

# PERSISTENT HOMOLOGY FOR SPEECH RECOGNITION

KEVIN P. KNUDSON

ABSTRACT. We use the persistent homology of Edelsbrunner, Letscher, and Zomorodian to associate to a signal an invariant called a barcode which consists of a set of intervals of real numbers. The collection of all barcodes may be equipped with a metric, allowing us to calculate the distances between these objects. This provides a topological mechanism to distinguish signals. We present the results of experiments on human speech signals where the barcodes associated to eight different sounds are calculated and the distances among them computed. The resulting point clouds in “barcode space” are studied; often such a cloud has eight components, one for each sound. This technique may therefore be useful in speech recognition problems.

## 1. INTRODUCTION

As computing capabilities have improved, it has become more practical to analyze large data sets. A relatively new, and increasingly popular, tool for studying the structure of large point clouds is *persistent homology* [8],[11]. Here is one possible scenario. Suppose one has a collection  $X$  of points in some metric space (usually a Euclidean space, but not necessarily). Let  $\varepsilon > 0$  and construct a space  $R(X, \varepsilon)$  by placing a ball of radius  $\varepsilon$  around each point in  $X$ . If  $\varepsilon$  is small, then  $R(X, \varepsilon)$  is simply a disjoint collection of balls, but as  $\varepsilon$  grows, the space  $R(X, \varepsilon)$  may have more interesting topology. Also, if  $\varepsilon < \varepsilon'$ , there is an inclusion  $R(X, \varepsilon) \rightarrow R(X, \varepsilon')$ . One may then look for geometric features in the spaces  $R(X, \varepsilon)$ . Algebraic topology is the mathematical formalism developed to quantify such features. In particular, the *homology groups* of the spaces  $R(X, \varepsilon)$  are vector spaces which give a measure of the number of distinct features in the space; such objects are called *homology classes*. For each  $i \geq 0$  and a field  $k$ , one then gets a direct system of homology groups  $\{H_i(R(X, \varepsilon); k)\}_{\varepsilon > 0}$  measuring the  $i$ -dimensional geometric classes. Some homology classes live over only a small interval of  $\varepsilon$ -values, while others persist for long periods. The latter are more likely to correspond to real homological features in  $X$ ; we are therefore interested in measuring these persistent classes as a means to find structure in point cloud data sets. The persistence of the homology classes is neatly encapsulated in a structure called a *barcode*, which is a collection of intervals measuring the

---

2000 *Mathematics Subject Classification.* 55N99, 55U10.

*Key words and phrases.* persistent homology, metric space, Betti numbers, connected components.

longevity of the various classes. This approach has been used to study the structure of data sets arising in shape recognition [4], natural images [3], speech recognition [1], and wheeze detection [9].

This technique of growing  $\varepsilon$ -balls is difficult to automate, however, so we model the space underlying the data by an object called a *simplicial complex* (see Section 2 below). These are easily stored in a computer, and the linear algebra calculations required for computing their homology can be carried out efficiently. Moreover, it is not difficult to construct a whole family of complexes to model the data, in a way similar to growing  $\varepsilon$ -balls, so that we may compute the persistent homology of the family and use the associated barcode to infer features in the data.

In this paper, we apply this technique to the following problem. We consider a collection of human speech signals, eight different phones consisting of three vowels, two nasal sounds, and three “noisy” sounds. Ideally, we would like to be able to distinguish these signals in some fashion. The approach we take is to consider a short segment of the signal, view the graph as a one-dimensional simplicial complex, define an interesting nested collection of subcomplexes, and then compute the zero-dimensional persistent homology of this family. This gives a means to associate to a signal a barcode. For each sound, we do this procedure on 100 different segments; the result is a collection of 800 barcodes. In turn, these barcodes are points in a space, called *barcode space*, which is equipped with a distance function. We then study the topology of this point cloud in barcode space and show that it often has eight components, one for each sound. This leads to a possible algorithm for speech recognition.

This paper is organized as follows. Section 2 contains a brief introduction to simplicial complexes and their homology. In Section 3 we define persistent homology and the metric on barcode space. We discuss the procedure for associating a barcode to a signal in Section 4 and give examples of this for human speech signals in Section 5. This section also includes an analysis of the structure of the resulting point clouds in barcode space.

