On Frink's Type Metrization of Weighted Graphs

T C VENKATA SIVA, V NIRANJAN REDDY, K JAGAN MOHAN,
ASSISTANT PROFESSOR [1,2,3]
TCSIVA@SVITATP.AC.IN, anjanreddymsc@gmail.com, mohan.kjagan56@gmail.com
department of Mathematics, Sri Venkateswara Institute of Technology,
N.H 44, Hampapuram, Rapthadu, Anantapuramu, Andhra Pradesh 515722

Abstract

In this note, we present, test, and compare an explicit approach to generate a metric d(x, y) between the vertices x and y of an affinity weighted undirected graph using the method of the metrization theorem of uniformities with countable bases.

Keywords: Weighted graphs, uniform spaces, and metrization.

• Introduction

One area of current interest in data analysis is the development of metrics in data sets. It goes without saying that the metrics constructed from a particular data collection should quantitatively represent the affinities between the various data pieces. The search for such metric structures on data sets is motivated by various factors. Specifically, appropriate metrics offer concepts of the vicinity of a given location that are not directly offered by the affinity a priori. More significant, though, is the fact that covering and partitioning may be done with metric spaces and that many of the characteristics of Euclidean spaces remain true.

a metric control that makes sense in every situation. Data structure metrization is important and has been mentioned in several of the seminal works on learning and data analysis, including [1, 2, 3, 4, and 5].

Diffusive metrics, developed by Coifman and Laffon [5], are arguably the most well-known metrization technique. The affinity matrix between the data is used to build a Laplace type operator. Then, the spectral analysis of this operator yields a diffusion kernel that provides a variety of metrics on the data set at various moments. A high dimensional space can be approximated by a low dimensional space by identifying its key properties according to the size of the eigenvalues. The metrization of general topological spaces is a well-known and ancient issue in pure mathematics. Specifically, when the uniform structure has a countable basis, the metrization of the topology generated on a set X by a uniformity on X × X was taken into consideration and solved in [6], see also [7] and [8]. As a result, if and only if the uniformity has a countable basis, a topology generated by a uniform structure is metrizable. Despite the fact that the results appear to have a qualitative nature, they are supported by a quantitative lemma by Frink that makes it possible to derive a metric from the affinity by passing through the uniform structure that the affinity between the data points has created.

This quantitative lemma is originally used by Macias and Segovia ([9]) to demonstrate the equivalence of quasi-distances and powers of metrics. In [10], a Newton type potential form for a universal affinity kernel K on an abstract set X is obtained by providing sufficient conditions on K in terms of a natural metric on X. In general, [10] indicates that, under the assumption of quantitative transitivity, we have that K(x, y) = ϕ(d(x, y)) for a quasi-convex decreasing function ϕ defined on the positive real numbers and some "metric" d.

In order to obtain a metric type function d(x, y) between the vertices x and y associated with an affinity weighted network, we explicitly provide, test, and compare an approach in this note. In fact, the method produces a homogeneous family of metrics that collectively yield a sufficiently abundant family of balls.

The major result as a consequence of Frink's Lemma, as stated and proved in [8], is the focus of the second section of this note. The algorithm for the case of finite X is described in Section 3. We test and compare the technique in a few unique weighted graphs in Section 4.

Pseudometrization of Affinity Kernels and Weighted Undirected Graphs Through Frink's Lemma

The core theory requires no assumption on cardinality, even when the problem is motivated by the finite setting offered by weighted graphs. Therefore, we will assume in this section that X is a set and K: X × X → [0, ∞) is a nonnegative function such that K(x, y) is a measure of affinity between x and y for x and y in X.

A function d: X × X → [0, ∞) such that (p-m.1) d(x, x) = 0 for every x ∈ X; (p-m.2) d(x, y) = d(y, x), x, y ∈ X; and (p-m.3) d(x, z) < d(x, y) + d(y, z) for every x, y, and z ∈ X are examples of a pseudo-metric on the set X.

