7-19-2019

Merge Trees in Discrete Morse Theory

Benjamin Johnson
Ursinus College, bejohnson@ursinus.edu

Merge trees

Benjamin Johnson

May 2019

Abstract

The field of topological data analysis seeks to use techniques in topology to study large data sets. The hope is that rather than single quantities that summarize the data, such as mean or standard deviation, information about the data can be learned by studying the overall "shape" of the data. One way to summarize this data is through a merge tree. Merge trees can be thought of as keeping track of certain clusters of data and determining when they merge together. In this paper, we will study merge trees induced by a discrete Morse function on a tree. Under a suitable notion of equivalence of merge trees, we then count the number of merge trees that can be induced on a star graph.

1 Introduction

Topological data analysis seeks to understand a set of data by studying topological properties of that data. One highly successful tool in this regard is persistent homology. Persistence has been used to study statistical mechanics [19], hypothesis testing [6], image analysis [7], complex networks [18], and many other phenomena. There has been recent interest in studying merge trees, a special kind of persistence [15, 17]. Part of the advantage of studying a merge tree over the persistence diagram is that the merge tree gives more detailed information about precisely which components merged with which other components. It tracks not only the lifetime of a component but its evolution as well.

In [9], Justin Curry studies functions on the unit interval that have the same persistent homology. In this smooth setting, Curry develops a merge tree associated to a Morse set, an abstraction of path components associated to a Morse function on a compact, connected manifold. He is then able to count merge trees under a suitable notion of equivalence. In this paper, we take up a similar problem in a purely discrete setting; that is, given a discrete Morse function on a tree (i.e. 1-dimensional abstract simplicial complex), we associate a tree, appropriately called a merge tree (Definition 9). Under a notion of equivalence of merge trees, we then count the number of merge trees on a star graph S_n.

2 Background

2.1 Graphs and trees

Let $G = (V(G), E(G))$ be a finite, loopless graph without multi-edges (i.e. a 1-dimensional abstract simplicial complex). We call an edge or a vertex of G a **simplex**. If $e = uv$ is an edge, we say that the edge e is **incident** with vertex v and that u and v are adjacent.

We work exclusively with trees in this paper. Here we recall several important characterizations of trees. They will be utilized without further reference.

Theorem 1. (Characterization of trees) Let G be a connected graph with v vertices and e edges. The following are equivalent:

a) Every two vertices of G are connected by a unique path.

b) $v = e + 1$.

c) G contains no cycles.

d) $b_1(G) = 0$ where b_1 is the first Betti number of G ([10, Chapter II.4]).

e) The removal of any edge from G results in a disconnected graph.

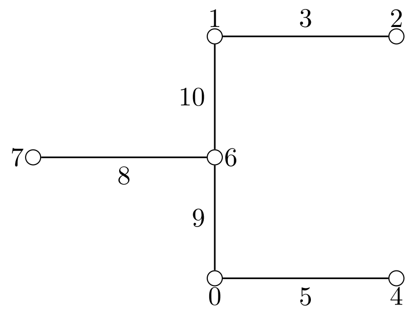
A connected graph that satisfies any of the above characterizations is called a **tree**. Proofs of the equivalence of the statements may be found in any graph theory textbook (e.g. [8, Chapter 2.2]). A disconnected graph F such that each component of F is a tree is called a **forest**. For any vertex $v \in F$, we let $F[v]$ denote the connected component of F containing v. It immediately follows that if F is a forest with two distinct vertices $u, v \in F$, then there is a path between two vertices u and v if and only if $F[u] = F[v]$.

2.2 Discrete Morse theory

Our references for the basics of discrete Morse theory are [11, 13, 14]. There are several different ways of viewing a discrete Morse function. For our purposes, we make the following definition:

Definition 2. Let G be a graph. A function $f\colon G \to \mathbb{R}$ is called **weakly increasing** if $f(v) \leq f(e)$ whenever $v \subseteq e$. A **discrete Morse function** $f\colon G \to \mathbb{R}$ is a weakly increasing function which is at most 2–1 and satisfies the property that if $f(\sigma) = f(\tau)$, then either $\sigma \subseteq \tau$ or $\tau \subseteq \sigma$. Any simplex σ on which f is 1–1 is called **critical** and the value $f(\sigma)$ is a **critical value** of f.