## 2. SIMPLICIAL HOMOLOGY

An *abstract simplicial complex*  $K$  is specified by the following data:

- A vertex set  $V$ ;
- A rule specifying when a  $p$ -simplex  $\sigma = [v_0 v_1 \dots v_p]$  belongs to  $K$ ; here the vertices  $v_0, v_1, \dots, v_p$  are distinct elements of  $V$ ;
- Each  $p$ -simplex  $\sigma$  has  $p + 1$  faces which are the  $(p - 1)$ -simplices obtained by deleting one of the vertices of  $\sigma$ . The membership rule has the property that if  $\sigma$  belongs to  $K$ , then all of its faces belong to  $K$ .

Given a simplicial complex  $K$ , we define a collection of vector spaces, which tell us the number of holes of various dimensions in  $K$ , as follows. Let  $k$  be the field of 2 elements. The  $i$ -th homology group,  $H_i(K; k)$ , will

measure the number of  $(i + 1)$ -dimensional voids, and it is constructed as follows. Let  $C_i(K; k)$  be the  $k$ -vector space with basis the set of  $i$ -simplices in  $K$ . If  $\sigma = [v_0 v_1 \cdots v_i]$  is such a simplex, we define  $\partial\sigma$  to be the element of  $C_{i-1}(K; k)$  given by the formula

$$\partial\sigma = \sum_{j=0}^i (-1)^j [v_0 v_1 \cdots \hat{v}_j \cdots v_i],$$

where  $[v_0 v_1 \cdots \hat{v}_j \cdots v_i]$  is the  $(i - 1)$ -simplex with vertices  $\{v_0, \dots, v_i\} - \{v_j\}$ . Note that in the field of 2 elements we have  $-1 = 1$ , but we present the definition this way because it works over any field (e.g., the real numbers). We may extend this linearly to  $C_i(K; k)$  to obtain a linear transformation

$$\partial^i : C_i(K; k) \rightarrow C_{i-1}(K; k).$$

It is a straightforward exercise to show that  $\partial^i \circ \partial^{i+1} = 0$  and hence  $\text{im}(\partial^{i+1}) \subseteq \text{null}(\partial^i)$  ( $\text{null}(\partial^i)$  denotes the null space of the map  $\partial^i$ ). We then define the  $i$ -th homology group as

$$H_i(K; k) = \text{null}(\partial^i) / \text{im}(\partial^{i+1}).$$

Elements of  $\text{null}(\partial^i)$  are called *cycles*; the set of all such is denoted by  $Z_i$ . Elements of  $\text{im}(\partial^{i+1})$  are called *boundaries*, denoted by  $B_i$ . Homology measures how many cycles are inequivalent and essential in the sense that they do not bound an object of higher dimension.

For computational purposes, we are interested in the *Betti numbers*,  $\beta_i$ , defined as  $\beta_i = \dim_k H_i(K; k)$ . Note that we have the simple equation

$$\begin{aligned} \beta_i &= \dim_k Z_i - \dim_k B_{i+1} \\ &= \dim_k C_i - \text{rank } \partial^i - \text{rank } \partial^{i+1}. \end{aligned}$$

This therefore reduces the calculation of Betti numbers to computing ranks of matrices over the field  $k$ .

As a simple example, consider the tetrahedron  $T$ . There are four vertices  $v_0, v_1, v_2, v_3$ ; six edges  $[v_0 v_1], [v_0 v_2], [v_0 v_3], [v_1 v_2], [v_1 v_3], [v_2 v_3]$ ; and four faces  $[v_0 v_1 v_2], [v_0 v_1 v_3], [v_0 v_2 v_3], [v_1 v_2 v_3]$ . The groups  $C_i(T; k)$  are then

$$C_0(T; k) = k^4 \quad C_1(T; k) = k^6 \quad C_2(T; k) = k^4$$

and the maps  $\partial^i$  are given by  $\partial^0 = 0$ ,  $\partial^i = 0, i \geq 3$ , and

$$\partial^1 = \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\partial^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

An easy calculation shows that  $\text{rank } \partial^1 = 3$ ,  $\text{rank } \partial^2 = 3$ , and therefore that

$$\begin{aligned} H_0(T; k) = k &\Rightarrow \beta_0 = 1 \\ H_1(T; k) = 0 &\Rightarrow \beta_1 = 0 \\ H_2(T; k) = k &\Rightarrow \beta_2 = 1. \end{aligned}$$

A basis for  $H_0(T; k)$  is  $[v_0]$ , and for  $H_2(T; k)$ ,  $[v_0v_1v_2] - [v_0v_1v_3] + [v_0v_2v_3] - [v_1v_2v_3]$ . There are three linearly independent 1-cycles—the boundaries of the four triangles form a linearly dependent set of dimension 3—but each is also a boundary, filled in by the interior of the triangle. Geometrically, the fact that  $\beta_0 = 1$  means that  $T$  is connected;  $\beta_1 = 0$  means that every loop in  $T$  bounds a disc;  $\beta_2 = 1$  means that  $T$  contains a closed surface, namely  $T$  itself, that is not filled in by a 3-dimensional object.

