\exp(-(2\sigma^2)^{-1}u)\log(e^u - 1)du \\ &- Nm(-\log(\sigma^2) - 1 + \sigma^2)\end{aligned}$$

leaving us to optimise σ only. We can do this by gradient descent, but experiments showed that a fixed step size does not work well for all experiments, unless it very small. However, noticing that the free energy is concave with respect to σ, we can employ binary search to find the value that maximises it.

3.3.1 Relation To Graphons

A special case of the Borel isomorphism theorem allows us to map a Uniform[0, 1] random variable to a multivariate Gaussian, which allows us to work with the latter when generating graphs. Let $\phi : [0, 1] \to \mathbb{R}^n$ be that mapping. If we work with the squared distance between embeddings to randomly determine the existence of an edge, the graphon we are working

with is

$$w(x, y) = e^{-(\phi(x)-\phi(y))^T(\phi(x)-\phi(y))} \tag{3.7}$$

It is important to state that if the embedding space is fixed, as it is the case in our model, the data can be considered exchangeable, but if the space grows with the number of vertices, then the exchangeability property is broken. Recall that the exchangeability amounts to the probability of seeing the data does not change when we swap two vertices, assuming that our data consists of a finite number of elements from an infinite structure. But if the embedding space changes with the number of data points seen, then seeing the value of the last embedding as the first might be impossible or arise with a different probability, hence the lack of exchangeability.

3.3.2 Synthetic Experiments

Assessing model fitness on graph data is not straightforward. For other types of data, cross-validation is usually employed, but using it in a network setting implies subsampling the observed graph, which induces strong assumptions on how the data was generated.

Therefore, we use a protocol devised by Hunter et al. in [54], where models are compared to the data by fixing a set of graph statistics. A fitted model is evaluated by using it to simulate a network, then comparing its statistics with those of the original data. The following statistics presented in [54] capture a range of structures:

- normalised degree statistics d_k, the number of vertices of degree k, divided by the total number of vertices;
- normalised edgewise shared partner statistics χ_k, the number of unordered connected pairs $\{i, j\}$ with exactly k common neighbours, divided by the total number of edges;

- normalized pairwise geodesic statistics γ_k , the number of unordered pairs $\{i, j\}$ with distance k in the graph, divided by the number of dyads.

We also include the number of edges, since the variance σ^2 on each dimension directly determines how close points are, thus modulating the presence of edges. Working in a Bayesian setting, implementing the protocol corresponds to performing posterior predictive checks [55, 56] in the following manner

- Sample the latent embeddings from the approximate posterior, $q(X) \approx \mathbb{P}(X \mid \mathcal{G})$;
- Simulate a graph using the embeddings, $\mathcal{G}^{(s)} \sim \mathbb{P}(\mathcal{G} \mid X)$;
- Calculate the statistics of interest in the simulated graph.

The tables below list the total variation distances between the empirical distribution of each statistic of the observed graph and on the graphs generated from the learnt model. For discrete distributions P and Q with sample space Ω, this distance is calculated as [57]:

$$\delta(P, Q) = \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)| \tag{3.8}$$

Each experiment is run with 300 vertices, and the standard errors are computed over 20 runs, smaller values indicating better fits. We employ various distributions to generate data, representing the vertices, and then draw edges stochastically, based on the distance between them, as presented in the beginning of Chapter 3. The resulting adjacency matrix is then used as data for our model.

By *Original Edges* and *Model Edges* we mean the number of edges of the original geometric graph and the simulated graph using the inferred parameter σ, respectively. *TVD* refers to the total variation distance for the statistics corresponding to the original graph

and the ones of the model.

We start off by generating graphs using a spherical multivariate normal distribution in two dimensions and trying to recover the variance. An example of how such a graph might look can be seen in figure 3.2, albeit for 100 nodes. Table 3.1 shows that the parameter is recovered fairly well for smaller values, but, as σ increases, the recovered parameter does not increase at the same rate, leading to more than five times the number of edges, compared to the original data.