Example 3. Define the function f on T as follows:



Then f is a discrete Morse function. Note that all values are critical.

Definition 4. Let G be a graph. Given $a \in \mathbb{R}$ the **level subcomplex** G_a is defined to be the induced subgraph of G consisting of all simplices σ with $f(\sigma) \leq a$. For each critical value $c_0 < \ldots < c_{m-1}$ of f, we consider the induced sequence of level subcomplexes $\{v\} = G_{c_0} \subset G_{c_1} \subset \ldots \subset G_{c_{m-1}}$. In the sequel, we will use the notation $G_{c_i - \epsilon}$ to denote the level subcomplex immediately preceding G_{c_i}; that is, ϵ is chosen so that $f(\sigma) < c_i - \epsilon < c_i$ for every $\sigma \in G$ such that $f(\sigma) < c_i$.

3 Merge trees

In this section we introduce merge trees, our main object of study.

3.1 Basics of merge trees

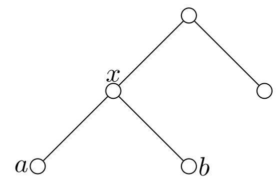
Definition 5. A **binary tree** is a rooted tree where each vertex has at most two children, and each child is designated as its **left** or **right** child. A binary tree is **full** if every vertex has 0 or 2 children.

Definition 6. A **Merge tree** is a full binary tree reflected across a horizontal axis that does not intersect the tree. A node with exactly one neighbor is a **leaf node** or leaf. Otherwise, a node with more than one neighbor is an **internal node**.

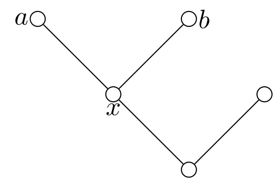
We use the term node for merge trees and vertex for graphs.

Remark 7. Because we flip across a horizontal axis, left (L) and right (R) maintain their same relationship to the original node. Not only do we maintain the language of left and right, but also“child" and “parent." When considering the merge tree, this will look backwards as a child will be above the parent.

Example 8. Consider the full binary tree below:



The corresponding merge tree is given by reflecting across a horizontal axis:



Note that in both the binary tree and the merge tree, we say that a and b are children of x, even though in the merge tree this appears to be reversed.

To any discrete Morse function, we are able to associate a merge tree through the following construction.

Theorem 9. Let $f: T \to \mathbb{R}$ be a discrete Morse function on a tree T. Then f induces a merge tree $M_f = M$.

Proof. Let $f: T \to \mathbb{R}$ be a discrete Morse function with critical values $c_0 < c_1 < c_2 < \ldots < c_m$. We will construct a labeled merge tree whose node set is in 1–1 correspondence with the vertex and edge set of T. If $v \in T$ and $f(v) = a$, the corresponding node in M is denoted n_v or n_a, and if $e \in T$ is an edge with $f(e) = b$, the corresponding node in M is denoted n_e or n_b. Since critical values are distinct, there should be no confusion. We will furthermore equip each node in M with both a label and a direction (L or R). Each node of M is given the same label or value as its corresponding simplex in T under the discrete Morse function f. We now construct M inductively on the critical edges of f in reverse order.

Begin by creating a node n_{c_m}, corresponding to the critical edge labeled c_m, with label c_m along with the direction L.

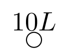
Now let n_{c_i} be a node of M corresponding to an edge edge $uv \in T$. Create two nodes above n_{c_i} with labels $\max\{f(\sigma) : \sigma \in T_{c_i - \epsilon}[u], \sigma \text{ critical}\}$ and $\max\{f(\sigma) : \sigma \in T_{c_i - \epsilon}[v], \sigma \text{ critical}\}$ (see Definition 4 for meaning of $T_{c_i - \epsilon}$). If $\min\{f(\sigma) : \sigma \in T_{c_i - \epsilon}[u]\} < \min\{f(\sigma) : \sigma \in T_{c_i - \epsilon}[v], \sigma \text{ critical}\}$, then give the node $\max\{f(\sigma) : \sigma \in T_{c_i - \epsilon}[u], \sigma \text{ critical}\}$ the same direction (L or R) as that of n_{c_i} and give $\max\{f(\sigma) : \sigma \in T_{c_i - \epsilon}[u], \sigma \text{ critical}\}$ the opposite direction.