A metric that has d(x, y) = 0 only when x = y is called a pseudo-metric.

Now, let us state Frink's Lemma as it is presented in Kelley's book [8], namely in

Chapter 6. It's necessary to make some notations to make more statements simpler. We indicate the diagonal of X × X with △. Stated otherwise, △ = {(x, x): x ∈ X}. To represent the set {(x, y) ∈ X × X: (y, x) ∈ U }, given a subset U of X × X, we write U −1. If U = U −1, then we say that U is symmetric. The composition of X × X, given two subsets U and V, is defined as V ≦ U = {(x, z) ∈ X × X: there exist y ∈ X such that (x, y) ∈ U and (y, z) ∈ V }.

Lemma 2.1: Given a set X, specify {Um : m = 0, 1, 2,...} as a series of subsets of X × X that have the given characteristics.

╖U0 = X × X; ╖Un = Un−1 for each n; ╖△ ⊂ Un for each n; • Un+1 ≦ Un+1 ⪊ Un+1 ⊆ Un for each n.

Then, for each n = 1, 2, 3,..., there is a pseudo-metric d defined on X.

{(x, y) ∈ X × X: d(x, y) < 2−n} ⊂ Un+1.

The control that the level sets of the pseudo-metric d have over the provided sequence {Un : n = 0, 1, 2,...} appears to be of a qualitative nature. However, this control becomes quantitative when the sequence Un is itself given by level sets of some function K on X × X, which enables the discovery of a natural notion of distance supplied by K.

In the continuation, we will utilise V (n) to represent the composition V ⪋ V ⪋ V... ⪋ V n times for a given subset V of X × X.

Now let us demonstrate that it is possible to create rising sequences {λ(k): k = 0, 1, 2,...} such that, whenever Uk = {K > λ(k)}, Uk+1 ≦ Uk+1 ⊆ Uk. This may be done under certain moderate conditions in K. Thus, the sequence UK so obtained satisfies Lemma 2.1's primary criteria.

Lemma 2.2: Assume X is a set, and K is a nonnegative symmetric real function defined on X × X such that for every x ∈ X, ˙ K(x, x) = supy∈X K(x, y); ˙ 0 < Λ∞ = sup{α > 0 : {K > α}(m) = X × X for any integer m} ≤ .

Then, for every Λ such that 0 < Λ < Λ∞, there exists a finite sequence 0 = λ(0) < λ(1) \... \ λ(k) = Λ such that, for every i = 1, 2,..., k, {K > λ(i)}(3) ⊆ {K > λ(i − 1)}. Furthermore, for each i = 0, 1, 2,..., k, △ ⊂ {K > λ(i)}.

Evidence. First, note that the set A = > 0: K > (m) = X X for a given integer m is either the entire half line R+ or an interval. The monotonicity of K's level sets implies this fact. Stated differently, if α ∈ A and 0 < β < α, then {K > } ⊃ {K > α}, meaning that {K > }(m) ⊃ {K > ϱ}(m) = X × X and β ∈ A. However, we have that △ ⊂ {K > ϱ} for any α ∈ A. This is inferred from the kernel K's attribute a). In actuality, if K(x0, x0) < α for some x0 ∈ X, then supy∈X K(x0, y) ≤ α, and the point (x0, x0) would belong to {K > ϱ} for no m ∈ N. However, since α ∈ A, {K > α}(m) = X × X ⊃ {(x0, x0)} for some m.