Homology groups are topological invariants; that is, if spaces  $X$  and  $Y$  are homotopy equivalent (one may be deformed to the other), then  $H_\bullet(X; k) \cong H_\bullet(Y; k)$ . They therefore provide a means to distinguish spaces, although it is possible for topologically distinct spaces to have the same homology groups.

### 3. FILTRATIONS AND BARCODES

An increasingly popular technique for analyzing data sets topologically is the *persistent homology* of Edelsbrunner, Letscher, and Zomorodian [8]. The idea is as follows. Suppose we are given a finite nested sequence of finite simplicial complexes

$$K_{R_1} \subset K_{R_2} \subset \cdots \subset K_{R_p},$$

where the  $R_i$  are real numbers  $R_1 < R_2 < \cdots < R_p$ . For each homological degree  $\ell \geq 0$ , we then obtain a sequence of homology groups and induced linear transformations

$$H_\ell(K_{R_1}) \rightarrow H_\ell(K_{R_2}) \rightarrow \cdots \rightarrow H_\ell(K_{R_p}).$$

Since the complexes are finite, each  $H_\ell(K_{R_i})$  is a finite-dimensional vector space. Thus, there are only finitely many distinct homology classes. A particular class  $z$  may come into existence in  $H_\ell(K_{R_s})$ , and then one of two things happens. Either  $z$  maps to 0 (i.e., the cycle representing  $z$  gets filled in) in some  $H_\ell(K_{R_t})$ ,  $R_s < R_t$ , or  $z$  maps to a nontrivial element in  $H_\ell(K_{R_p})$ . This yields a *barcode*, a collection of interval graphs lying above an axis parametrized by  $R$ . An interval of the form  $[R_s, R_t]$  corresponds to a class that appears at  $R_s$  and dies at  $R_t$ . Classes that live to  $K_{R_p}$  are usually

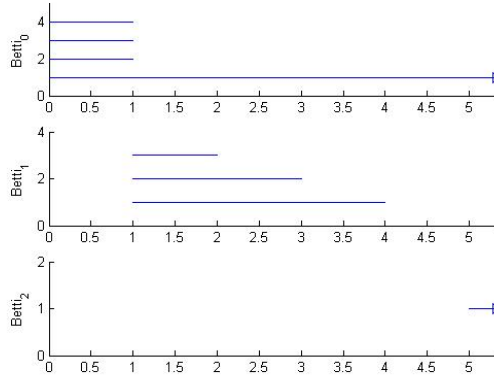


FIGURE 1. The barcodes for a filtration of the tetrahedron.

represented by the infinite interval  $[R_s, \infty)$  to indicate that such classes are real features of the full complex  $K_{R_p}$ .

As an example, consider the tetrahedron  $T$  with filtration

$$T_0 \subset T_1 \subset T_2 \subset T_3 \subset T_4 \subset T_5 = T$$

defined by  $T_0 = \{v_0, v_1, v_2, v_3\}$ ,  $T_1 = T_0 \cup \{\text{all edges}\}$ ,  $T_2 = T_1 \cup [v_0v_1v_2]$ ,  $T_3 = T_2 \cup [v_0v_1v_3]$ ,  $T_4 = T_3 \cup [v_0v_2v_3]$ , and  $T_5 = T$ . The barcodes for this filtration are shown in Figure 1. Note that initially, there are 4 components ( $\beta_0 = 4$ ), which get connected in  $T_1$ , when 3 independent 1-cycles are born ( $\beta_1 = 3$ ). These three 1-cycles die successively as triangles get added in  $T_2$ ,  $T_3$ , and  $T_4$ . The addition of the final triangle in  $T_5$  creates a 2-cycle ( $\beta_2 = 1$ ). We see that the classes that live forever yield the Betti numbers calculated in the previous section.