Continuing over all critical edges to obtain M. □

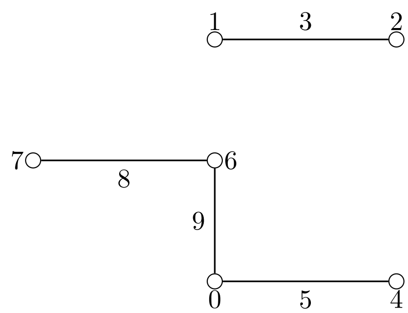
Definition 3.1. Two discrete Morse functions $f, g: T \to \mathbb{R}$ are **merge equivalent** if they induce the same unlabeled binary tree; that is, if there is there is a rooted graph isomorphism $\phi: M_f \to M_g$ such that node v has direction L if and only if node $\phi(v)$ has direction L.

It can be difficult to build the induced merge tree starting from the “top down" or the smallest value of the discrete Morse function since when adding new nodes to the merge tree, it is often unclear where a node is placed on the merge tree. This is because where it is placed depends on which component(s) it ends up merging to and when. Fortunately, Theorem 9 is starting from the “bottom up" or the largest value of the discrete Morse function. We give an example below.

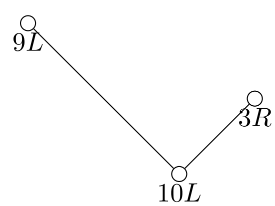
Example 10. To illustrate the construction of Theorem 1, we will take the discrete Morse function from Example 3. We begin by identifying the critical edge values and placing them in reverse order: $10, 9, 8, 5, 3$. The largest value is 10, so it corresponds to a node in M with label 10 and direction L:



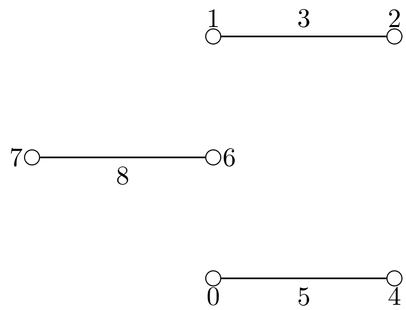
We then look at the level subcomplex $T_{10-\epsilon}$ and identify the largest value in each of the trees that were incident with 10.



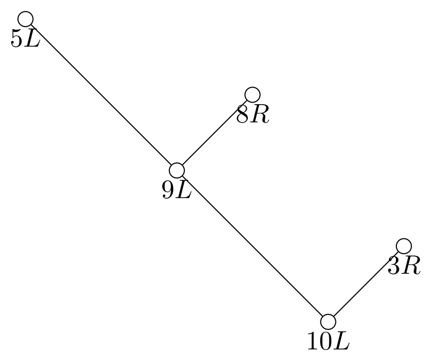
In this case, the two values are 3 and 9. To determine which is to the left and which is to the right, we look for the tree with the minimum value. In this case, $0 < 1$ so that 9 shares the same direction as 10. We thus obtain



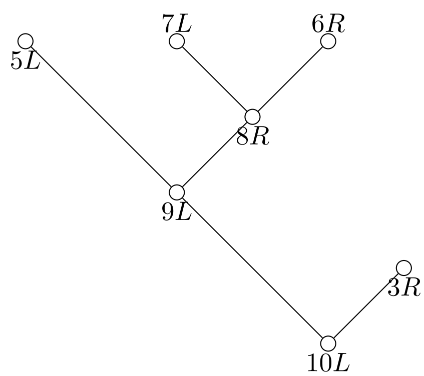
We move next to 9, and consider the level subcomplex $T_{9-\epsilon}$:



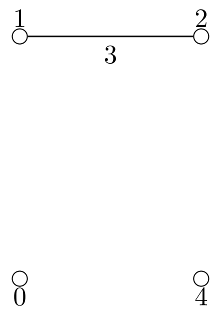
Now 9 was connected to 6 and 0, and the maximum value on each of their trees is 8 and 5, respectively, so these will be the labels of the two new nodes above 9. To see which one shares the direction with 9, we see that the tree with the vertex 0 has minimum value, so 5 shares the same direction as 9. We then obtain



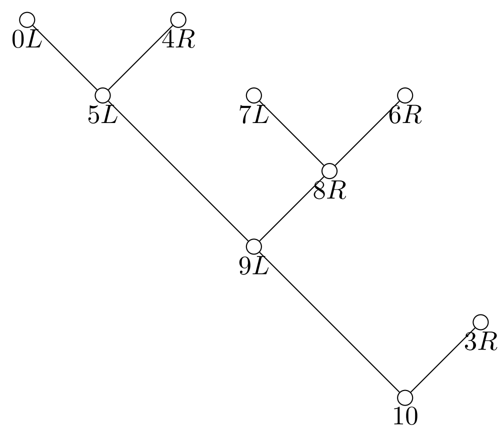
Now in $T_{8-\epsilon}$, 8 was connected to the isolated vertex 7 and isolated vertex 6. Hence the two new nodes connected to 8 will be 6 and 7. Since $6 < 7$, 6 and 8 share the same direction yielding



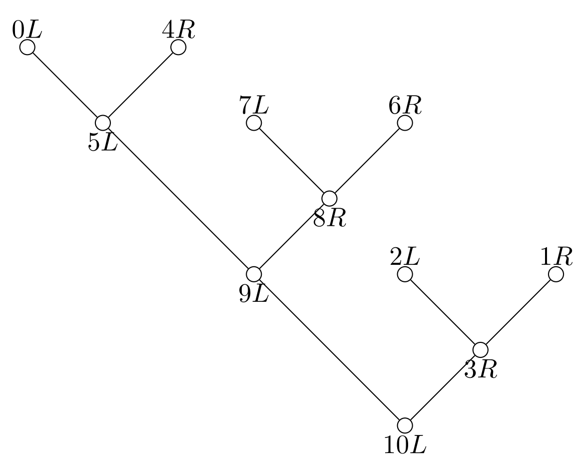
The next critical edge value is 5, so we consider the level subcomplex $T_{5-\epsilon}$:



The edge 5 was connected to isolated vertices 0 and 4, yielding



Finally, 3 is connected to 2 and 1, giving us the merge tree induced by the discrete Morse function:



3.2 Relation to Matching number

Recall that a **matching** in a graph is a set of edges such that no two edges share a common vertex. A matching is said to be **maximum** if it is a matching that contains the largest possible number of edges. The **matching number** of G, denoted $\nu(G)$, is the size of a maximum matching. We give a relationship between the matching number of a tree T and the induced merge tree of any discrete Morse function on T in Proposition 12. First a definition.

Definition 11. Let M be a merge tree. An internal node of M that is adjacent to exactly two leaves is called an **impasse**. The value $i(M)$ is the number of impasses of M.

Lemma 3.1. Every merge tree with more than one vertex has at least one impasse. That is, $i(M) \geq 1$ for every merge tree M.

Proof. Suppose we have a merge tree M without an impasse. Therefore, all internal nodes of M must have at least one internal node as a child. Consequently, each of those internal nodes must now have an internal node as a child. This continues on indefinitely, contradicting the fact that M is finite. □

Proposition 12. Let $f\colon T \to \mathbb{R}$ be discrete Morse function, M the induced merge tree of f. Then the set of edges of T corresponding to the set of impasses of M form a matching of T. In particular, $i(M) \leq \nu(T)$.