Original σ	Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
0.1	43124.9 ± 106.419	42813.55 ± 155.108	0.222 ± 0.043	0.183 ± 0.066	0.007 ± 0.003	0.109
0.4	27345.25 ± 438.729	29378.55 ± 601.806	0.408 ± 0.029	0.188 ± 0.05	0.045 ± 0.011	0.358
0.7	15331.75 ± 593.006	20350.2 ± 854.684	0.547 ± 0.032	0.399 ± 0.063	0.112 ± 0.014	0.543
1.0	8959.2 ± 509.642	15350.25 ± 955.845	0.619 ± 0.04	0.551 ± 0.066	0.148 ± 0.016	0.694
1.3	5768.0 ± 354.239	12305.05 ± 577.811	0.665 ± 0.038	0.636 ± 0.06	0.303 ± 0.027	0.816
1.6	4064.95 ± 205.598	10115.4 ± 446.654	0.685 ± 0.024	0.67 ± 0.022	0.424 ± 0.02	0.916
1.9	2948.55 ± 255.48	8916.75 ± 742.017	0.729 ± 0.034	0.719 ± 0.058	0.51 ± 0.03	1.012
2.2	2250.65 ± 139.114	7842.2 ± 448.166	0.756 ± 0.026	0.744 ± 0.039	0.554 ± 0.018	1.094
2.5	1724.5 ± 102.614	6669.15 ± 304.363	0.763 ± 0.017	0.757 ± 0.031	0.597 ± 0.026	1.178
2.8	1378.75 ± 101.247	6270.95 ± 410.756	0.789 ± 0.03	0.791 ± 0.038	0.666 ± 0.028	1.251
3.1	1132.65 ± 90.562	5612.65 ± 301.734	0.798 ± 0.03	0.799 ± 0.026	0.702 ± 0.027	1.316
3.4	942.85 ± 54.53	5319.45 ± 362.87	0.812 ± 0.016	0.819 ± 0.027	0.731 ± 0.013	1.377

Table 3.1: Spherical Gaussian data with covariance matrix $\sigma^2\mathcal{I}_2$

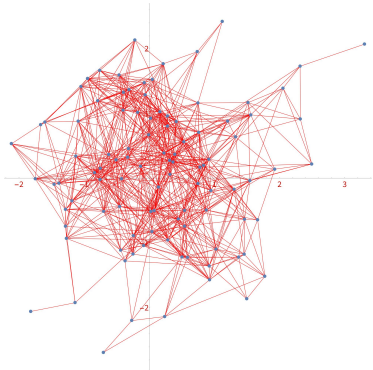


Figure 3.2: Soft random geometric graph using a multivariate Gaussian distribution with covariance matrix $\mathcal{I}$ to sample the nodes

Table 3.2 contains results for synthetic data generated using a multivariate Student t distribution, which gives rise to sparser graphs than the ones generated using a normal

distribution, due to the fatter tail in the probability density function. We can see again that the model severly underestimates the parameter, especially as σ grows larger.

Original σ	Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
0.1	42829.3 ± 143.05	42488.15 ± 220.543	0.227 ± 0.041	0.2 ± 0.062	0.008 ± 0.003	0.116
0.4	26261.0 ± 713.03	28666.35 ± 768.418	0.431 ± 0.035	0.182 ± 0.05	0.054 ± 0.012	0.373
0.7	14390.9 ± 667.362	20057.45 ± 1083.909	0.559 ± 0.037	0.412 ± 0.078	0.126 ± 0.018	0.561
1.0	8333.4 ± 436.009	14479.5 ± 652.509	0.604 ± 0.035	0.505 ± 0.066	0.175 ± 0.019	0.713
1.3	5385.9 ± 218.601	11976.2 ± 793.809	0.667 ± 0.038	0.63 ± 0.065	0.338 ± 0.02	0.835
1.6	3814.3 ± 263.402	9945.45 ± 462.539	0.696 ± 0.028	0.665 ± 0.061	0.458 ± 0.024	0.935
1.9	2739.45 ± 134.691	8670.4 ± 533.589	0.73 ± 0.027	0.718 ± 0.039	0.53 ± 0.02	1.034
2.2	2063.25 ± 189.725	7486.75 ± 595.039	0.751 ± 0.026	0.757 ± 0.033	0.558 ± 0.023	1.123
2.5	1674.75 ± 123.0	6766.5 ± 444.277	0.752 ± 0.031	0.757 ± 0.026	0.62 ± 0.021	1.188
2.8	1254.85 ± 85.145	5946.75 ± 322.501	0.788 ± 0.024	0.797 ± 0.023	0.682 ± 0.017	1.282
3.1	1067.4 ± 75.707	5581.9 ± 403.393	0.794 ± 0.031	0.787 ± 0.035	0.711 ± 0.02	1.336
3.4	888.45 ± 68.564	5035.3 ± 320.16	0.805 ± 0.032	0.799 ± 0.029	0.745 ± 0.024	1.398