We will select 0 < Λ < Λ∞. We know from the foregoing comments that Λ ∈ A and △ ⊂ {K > Λ}. mΛ = min{m ∈ N: {K > }(m) = X × X} should be set. Stated otherwise, {K > (mŸ) = X × X, but {K > (mŸ−1) ; X × X. We can presume mΛ ≥ 3. The set A1 = {α > 0: {K > }(3) ⊆ {K > }} is now under consideration. The sequence we are searching for contains only two elements, λ(0) = 0 and λ(1) = Λ, if A1 = ∅. Furthermore, it holds trivially that the required inclusion {K > λ(1)}(3) ⊆ X × X = {K > λ(0)}. If Λ1 ∈ A1 and Λ1 > sup A1 − ε for some fixed as small as required and positive ε, then A1~∅. A2 = {α > 0: {K > \1}(3) ⊆ {K > α}} should now be set. We are done with λ(0) = 0, λ(1) = Λ1, and λ(2) = Λ if A2 = ∅. The selection process can be repeated by selecting λi ∈ Ai = {α > 0: {K > Λi−1}(3) ⊆ {K > α}} with Λi > sup Ai − ε. Since the procedure finishes after at most the integer part of m\/3 + one iteration for {K > }(mΛ) = X × X, it yields a finite sequence of levels Ÿ0:= Λ > Λ1 > Λ2 >... > Ÿk. Taking λ(i) = Ÿk−i for i = 0, 1,..., k yields the intended outcome.

It should be noted that selecting the sequence Γi in the above argument for discrete settings or continuous kernels K can be achieved by calculating the maximum of each Ai. Therefore, there's no need for the ε-approximation argument. We may declare and demonstrate the primary findings of this section using the two lemmas mentioned before.

Theorem 2.3. *Let X represent a set. In Lemma 2.2, let K be a nonnegative symmetric function defined on X × X that satisfies a) and b). Then, as stated in Lemma 2.2, for every sequence λ = {λ(i) : i = 0, 1,..., k = k(λ)}, there is a pseudo-metric dλ defined on X such that 1) {K > λ(i)} ⊆ {dλ < 2−i} ⊆ {K > λ(i − 1)} for each i = 1, 2,..., k; 2) the function λ δ = 2−λ−1 ∘ K, where λ−1 is the inverse of any increasing extension of λ(i) to the whole interval [0, k(λ)], is equivalent to the pseudo-metric dλ with constants that are uniform in λ. Specifically, d (x, y) ≤ 2d (x, y) < 4 Ψλ(x, y).*

Evidence. The sequence Ui = {K > λ(i)} from Lemma 2.2 fulfils Lemma 2.1's i) through iv). Therefore, 1) holds for any pseudo-metric dλ defined on X. To demonstrate 2), assume that (x, y) ∈ X × X and that dλ(x, y) > 0. Hence, we get 2−(i+1) ≤ dλ(x, y) < 2−i for some i = 0, 1,..., k(λ).. ≤ −

K(x, y) > λ(i 1) as seen by the second inclusion in 1) and the condition dλ(x, y) < 2−i. K(x, y) = λ(i + 1) is demonstrated by the inequality 2−(i+1) dλ(x, y) and the first inclusion in 1. Since λ represents the inverse function of each strictly increasing extension of the sequence λ(i) for i = 0,..., k to the interval [0, k], we can write: 2−(i+1) ≤ dλ(x, y) < 2−i, and i − 1 < (λ−1 ≴ K)(x, y) ≤ i + 1.

It may be easily deduced from these inequalities that δλ = 2−λ−1 ∘ K is equal to d. Actually,

d (x, y)2(λ−1 ∘ K)(x,y) ≤ 2−i2i+1 = 2. λ4 λ1 = 2−(i+1)2i−1.

Let us note that, regardless of the kernel K or the sequence λ, which fulfil Lemma 2.2, the function δλ in the preceding conclusion satisfies a triangle type inequality with triangular constant equal to 8. For every x, y, and z ∈ X, δλ(x, z) ≤ 4dλ(x, z) ≤ 4(dλ(x, y) + dλ(y, z)) < 8(δλ(x, y) + δλ(y, z)).