Proof. Let x, y be two impasses of M. In particular, x, y are not leaves and correspond to edges e_x, e_y, respectively, in T. We must show that e_x and e_y do not share a vertex. Suppose by contradiction that $e_x = uv$ and $e_y = uw$. Then the two leaves of x must be nodes corresponding to u and v, say n_u and n_v. Likewise for y. But then n_u is adjacent to both x and y, contradicting the fact that n_u is a leaf. Thus e_x and e_y do not share a vertex in common. It follows that the corresponding set of edges forms a matching, hence $i(M) \leq \nu(T)$. □

4 Comparison with other notions of equivalence

There are several other notions of equivalence of discrete Morse functions in the literature. In this section, we compare merge equivalence with these other notions.

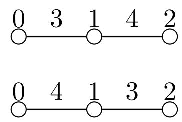
4.1 Forman equivalence

Definition 4.1. Let f be a discrete Morse function on G. The **induced gradient vector field** V_f is defined by

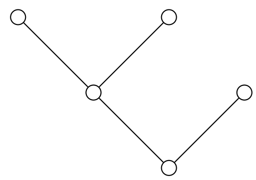
$$V_f := \{(\sigma^{(p)}, \tau^{(p+1)}) : \sigma < \tau, f(\sigma) \geq f(\tau)\}.$$

Recall that two discrete Morse functions $f, g\colon G \to [0, n]$ are **Forman equivalent** if and only if $V_f = V_g$ [3]. It is easy to see that neither Forman equivalence nor merge equivalence implies the other.

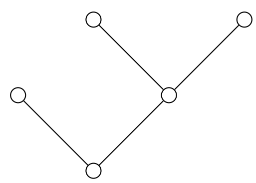
Example 13. Suppose we have the following discrete Morse functions:



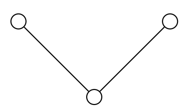
All simplices for both functions are critical, and hence the gradient vector field induced by both these functions has no arrows. Thus these functions are Forman equivalent. However, the merge trees are given by



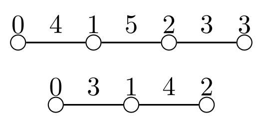
and



respectively. Thus they are not merge equivalent. On the other hand, consider the following merge tree



This merge tree is induced by both of the following discrete Morse functions:



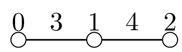
However, these functions are not Forman equivalent, as one has a regular pair and the other does not. Thus merge equivalence does not imply Forman equivalence.

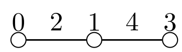
4.2 Homological equivalence

Given a graph with a discrete Morse function, one may study the Betti numbers of the level subcomplexes induced by the critical values. This gives rise to a non-negative sequence of integers. Such a sequence is a **homological sequence** and two discrete Morse functions are **homologically equivalent** if they induce the same homological sequence. See [2, 5, 4, 1].

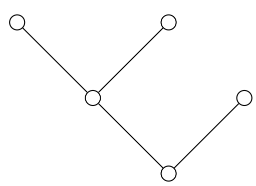
Example We show that homological equivalence and merge equivalence do not imply each other. First, consider the first two discrete Morse functions from Example 13. It is easy to see that they both induce the homological sequence $1, 2, 3, 2, 1$, hence they are homologically equivalent. However, their merge trees were shown in that same example to be different, hence they are not merge equivalent.

Now suppose we have the following functions





The homological sequence for these functions is given by $1, 2, 3, 2, 1$ and $1, 2, 1, 2, 1$, respectively so that they are not homologically equivalent. But they both induce the merge tree



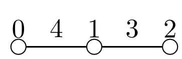
so that they are merge equivalent.

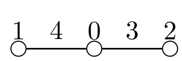
4.3 Persistence equivalence

Another notion of equivalence of discrete Morse functions, closely related to merge equivalence, is persistence equivalence. Two discrete Morse functions are **persistent equivalent** if they induce the same persistence diagram. See [16] for more details.

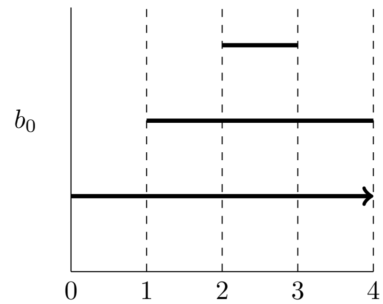
Example

Suppose we have the following discrete Morse functions:





These functions both create the same persistence diagram,



but different merge trees.