Table 3.2: Spherical Multivariate T with covariance matrix $\sigma^2\mathcal{I}$ and 15 degrees of freedom

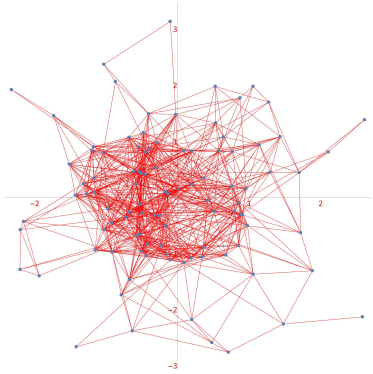
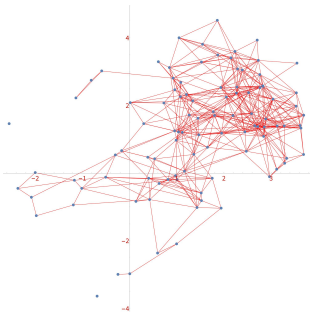


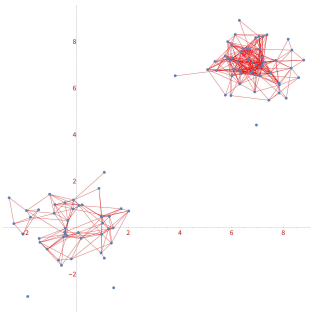
Figure 3.3: Multivariate T with $\Sigma = \mathcal{I}$ and 15 degrees of freedom.

The tables 3.3 and 3.4 contain results for data generated by a mixture of two normal distributions. The former has the distributions closer together, so the shorter distances between nodes leads to edges between vertices generated by different Gaussians, while the latter places the means further apart, resulting in two components that do not communicate. The total variation distances for the d_k and χ_k statistics are comparable, but there is a sharp increase for γ_k, when moving from the mixture with the distributions closer together to the one that has them further apart. This is expected, since γ_k is based on the shortest distances between all pairs of nodes and the separate components give rise to a significant number of pairs that do not communicate, and this behaviour cannot be

reliably captured by a spherical Gaussian posterior.



(a) Mixture of two Gaussians, $\mu_1 = (0, 0)$, $\mu_2 = (2, 2)$, $\Sigma_1 = 1.5\mathcal{I}$, $\Sigma_2 = \mathcal{I}$, $\pi = (0.4, 0.6)$



(b) Mixture of two Gaussians, $\mu_1 = (0, 0)$, $\mu_2 = (2, 2)$, $\Sigma_1 = 1.5\mathcal{I}$, $\Sigma_2 = \mathcal{I}$, $\pi = (0.4, 0.6)$

Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
5082.8 ± 265.968	11342.1 ± 550.784	0.673 ± 0.037	0.623 ± 0.059	0.38 ± 0.023	0.852

Table 3.3: Mixture of two Gaussians, centered at $\mu_1 = (0, 0)$ and $\mu_2 = (2, 2)$, with covariance matrices $1.5\mathcal{I}$ and $\mathcal{I}$, and mixing distribution $\pi = (0.4, 0.6)$.

Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
4104.45 ± 268.468	10177.5 ± 837.822	0.661 ± 0.033	0.579 ± 0.06	0.523 ± 0.018	0.914

Table 3.4: Mixture of two Gaussians, centered at $\mu_1 = (0, 0)$ and $\mu_2 = (7, 7)$, with covariance matrices $1.5\mathcal{I}$ and $\mathcal{I}$, and mixing distribution $\pi = (0.4, 0.6)$.