λ

Considering the expansion of λ to generate the function λ−1, which is required to provide the quasi-metric δ explicitly, we can note that two extreme scenarios can be provided. Let λ−1 :

[0, λ(k)] → [0, k] actually be defined as follows: λ−1(t) = i for λ(i − 1) < t ≤ λ(i) and i = 1,..., k. Furthermore, λ−1(0) = 0. A lower case λ−1: [0, λ(k)] → [0, k − 1] is another potential λ−1, and it is represented by λ−1(t) = i − 1 for λ(i − 1) < t ≤ λ(i) for i = 1,..., k.

It is also important to note that the scaling factor related to the selection of Λ in Lemma 2.2 is not reflected in Frink's metric, and consequently, in δλ. The reason for this is that only values between zero and one are accepted by Frink's measure, dλ. Because δλ is equal to dλ, our quasi-metric δλ is likewise bounded.

Additionally, a family of δλ balls that are directly defined as level sets of the affinity kernel K are contained in the sequence λ(i).

r Proposition 2.1 states that the open λ ball with radious r and centre at x in X is given by for 0 < r < 1.

y ∈X: K(x, y) > λ(log2 1)} is the result of Bδ⍰{(x, r).

Evidence. The condition δλ(x, y) < r, which determines Bδ, is similar to K(x, y) > λ(log2 1).

(x, r).

r λ

It is important to note that K alone will determine how the sequence λ(i) is really constructed. Therefore, K is the only one who can supply the 쌨 balls.

- # **The Algorithm for the Explicit Computation of the Sequences *λ*. The Finite Case**
- We examine the scenario where X = {1, 2,..., n} for a large integer n in this section. It is possible to think of the kernel K defined on X × X as a n × n symmetric matrix with positive entries Kij. Since Λ∞ ≥ min Kij > 0, hypothesis b) in Lemma 2.2 holds trivially because each Kij is positive. Alternatively, if Kii = supj Kij, then Lemma 2.2's hypothesis a) is true.
- To build the sequences λ and δλ related to this matrix K, we will have to deal with the diagonal's neighbourhood composition in the algorithm.
- Let {1, 2,..., n}2 = X × X consist of two subsets, U and V. Then, for some j = 1, 2,..., n}, V ≦ U = {(i, k): (i, j) ∈ U and (j, k) ∈ V, as before.
- Section 3.1. Set AU = (aij(U)) to represent the n × n rest matrix for a given U ⊆ {1, 2,..., n}n, which is defined by aij(U) = 1 of (i, j) ∈ U and aij(U) = 0 otherwise. The non-vanishing entries of the product matrix AU AV thus produce the set V ≦ U. Specifically