The same functions in Example 4.2 which show that merge equivalence does not imply homological equivalence also shows that merge equivalence does not imply persistence equivalence.

5 Merge tree of a star graph

Definition 14. Let $n \geq 2$ be an integer. The **star graph on n vertices** is defined by $S_n = K_{1,n-1}$ ([12, p. 17]). We call the unique vertex $c \in S_n$ of degree $n-1$ the **center of S_n** or **center vertex**.

We will characterize all merge trees induced by a critical excellent discrete Morse function on a star graph.

Definition 15. A merge tree M is called **thin** $i(M) = 1$, i.e, M has a unique impasse.

We can, in general, count the number of thin merge trees.

Proposition 16. There are exactly 2^{n-1} thin merge trees with n internal nodes.

Proof. We will count the number of ways to construct a thin merge tree with n internal nodes. Beginning with the very lowest node, we will choose which of its neighbors, left or right, is the internal node, forcing the other node to be a leaf, since by definition, a thin tree has only one internal node incident two leaves with all other internal nodes being incident to one leaf and one internal node. At each internal node, we choose the next internal node to be on the left or the right. This choice is made for every internal node except for the very last internal node, which ends with the node adjacent to two leaves. Since this choice is made for $n-1$ internal nodes, there are exactly 2^{n-1} thin merge trees with n internal nodes. □

The goal of the remainder of the section is to show that thin merge tree with $n+1$ leaves are in bijective correspondence with the merge equivalent discrete Morse functions on S_n. The next proposition tells us that the merge tree induced by a discrete Morse function on a star graph is always thin.

Proposition 17. If $T = S_n$ is a star graph and $f: S_n \to \mathbb{R}$ any discrete Morse function, then $i(M_f) = 1$.

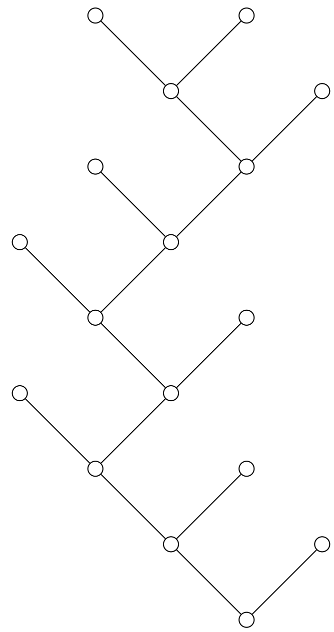
Proof. By Proposition 12, $i(M_f) \leq \nu(S_n) = 1$ since $\nu(S_n) = 1$. To show see that we have equality, apply Lemma 3.1. □

Proposition 18. Suppose M is a thin merge tree. Then there is a star graph S_n and a discrete Morse function $f: S_n \to M$ such that $M_f = M$.

Proof. Let M be a merge tree with $n+1$ leaves and choose $T = S_n$. We construct $f: S_n \to \mathbb{R}$ such that $M_f = M$. Consider the path from the root vertex r (unique vertex of degree 2) to the unique impasse of M. This path can be described by a sequence of L and R moves. Starting with an implicit L, let $n_0, \ldots, n_k$ be the nodes on the path that switch directions from L to R or R to L. Label the unique leaf of n_i with value i for $0 \leq i \leq k-1$ and either of the leafs of the unique impasse n_k with value k. Now choose k corresponding non-center vertices of S_n and label them $0, 1, \ldots, k-1$. Choose the center vertex of S_n to label k. Traversing the path from n_k to n_0, label each unlabeled leaf $k+1, k_2, \ldots, k+j$ and label the remaining vertices of S_n by $k+1, \ldots, k+j$. Finally, we know that each internal node corresponds to the edge in S_n. Traversing the path from n_k to n_0 again, label each internal node $k+j+1, k+j+2, \ldots, k+j+\ell$. The node n_k is labeled $k+j+1$ and has children labeled k and $k+1$. Hence label the edge in S_n connecting vertices k and $k+1$ with label $k+j+1$. Continuing along the path n_k to n_0, suppose we are at node n_{k-i} with label $k+j+i$, $o \geq 1$. Then n_{k-i} is adjacent to a leaf. This leaf corresponds to a vertex in S_n which is incident to a unique edge. Label this edge $k+j+i$. This completes the labeling of S_n. □