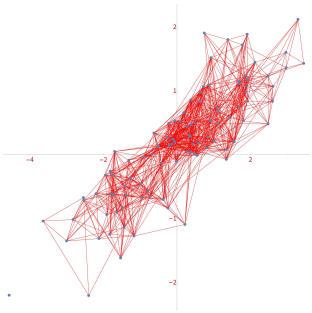
Tables 3.5 and 3.6 contain results for data generated by a non-spherical Gaussian and a non-spherical Student t distribution, respectively. When comparing the total variation distance for γ_k to those of d_k and χ_k, its value is larger than that of a similar graph, in terms of number of edges, generated by a spherical Gaussian or Student t. This can be attributed to the greater diameter of the structure on its widest dimension, leading to a greater mass on higher geodesic values.

Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
8338.75 ± 394.827	14763.45 ± 686.497	0.648 ± 0.038	0.503 ± 0.065	0.305 ± 0.02	0.713

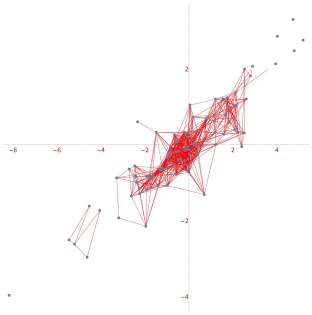
Table 3.5: Gaussian with covariance matrix $\begin{pmatrix} 3 & 1.5 \\ 1.5 & 1 \end{pmatrix}$

Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
7653.9 ± 318.023	14312.55 ± 838.109	0.659 ± 0.043	0.528 ± 0.072	0.341 ± 0.019	0.737

Table 3.6: Multivariate T with 5 degrees of freedom and covariance matrix $\begin{pmatrix} 3 & 1.5 \\ 1.5 & 1 \end{pmatrix}$



(a) Gaussian prior with $\Sigma = \begin{pmatrix} 3 & 1.5 \\ 1.5 & 1 \end{pmatrix}$



(b) Multivariate T with $\Sigma = \begin{pmatrix} 3 & 1.5 \\ 1.5 & 1 \end{pmatrix}$ and 5 degrees of freedom.

So far we have only dealt with $2D$ data, so we ought to vary the number of dimensions as well. As we increase the number of dimensions n, the distances between points decrease, leading to a smaller number of edges. As a result, the largest total variation distances are for smaller n, since for larger n more of the mass is concentrated around a smaller set in the distributions of the d_k, χ_k and γ_k statistics.

m	Original Edges	Model Edges	TVD d_k	TVD χ_k	TVD γ_k	Inferred σ
1	9003.15 ± 441.038	15240.7 ± 690.233	0.609 ± 0.034	0.521 ± 0.067	0.143 ± 0.014	0.692
2	1821.05 ± 138.203	4380.55 ± 259.205	0.512 ± 0.037	0.503 ± 0.069	0.411 ± 0.023	0.74
3	348.9 ± 33.139	1120.65 ± 126.861	0.47 ± 0.031	0.359 ± 0.064	0.562 ± 0.047	0.78
4	72.4 ± 10.2	316.35 ± 37.373	0.395 ± 0.037	0.118 ± 0.06	0.141 ± 0.034	0.783
5	14.55 ± 4.748	110.6 ± 16.951	0.338 ± 0.048	0.044 ± 0.037	0.011 ± 0.004	0.762
6	2.3 ± 1.52	43.9 ± 8.831	0.206 ± 0.032	0.083 ± 0.177	0.002 ± 0.001	0.737

Table 3.7: Gaussian embeddings with identity covariance matrix for different values of n.

The inferred σ is generally underestimated, and, since the variance of the spherical Gaussian distribution we are using as the approximate distribution directly determines how spread the embeddings are, hence the distances between pairs of them, the measured statistics expose subpar results. An obvious problem is the presence of the intractable integral, which leads to worse approximations as σ increases. Another aspect that contributes to the poor results is the inflexible nature of the approximating distribution. The normal distribution is one of the easiest to work with, nevertheless, it still leads to intractable probability density functions for the squared distance between embeddings governed by it, unless we make it spherical.

Chapter 4

Summary and Conclusion

In this thesis, we studied the potential of using random vertex embeddings as latent variables for a given adjacency matrix representing a graph, working in the framework of random geometric graphs. We investigated Euclidean, spherical and hyperbolic spaces and relied on Bayesian variational inference to approximate the posterior.