-
- roof. Note that if and only if there exists k ∈ {1,..., n} such that akj(V) = 1 and aik(U) = 1, then ∑n}aik(U)akj(V) ≥ 1. Stated differently, if and only if (i, k) ∈ U and (k, j) ∈ V, in the appropriate manner.
- The following conclusion is significant because it illustrates the point at which the iterated composition of a diagonal neighbourhood eventually encompasses the entire space {1, 2,..., n}.2.
- 2.
- Lemma 3.1. Assume that U is a set in {1, 2,..., n}2 that includes each of the three major diagonals of {1, 2,..., n}. For each i = 1, 2,..., n, precisely, (i, i − 1), (i, i) and (i, i + 1) belong to U. Next, let m be such that U (m) = {1, 2,..., n}.2. Evidence. We know that the matrix AU contains ones in at least the three major diagonals based on the representation of U in terms of the matrix AU and the current hypothesis in U. Stated differently, ai,j >= 0, U ai,i = ai−1,i = ai,i+1 = 1. Then, at least in the entries of the five diagonals, A2 has positive values.
- 1 1△ = {(i, i) : i = 1,..., n}, △+ = {(i, i + 1) : i = 1,..., n − 1}, △− = {(i − 1, i) : i = 2,..., n}, 2 2△+ = {(i, i + 2) : i = 1,..., n − 2} and △− = {(i − 2, i) : i = 3,..., n}. An iteration of the previous reasoning demonstrates that U's composition widens around the diagonal and that the set {1,..., n}2 is fully covered after a finite number of compositions.
- We are now prepared to outline the fundamental stages of an algorithm designed to identify a sequence λ(i) associated with the kernel K.
-
- algorithm. Let K = (Kij) be a positive-entry n × n symmetric matrix.
-
-
- Step 1: Determine K's minimum value on each of the three major diagonals. Min{Ki−1,i; Ki,i; Ki,i+1: i = 1,..., n}, Λ0
- Step 2: As in Proposition 3.1; 0 construct the matrix A0 = A{(i,j):Kij ≥Λ0}.
- Step 3: Determine A3; 0
- Step4: Define U0 as the subset of all (i, j) in {1,..., n}2 such that A3's (i, j) entry is positive;
- Using the formula from Proposition 3.1, step 5. Find Λ1 = max{α : {K ≥ α} ⊆ U0}; step 6. Create the matrix A1 = A{(i,j):Kij ≥Λ1}.
- Step 7: Determine A3; 1
- Phase 8: Establish U1 = {(i, j): the A3 entry (i, j) is positive};
- Step 9: Determine Λ2 = max{α : {K ≥ α} ⊆ U1};... The sequence Λ0, Λ1,..., Λk is obtained when the iteration terminates after a finite number of steps. Λk < Λk−1 < · · · < Λ2 < Λ1 is evident. In the absence of any further conditions on K, it is possible for Γ0 ≤ Λ1. However, we have Λk < Λk−1 \ · · · \~ \2 \ Λ1 < Λ0 if Λ0 is greater than all of K's entries outside of the three main diagonals.
- Proceed k + 1. Assign λ(i) = Λk−i for all i = 0,..., k; Step k + 2. Determine an iteration of λ−1; λ
- Proceed k + 3. Let δ (i, j) = 2−λ−1(Kij) be defined.
-
- For each i that is fixed and 0 < r < 1, plot the δλ balls Bδλ {(i, r) = {j : Kij > λ(log2 1)}.
-
-
- r
- This is the Python script that implements this algorithm.
-
-
-
-
- import numpy as np Listing 1: Python Algorithm
-
- m a tp lo t l ib import. connect networkx as nx import pyplot as p l t
-
- ## Value of n = n ## Calculate K's minimum Kmin= np. amin (K)
- ## Determine Lambda 0: Lambda 0=0
-