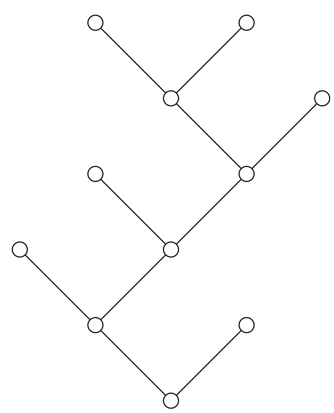
Corollary 19. For any star graph S_n, there are exactly 2^{n-1} possible merge trees induced by a critical discrete Morse function on S_n.

Remark 20. Given Proposition 16, we may specify a thin tree through a sequence of Ls and Rs. For example, the sequence LLRLRRL would correspond to the merge tree

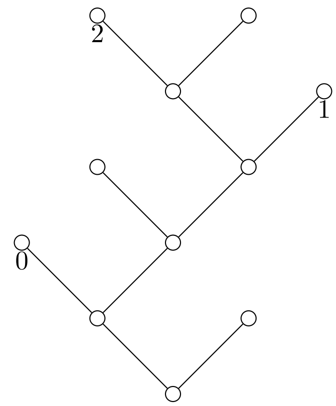


Notice that the sequence corresponds to a subpath of a path of maximum length in this merge tree.

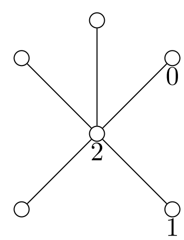
Example We give an example illustrating the construction of Proposition 18. We will find a star graph and discrete Morse function on that graph that induces the merge tree



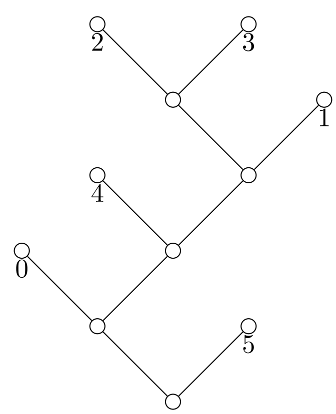
We first observe that because there are 6 leaves, we choose $T := S_5$. Beginning at the root vertex of M, we arrive at the impasse through the sequence of moves LRRL. At each switch from L to R or R to L, we label the corresponding leaf with with the next integer value yielding



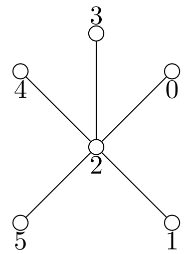
This corresponds to vertices in S_5, with the last value given to the center vertex



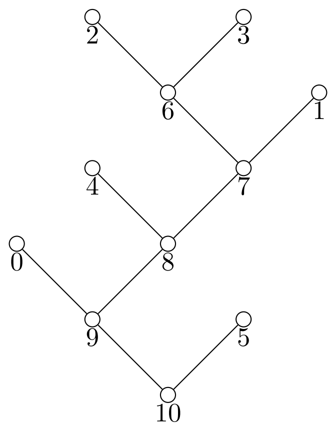
Now traverse the path in M from the impasse to the root, labeling the leaves $3, 4 \ldots$.



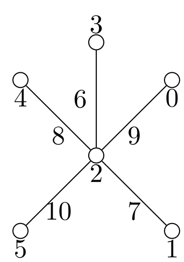
This corresponds to the same labels on the remaining leaves of S_5:



Traverse the same path in M, labeling the internal nodes $6, 7 \dots$



Finally, each edge of S_5 is labeled with the same label as the internal nodes of M. If edge $e \in S_5$ is incident with non-center vertex labeled a, then the leaf in M labeled a is incident with an interior node labeled b. Thus define the b to be the label of e so that



6 Conclusion and future work

Conjecture: $\exists$ discrete Morse functions on G that induce any given merge tree if and only if G is a path.