A possible extention of this model is approximating the posterior using a mixture of Gaussians. However, the probability density function of the squared distance between two random variables governed by different normal distributions has a closed form only if the two distributions share the mean, so not a lot of flexibility is provided by this. One can instead work with the distance as a random variable, instead of the squared distance, thus relying on noncentral χ^2 distribution for the probability density function, although this introduces the modified Bessel function of the first kind which, combined with another intractable integral, might give rise to issues when approximating.

The most challenging aspect of this work was the form of the probability density function for the distance between pairs of embeddings, which forced us to make assumptions

that severly limited our model. Even for something as simple as a spherical multivariate Gaussian prior, we run into an intractable integral and the right parameters are not recovered for simple data. It gets even more complicated when placing the embeddings on the unit sphere or in a hyperbolic space, with little hope of generating sensible results when considering the distances as random variables.

Bibliography

[1] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[3] Piotr Bojanowski et al. “Enriching word vectors with subword information”. In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

[4] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[6] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. "Latent space approaches to social network analysis". In: *Journal of the american Statistical association* 97.460 (2002), pp. 1090–1098.

[7] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. "A three-way model for collective learning on multi-relational data." In: *Icml*. Vol. 11. 2011, pp. 809–816.

[8] Antoine Bordes et al. "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems*. 2013, pp. 2787–2795.

[9] Sebastian Riedel et al. "Relation extraction with matrix factorization and universal schemas". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pp. 74–84.

[10] Lars Backstrom and Jure Leskovec. "Supervised Random Walks: Predicting and Recommending Links in Social Networks". In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. Hong Kong, China: Association for Computing Machinery, 2011, pp. 635–644. ISBN: 9781450304931. DOI: 10.1145/1935826.1935914. URL: https://doi.org/10.1145/1935826.1935914.

[11] D. Duvenaud et al. "Convolutional Networks on Graphs for Learning Molecular Fingerprints". In: *ArXiv* abs/1509.09292 (2015).

[12] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1985, p. 2. ISBN: 9780521288811. URL: https://books.google.co.uk/books?id=Be6t04pgggwC.

[13] Jian Tang et al. "LINE". In: *Proceedings of the 24th International Conference on World Wide Web - WWW '15* (2015). DOI: 10.1145/2736277.2741093. URL: http://dx.doi.org/10.1145/2736277.2741093.

[14] Sami Abu-El-Haija et al. *Watch Your Step: Learning Node Embeddings via Graph Attention*. 2017. arXiv: 1710.09599 [cs.LG].

[15] Zheng Gao et al. “edge2vec: Representation learning using edge semantics for biomedical knowledge discovery”. In: *BMC bioinformatics* 20.1 (2019), p. 306.

[16] Sambaran Bandyopadhyay et al. *Beyond Node Embedding: A Direct Unsupervised Edge Representation Framework for Homogeneous Networks*. 2019. arXiv: 1912.05140 [cs.SI].

[17] Oana Balalau and Sagar Goyal. “SubRank: Subgraph Embeddings via a Subgraph Proximity Measure”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Hady W. Lauw et al. Cham: Springer International Publishing, 2020, pp. 487–498. ISBN: 978-3-030-47426-3.

[18] Bijaya Adhikari et al. “Sub2vec: Feature learning for subgraphs”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2018, pp. 170–182.

[19] John Venn. *The logic of chance*. eng. London: MacMillan, 1876. URL: http://eudml.org/doc/204438.

[20] Stephen Walker and Nils Lid Hjort. “On Bayesian consistency”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.4 (2001), pp. 811–821. DOI: 10.1111/1467-9868.00314. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.00314. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00314.

[21] Graham Upton and Ian Cook. *A Dictionary of Statistics*. Oxford University Press, 2008. ISBN: 9780199541454. DOI: 10.1093/acref/9780199541454.001.0001. URL: https://www.oxfordreference.com/view/10.1093/acref/9780199541454.001.0001/acref-9780199541454.

[22] Gareth O. Roberts and Jeffrey S. Rosenthal. “General state space Markov chains and MCMC algorithms”. In: *Probab. Surveys* 1 (2004), pp. 20–71. DOI: 10.1214/154957804100000024. URL: https://doi.org/10.1214/154957804100000024.

[23] W. K. Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97. eprint: https://academic.oup.com/biomet/article-pdf/57/1/97/23940249/57-1-97.pdf. URL: https://doi.org/10.1093/biomet/57.1.97.