- ze ro s ((n)) for i in range (n−1): aux=np
- Comparing the inner values of the major d iagonals and aux [i] equals min (K[i, i],K[i, i +1]).
- ## Examine the residual values in the primary d iagonals aux [n-1]=K[n-1,n-1] lambda 0=min (auxiliary)
-
-
-
- The matrix A is defined as follows: A=np. ze ro s ((n, n))
-
- When i and j are in the same range (n), their formulas are as follows: i f K[i, j]>=lambda 0: A[i, j]=1.
-
- ## CalculateB=(A. dot (A)). dot (A) B=A^3
-
- ## Calculate Bpos
- Ze ro s ((n, n)) for i in range (n): for j in range (n): i f B[i, j] >=1: Bpos [i, j]=1 Bpos=np
-
- ## Calculate C = K * Bpos ## Determine the lowest of the p o s i t i v e values of C auC=np.max(K) for i in range (n): regarding j in range (n): When C[i, j] >0, then auxC=min (auxC, C[i, j]) lambda 1 equals auxC
-
- ## Indices ## Attributes
-
- ze ro s ((n)) lambda i [0] = lambda 0 lambda i [1] = lambda 1 lambda i=np
-
- ze ro s ((n, n, n)) = A i=np
- I = A i [0, :, :]
-
- •i=np. ze ro s ((n, n, n)) B i [0, :, :] =B
-
- Ze ro s ((n, n, n)) Bpos i=np
- Bpos = Bpos i [0, :, :]
-
- •i=np. ze ro s ((n, n, n)) C i [0, :, :] =C
-
- ## When h equals one
-
-
- lambda i [h]>Kmin while: ## Define matrix A as follows for i in range (n): i f K[i, j]> for j in range (n).i [h] = lambda i: A i [h, i, j]=1
-
- ## Calculate I [h, :, :] = (A i [h, :, :]) B=A^3. A i [h, : , :]))) dot. (A i [h, :, :]) dot
-
- ## Bpos for i in range (n): for j in range (n): i f B i [h, i, j] >=1: Bpos i [h, i, j]=1
-
- ## Determine C C i [h, :, :] Bpos i =K∗ [h, :, :]
- ## Determine the minimum of the p o s i t i v e values o f C with the formula auC=np.max(K) for i in range (n): for j in range (n): i f C i [h, i, j] >0: auC=min (auxC, C i [h, i, j]) [h+1] lambda i = auxC h+=1
-
- ## Conclude whilst
-
- ## Moving Lambda Around
- lambda I equals lambda I [0: h+1] [: : − 1] lambda i=lambda i
- ## Reversing the function of Lambda funct inv (t, lambd) def lambda :
- If t is less than zero, print (' t must to be less than or equal to the minimum value of lambda')
- For each kk in the range (le n (lambd) −1), i f lambd [kk]<=t\lambd [kk + 1]: inv=kk+1
- For any t>=lambd [le n (lambd) − 1], the formula is inv=le n (lambd) return inv.
-
- ## Calculate the matrix (nodo1, nodo2) def d i s t f r i n k i n v:
- Finv =2∗∗(−lambda funct inv (K[nodo1, nodo2], lambda i)) Dist Finv
- return to this Finv
-
-
-
- d i s t a r r a y F i n v=np. ze ro s ((n, n)) for v in range (n): for w in range (n): d i s t a r r a y F i n v [v, w]= d i s t f r i n k i n v (v, w)
-
- Create the graph s t a r t i n g by using K G = nx. Graph () from numpy matrix (np. matrix (K)) to G = nx

-
- Plot the graph layout as follows: ## nx.spr in g la yo u t (G)
-
- f i g u r e (p l t)
- p l t. t i t l e (' Graph') n o d e c o lo r=np. ones (n) nx. draw (G, layout, n o d e c o lo r=no de co lo r, w i t h l a b e l s=False) nx. draw ne two rkx labe ls (G, layout, f o n t s i z e =12, f o n t f a m i l y=' sans−s e r i f') p l t. show ()
-
- ## Sketching b a l l s centred at i for k in range (n): for v in range (h+1): i f d i s t a r r a y F [i] [k] > lambda i [v]: n o d e c o lo r [k]=h−v n o d e c o lo r [i]=h+1

Test and Comparison with the Diffusive Metric for Newtonian Type Affinities

The results in [10] suggest testing the algorithm on affinities defined as discretizations of Newtoniantype potentials of the form

$$K^{\alpha}(x, y) = \frac{1}{|x-y|^{\alpha}}$$

}
{

for α positive. Once a discretization of K_α is given we may run our algorithm and also the well known diffusion metric introduced in [5]. See also [11]. Let us recall that the diffusive metric at time $t > 0$ is given by

l

where x^l, v_l, $l = 1, \ldots, L$ are the eigenvectors and the eigenvalues of the Laplace operator on the graph with affinity given by the metric K_{ij}.

We shall only write down the comparison of the families of δ_λ-balls, d_t-balls and Euclidean balls for a couple of values of the radio, when we consider the discretization

It is worthy pointing out at here that the choice of 60 points of discretization is only taken for the sake of getting better images for the graphs. In particular for the visibility of some edges.