Conjecture: If $\exists v \in G$ such that $deg(v) > 2$, then $\exists$ a merge tree that cannot be realized by any discrete Morse function on G.

References

[1] M. Agiorgousis, B. Green, A. Onderdonk, N. A. Scoville, and K. Rich, *Homological sequences in discrete Morse theory*, Topology Proc. **54** (2019), 283–294.

[2] R. Ayala, L. M. Fernández, D. Fernández-Ternero, and J. A. Vilches, *Discrete Morse theory on graphs*, Topology Appl. **156** (2009), no. 18, 3091–3100.

[3] R. Ayala, L. M. Fernández, and J. A. Vilches, *Characterizing equivalent discrete Morse functions*, Bull. Braz. Math. Soc. (N.S.) **40** (2009), no. 2, 225–235.

[4] R. Ayala, D. Fernández-Ternero, and J. A. Vilches, *Counting excellent discrete Morse functions on compact orientable surfaces*, Image-A : Applicable Mathematics in Image Engineering **1** (2010), no. 1, 49–56.

[5] R. Ayala, D. Fernández-Ternero, and J. A. Vilches, *The number of excellent discrete Morse functions on graphs*, Discrete Appl. Math. **159** (2011), no. 16, 1676–1688.

[6] A. J. Blumberg, I. Gal, M. A. Mandell, and M. Pancia, *Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces*, Found. Comput. Math. **14** (2014), no. 4, 745–789.

[7] G. Carlsson, *Topology and data*, Bull. Amer. Math. Soc. (N.S.) **46** (2009), no. 2, 255–308.

[8] G. Chartrand, L. Lesniak, and P. Zhang, *Graphs & digraphs*, sixth ed., Textbooks in Mathematics, CRC Press, Boca Raton, FL, 2016.

[9] J. Curry, *The fiber of the persistence map for functions on the interval*, J. Appl. Comput. Topol. **2** (2018), no. 3-4, 301–321.

[10] D. L. Ferrario and R. A. Piccinini, *Simplicial structures in topology*, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, Springer, New York, 2011, Translated from the 2009 Italian original by Maria Nair Piccinini.

[11] R. Forman, *Discrete Morse theory and the cohomology ring*, Trans. Amer. Math. Soc. **354** (2002), no. 12, 5063–5085.

[12] F. Harary, *Graph theory*, Addison-Wesley Publishing Co., Reading, Mass.-Menlo Park, Calif.-London, 1969.

[13] K. Knudson, *Morse theory: Smooth and discrete*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015.

[14] D. N. Kozlov, *Combinatorial algebraic topology*, Algorithms and Computation in Mathematics, vol. 21, Springer, Berlin, 2008.

[15] A. G. Landge, P. Bremer, A. Gyulassy, and Valerio Pascucci, *Notes on the distributed computation of merge trees on CW-complexes*, Topological methods in data analysis and visualization. IV, Math. Vis., Springer, Cham, 2017, pp. 333–348.

[16] Y. Liu and N. A. Scoville, *The realization problem for discrete Morse functions on trees*, Algebra Colloquium (to appear).

[17] P. Oesterling, C. Heine, G. H. Weber, D. Morozov, and G. Scheuermann, *Computing and visualizing time-varying merge trees for high-dimensional data*, Topological methods in data analysis and visualization. IV, Math. Vis., Springer, Cham, 2017, pp. 87–101.

[18] N. A. Scoville and K. Yegnesh, *A Persistent Homological Analysis of Network Data Flow Malfunctions*, Journal of Complex Networks **6** (December 2017), no. 1, 884–892.

[19] G. Spreemann, B. Dunn, M. B. Botnan, and N. A. Baas, *Using persistent homology to reveal hidden covariates in systems governed by the kinetic Ising model*, Phys. Rev. E **97** (2018), no. 3, 032313, 14.