[24] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *Ann. Math. Statist.* 22.1 (Mar. 1951), pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[25] E. Platen. “BRÉMAUD, PIERRE: An Introduction to Probabilistic Modeling. Springer-Verlag, Berlin — Heidelberg — New York — London — Paris — Tokyo — Hong Kong 1988, XVI, 207 pp., 90 Figs., DM 74,—. 3—540—96460—6”. In: *Biometrical Journal* 32.4 (1990), pp. 448–448. DOI: 10.1002/bimj.4710320409. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bimj.4710320409. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.4710320409.

[26] Karl Friston. “Friston, K.J.: The free-energy principle: a unified brain theory? Nat. Rev. Neurosci. 11, 127-138”. In: *Nature reviews. Neuroscience* 11 (Feb. 2010), pp. 127–38. DOI: 10.1038/nrn2787.

[27] J. L. W. V. Jensen. “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”. In: *Acta Math.* 30 (1906), pp. 175–193. DOI: 10.1007/BF02418571. URL: https://doi.org/10.1007/BF02418571.

[28] P. Erdös and A. Rényi. “On Random Graphs I”. In: *Publicationes Mathematicae Debrecen* 6 (1959), p. 290.

[29] Erdös and Rényi. “On the evolution of random graphs”. In: *Publication of Mathematics Institute of Hungian Academy of Sciences* 5 (1960), p. 1761.

[30] P. Erdős and A. Rényi. "On the strength of connectedness of a random graph". English. In: *Acta Mathematica Hungarica* 12.1-2 (Mar. 1964), pp. 261–267. ISSN: 0236-5294. DOI: 10.1007/BF02066689.

[31] E. N. Gilbert. "Random Plane Networks". In: *Journal of the Society for Industrial and Applied Mathematics* 9.4 (1961), pp. 533–543. ISSN: 03684245. URL: http://www.jstor.org/stable/2098879.

[32] Günter Last and Mathew Penrose. *Lectures on the Poisson Process*. English. UK United Kingdom: Cambridge University Press, Dec. 2017. ISBN: 978-1-107-08801-6.

[33] D.M.S.M. Penrose, M. Penrose, and Oxford University Press. *Random Geometric Graphs*. Oxford studies in probability. Oxford University Press, 2003. ISBN: 9780198506263. URL: https://books.google.co.uk/books?id=RHvnCwAAQBAJ.

[34] Josep Diaz, Dieter Mitsche, and Xavier Perez. *Dynamic Random Geometric Graphs*. 2007. arXiv: cs/0702074 [cs.DM].

[35] Carl P. Dettmann and Orestis Georgiou. "Random geometric graphs with general connection functions". In: *Physical Review E* 93.3 (Mar. 2016). ISSN: 2470-0053. DOI: 10.1103/physreve.93.032313. URL: http://dx.doi.org/10.1103/PhysRevE.93.032313.

[36] Alfonso Allen-Perkins. "Random spherical graphs". In: *Phys. Rev. E* 98 (3 Sept. 2018), p. 032310. DOI: 10.1103/PhysRevE.98.032310. URL: https://link.aps.org/doi/10.1103/PhysRevE.98.032310.

[37] B. M. Waxman. "Routing of multipoint connections". In: *IEEE Journal on Selected Areas in Communications* 6.9 (1988), pp. 1617–1622.

[38] Peter Orbanz and Daniel M. Roy. *Bayesian Models of Graphs, Arrays and Other Exchangeable Random Structures*. 2013. arXiv: 1312.7857 [math.ST].

[39] Martin Bálek and Andrew Goodall. "Large Networks and Graph Limits, L. Lovász (2012), xiv + 475 pp." In: *Computer Science Review* 10 (Jan. 2013), pp. 35–46.

[40] Charles Kemp et al. "Learning Systems of Concepts with an Infinite Relational Model". In: *Cognitive Science* 21 (Jan. 2006).

[41] Matej Balog and Yee Whye Teh. *The Mondrian Process for Machine Learning*. 2015. arXiv: 1507.05181 [stat.ML].

[42] James Lloyd et al. "Random function priors for exchangeable arrays with applications to graphs and relational data". In: *Advances in Neural Information Processing Systems (NIPS)* 25 (Jan. 2012).