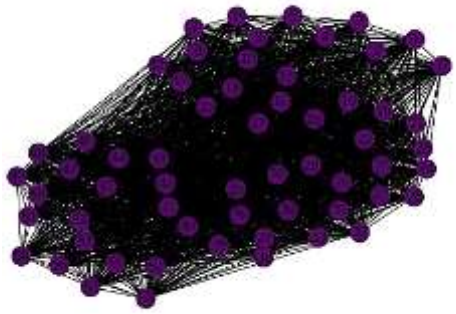
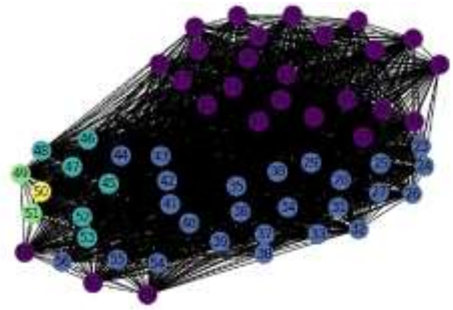
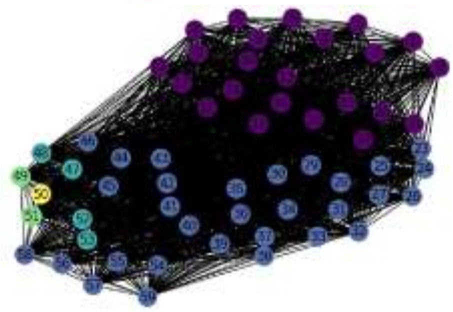


Fig. 1. Graph

Let us also point out that in the following graphs, the numerical label of each vertex is assigned according to the order of the rows in the affinity matrix, but a priori has nothing to do with distanceor affinity.

Fig. 1 labels with the integers $0, 1, \ldots, 59$ the 60 vertices of our graph.

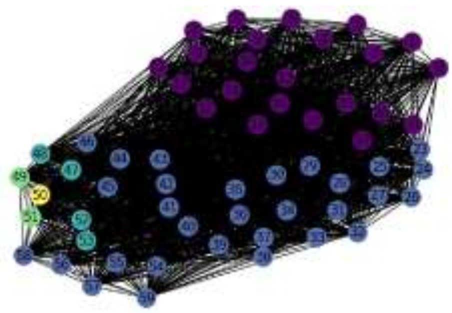




(D) Y, G, 0.11, T, 0.135, L, 0.31, P, **(F)** Y, 0.0169492, G, 0.037037, T, 0.111111,

0.404327

L, 0.333333, P,
1



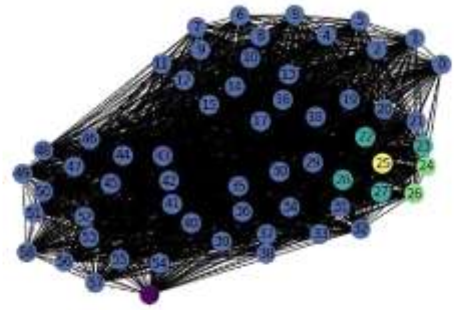
(E) Y, G, 1, T, 3, L, 27, P, 59

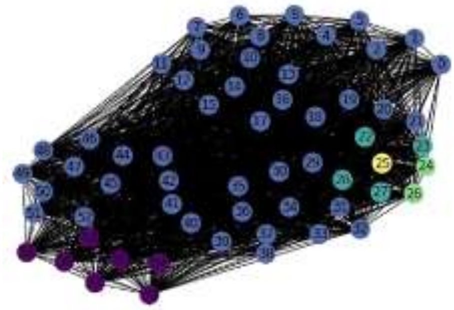
Fig. 2. Center at 50

Now, for each of the three metrics—the Euclidean metric (E), the diffusive metric (D) with t = 0.005, and the Frink's metric—we will depict several balls with centres at 25 and 50. Since K is constructed in terms of the Euclidean (E), it is imperative to compare (D) and (F) with (E). Again, let us state that the form of the balls is what interests us, not the specific radii at which they are obtained. In this instance, where the Euclidean metric is unbounded, this point is very evident. Nevertheless, we will include the radii for each ball in each measure that are depicted. Actually, the annuli between two successive balls are depicted in various colours in the following photos. We utilise green for the first annulus, turquoise for the second, lavender for the third, and purple for the final annulus, with yellow serving as the centre.