[43] Arak Mathai and Serge Provost. "Quadratic Forms in Random Variables: Theory and Applications". In: vol. 87. Dec. 1992. DOI: 10.2307/2290674.

[44] Md Hasnat. "Unsupervised 3D image clustering and extension to joint color and depth segmentation". In: (Oct. 2014).

[45] Dmitri Krioukov et al. "Hyperbolic geometry of complex networks". In: *Physical Review E* 82.3 (Sept. 2010). ISSN: 1550-2376. DOI: 10.1103/physreve.82.036106. URL: http://dx.doi.org/10.1103/PhysRevE.82.036106.

[46] Maximilian Nickel and Douwe Kiela. *Poincaré Embeddings for Learning Hierarchical Representations*. 2017. arXiv: 1705.08039 [cs.AI].

[47] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[48] Matthias Leimeister and Benjamin J. Wilson. *Skip-gram word embeddings in hyperbolic space*. 2018. arXiv: 1809.01498 [cs.CL].

[49] Maximilian Nickel and Douwe Kiela. *Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry*. 2018. arXiv: 1806.03417 [cs.AI].

[50] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. *Poincaré GloVe: Hyperbolic Word Embeddings*. 2018. arXiv: 1810.06546 [cs.CL].

[51] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. *Hyperbolic Neural Networks*. 2018. arXiv: 1805.09112 [cs.LG].

[52] Caglar Gulcehre et al. "Hyperbolic Attention Networks". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJxHsjRqFQ.

[53] Christopher De Sa et al. *Representation Tradeoffs for Hyperbolic Embeddings*. 2018. arXiv: 1804.03329 [cs.LG].

[54] David R. Hunter, Steven M. Goodreau, and Mark S. Handcock. "Goodness of Fit of Social Network Model". In: *Journal of the American Statistical Association* 103.481 (2008), pp. 248–258. ISSN: 01621459. URL: http://www.jstor.org/stable/27640035.

[55] George E. P. Box. "Sampling and Bayes' Inference in Scientific Modelling and Robustness". In: *Journal of the Royal Statistical Society. Series A (General)* 143.4 (1980), pp. 383–430. ISSN: 00359238. URL: http://www.jstor.org/stable/2982063.

[56] Andrew Gelman, Xiao-li Meng, and Hal Stern. "Posterior Predictive Assessment of Model Fitness Via Realized Discrepancies". In: *Statistica Sinica* (1996), pp. 733–807.

[57] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006. URL: http://scholar.google.com/scholar.bib?q=info:3wf9IU94tyMJ:scholar.google.com/&output=citation&hl=en&as_sdt=2000&ct=citation&cd=0.

Appendix A

Distance PDF

Let

$$X_i, X_j \sim \mathcal{N}(\mathbf{0}, \frac{1}{2}\sigma^2\mathcal{I}_n)$$

Then

$$Y_{ij} = X_i - X_j \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathcal{I}_n)$$

Denoting the squared distance between X_i and X_j by the random variable

$$D_{ij} = Y_{ij}^T Y_{ij}$$

we find ourselves in the context defined in Mathai, so we have that the pdf of D_{ij} is

$$f(u) = \sum_{k=0}^{\infty} (-1)^k c_k \frac{u^{\frac{n}{2}+k-1}}{\Gamma(\frac{n}{2}+k)}$$

where

$$c_0 = \prod_{j=1}^{n} (2\lambda_j)^{-\frac{1}{2}}$$

$$c_k = \frac{1}{k}\sum_{r=0}^{k-1} d_{k-r}c_r, \quad k \geq 1 \tag{A.1}$$
$$d_k = \frac{1}{2}\sum_{j=1}^{n}(2\lambda_j)^{-k}, \quad k \geq 1$$

with $\{\lambda_j\}$ denoting the eigenvalues of the covariance matrix of Y_{ij}, $\sigma^2\mathcal{I}_n$, so

$$\lambda_j = \sigma^2, \quad j \in \{1, \ldots, n\}$$

implying

$$c_0 = (2\sigma^2)^{-\frac{n}{2}} \tag{A.2}$$
$$d_k = \frac{1}{2}\sum_{j=1}^{n}(2\sigma^2)^{-k}$$
$$= \frac{n}{2}(2\sigma^2)^{-k}, \quad k \geq 1 \tag{A.3}$$

Assuming that $n = 2m$, $m \in \mathbb{N}$, we will prove by induction that

$$c_k = \begin{pmatrix} m+k-1 \\ k \end{pmatrix} 2^{-k-m}(\sigma^2)^{-m-k}, \quad k \geq 0$$

where $\begin{pmatrix} a \\ b \end{pmatrix}$ denotes combinations of a taken b. For the sake of the notation, let us assume that $\begin{pmatrix} 0 \\ 0 \end{pmatrix} = 1$.