The colours are denoted in Figs. 2 and 3 by the capital letters Y, G, T, L, and P. The inner and outer radii of each annulus are indicated by the letter and number sequences.

It is important to note that the raddi sequence for (D) has been selected so that the dt balls approach Euclidean balls as closely as feasible. The metrization approach (F) described here appears to recreate, at least for this basic case of a kernel defined by a metric, the exact forms of the balls associated with the metric defining the kernel. One could argue that Frink's construction's exponential nature only yields a small number of the graph's balls. However, we know from the very proof of our primary result that we can obtain a profuse diversity of sequences λ(i) by just modifying the initial parameter Λ < Λ∞. The use of our affinity matrix K's main three diagonals is another somewhat discretionary algorithmic step. Restarting with the primary five diagonals will yield an additional family of annuli and F-balls.





(D) Y, G, 0.13, T, 0.17, L, 0.212, P, 0.404327

(F) Y, 0.0169492

L, 0.333333, P, 1

(E) Y, G, 1, T, 3, L, 27, P, 59

Fig. 3. Center at 25

Acknowledgement

The Ministry of Science, Technology, and Innovation (MINCYT) in Argentina, the Consejo Nacional de Investigaciones Cient´ficas y T´ecnicas (CONICET) and the Agencia Nacional de Promoci´on de la Investigaci´on, el Desarrollo Tecnologico y la Innovaci´on (Agencia I+D+i) provided funding for this work.

References

• Mark Tygert, Serkan Piantino, Arthur Szlam, Joan Bruna, Yann LeCun, and Soumith Chintala. A mathematical justification for convolutional networks with complex values. 815-825 in Neural Computation, 2016; 28(5). PMID: 26890348.

• Geoffrey E. Hinton, Yoshua Bengio, and Yann LeCun. profound understanding. 2015;521(7553):436-444; Nature.

• Andrew Zisserman, Karen Simonyan. Deep convolutional networks for large-scale picture recognition. Yann LeCun and Yoshua Bengio, editors. Conference Track Proceedings of the Third International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.

Coifman RR, Warner F, Lafon S, Lee AB, Maggioni M, Nadler B, and Zucker SW. Diffusion maps are a geometric diffusion tool for harmonic analysis and data structure definition. 2005;102(21):7426–7431; Proceedings of the National Academy of Sciences.

Coifman, Ronald R., and Stéphane Lafon. maps of diffusion. Harmon. Anal. Appl. Comput. 2006;21:5-30.

Frink AH. The metrization problem and distance functions. American Math. Soc. Bull. 1937;43(2):133–142.

Chittenden EW. Regarding the metrization issue and associated issues in the abstract set theory. Amer. Math. Soc. Bull. 1927;33:13–34.

Kelley, John L. Topology in general. Reprint of the 1955 edition [Van Nostrand, Toronto, Ont.], Graduate Texts in Mathematics, No. 27, Springer-Verlag, New York-Berlin, 1975.

Carlos Segovia and Roberto A. Macías. Lipschitz functions in homogeneous type spaces. Advances in Mathematics, 1979;33(3):257–270.

•Ivana G´omez, Hugo Aimar. Both distance and affinity. Regarding the Newtonian framework of certain data structures. 2018;6:89-95; Anal. Geom. Metr. Spaces.

Bronstein MM, Vandergheynst P, LeCun Y, Bruna J, Szlam A. Beyond Euclidean data: Geometric deep learning. 2017;34(4):18–42; IEEE Signal Processing Magazine.