From (A.2), we have that

$$c_0 = 2^{-\frac{n}{2}}(\sigma^2)^{-fracn2}$$

$$= \underbrace{1}_{\binom{m-1}{0}} \times 2^{-m}(\sigma^2)^{-m}$$

so the induction hypotesis holds for c_0.

Now, assuming that the hypothesis holds for c_j, $j \in \{0, \ldots, k-1\}$, we will prove it holds for c_k as well. Let us calculate each term of (A.1).

$$\begin{aligned} d_{k-r}c_r &= m2^{-k+r}(\sigma^2)^{-k+r} \times \binom{m+r-1}{r} 2^{-r-m}(\sigma^2)^{-m-r} \\ &= m\binom{m+r-1}{r} 2^{-k-m}(\sigma^2)^{-k-m} \end{aligned}$$

Then

$$\begin{aligned} c_k &= \frac{1}{k}\sum_{r=0}^{k-1} m\binom{m+r-1}{r} 2^{-k-m}(\sigma^2)^{-k-m} \\ &= \frac{m}{k}2^{-k-m}(\sigma^2)^{-k-m}\sum_{r=0}^{k-1}\binom{m+r-1}{r} \end{aligned} \tag{A.4}$$

(A.5)

We will prove by induction over k that $\sum_{r=0}^{k-1}\binom{m+r-1}{r} = \binom{m+k-1}{k-1}$. For $k = 1$ it obviously holds. Assuming our assumption is true for some k, we will prove it holds for

$k+1$ as well.

$$\begin{aligned}\sum_{r=0}^{k}\binom{m+r-1}{r} &= \sum_{r=0}^{k-1}\binom{m+r-1}{r} + \binom{m+k-1}{k} \\ &= \binom{m+k-1}{k-1} + \binom{m+k-1}{k} \\ &= \binom{m+k}{k}\end{aligned}$$

making the induction proof complete.

(A.4) then becomes

$$\begin{aligned}c_k &= \frac{m}{k}\binom{m+k-1}{k-1}2^{-k-m}(\sigma^2)^{-k-m} \\ &= \binom{m+k-1}{k}2^{-k-m}(\sigma^2)^{-k-m}\end{aligned}$$

so the induction is complete. We now have that

$$\begin{aligned}f(u) &= \sum_{k=0}^{\infty}(-1)^k\binom{m+k-1}{k}2^{-k-m}(\sigma^2)^{-k-m}\frac{u^{m+k-1}}{\Gamma(m+k)} \\ &= \sum_{k=0}^{\infty}(-1)^k\frac{(m+k-1)!}{(m-1)!k!}2^{-k-m}(\sigma^2)^{-k-m}\frac{u^{m+k-1}}{(m+k-1)!} \\ &= \sum_{k=0}^{\infty}(-1)^k\frac{\cancel{(m+k-1)!}}{(m-1)!k!}2^{-k-m}(\sigma^2)^{-k-m}\frac{u^{m+k-1}}{\cancel{(m+k-1)!}} \\ &= \frac{2^{-m}(\sigma^2)^{-m}u^{m-1}}{(m-1)!}\sum_{k=0}^{\infty}(-1)^k\frac{1}{k!}2^{-k}(\sigma^2)^{-k}u^k\end{aligned}$$

$$= \frac{2^{-m}(\sigma^2)^{-m}u^{m-1}}{(m-1)!}\sum_{k=0}^{\infty}\frac{(-(2\sigma^2)^{-1}u)^k}{k!}$$
$$= \frac{(2\sigma^2)^{-m}u^{m-1}}{(m-1)!}e^{-(2\sigma^2)^{-1}u}$$