For analyzing point cloud data, one needs a simplicial complex modeling the underlying space. Since it is impossible to know *a priori* if a complex is “correct”, one builds a nested family of complexes approximating the data cloud, computes the persistent homology of the resulting filtration, and looks for homology classes that exist in long sections of the filtration. Since we will only be interested in computing the number of components of the cloud of barcodes we construct, we will be able to use traditional clustering techniques. For a general discussion of constructions of complexes used to analyze data (in particular the popular *witness complex* construction), see [7],[5].

**3.1. Barcode space.** The collection  $\mathcal{B}$  of all barcodes itself forms a space which we call *barcode space*. This is a metric space where the distance function  $d : \mathcal{B} \times \mathcal{B} \rightarrow \mathbf{R}$  is defined as follows. For a pair of intervals  $I, J$ , we define their dissimilarity  $\delta(I, J)$  to be the measure of their symmetric difference:  $\delta(I, J) = \mu(I \cup J - I \cap J)$ , where  $\mu$  denotes one-dimensional measure. Note that  $\delta(I, J)$  may be infinite. Given barcodes  $S_1, S_2 \in \mathcal{B}$ ,

a *matching* is a set  $M(S_1, S_2) \subseteq S_1 \times S_2$  so that any interval in  $S_1$  or  $S_2$  occurs in at most one pair  $(I, J)$ . Denote the intervals in  $S_1$  and  $S_2$  that are matched in  $M$  by  $M_1$  and  $M_2$ , respectively. Let  $N$  be the non-matched intervals. For a matching  $M$ , define the distance between  $S_1$  and  $S_2$  relative to  $M$  to be the sum

$$D_M(S_1, S_2) = \sum_{(I, J) \in M} \delta(I, J) + \sum_{L \in N} \mu(L).$$

The (pseudo)-metric  $d$  is then defined by

$$d(S_1, S_2) = \min_M D_M(S_1, S_2),$$

where the minimum is taken over all matchings of  $S_1$  and  $S_2$ .

At first glance, this definition may seem computational infeasible. However, we may convert it to another question that has fast solutions. Note that  $\delta(I, J) = \mu(I) + \mu(J) - 2\mu(I \cap J)$ . Define the similarity of  $S_1$  and  $S_2$  with respect to  $M$  to be

$$\begin{aligned} S_M(S_1, S_2) &= \sum_{(I, J) \in M} \mu(I \cap J) \\ &= \frac{1}{2} \left( \sum_{S_1} \mu(I) + \sum_{S_2} \mu(J) - D_M(S_1, S_2) \right). \end{aligned}$$

The problem of minimizing  $D_M$  is therefore equivalent to maximizing  $S_M$  and this may be recast as a question in graph theory. Given barcodes  $S_1$  and  $S_2$ , define a weighted bipartite graph  $G(V, E)$  with vertex set  $V = S_1 \cup S_2$ . Place an edge in  $E$  for each pair  $(I, J) \in S_1 \times S_2$  with weight  $\mu(I \cap J)$ . Maximizing  $S_M$  is then equivalent to finding a maximal weight matching in this graph. There are efficient algorithms for achieving this (e.g. the Hungarian algorithm [10] with complexity  $O(|V||E|)$ ).

#### 4. BARCODES FOR SIGNALS VIA SWEEPING

Suppose  $f : I \rightarrow \mathbf{R}$  is a continuous real-valued function defined on the closed interval  $I$ . We may then define an increasing filtration  $\{I_\alpha\}_{\alpha \in \mathbf{R}}$  of  $I$  by setting  $I_\alpha = f^{-1}((-\infty, \alpha])$ , as  $\alpha$  varies over  $\mathbf{R}$ . A useful visualization of this is to imagine a horizontal line in the plane moving upward across the graph of  $f$ . As  $\alpha$  increases, the sets  $I_\alpha$  get larger until eventually the entire graph is swept out. We call this the *sweep filtration* of the interval  $I$ .

Since  $I$  is a contractible one-dimensional space, the only interesting homology groups associated to the sets  $I_\alpha$  are the various  $H_0(I_\alpha)$ . Call such a function *tame* if it has only a finite number of local maxima and minima. Observe that when the parameter  $\alpha$  passes a local minimum a new component is added and  $\beta_0$  increases by 1. When a local maximum is passed, two components merge and  $\beta_0$  decreases by 1. What we are really doing here is pairing up local minima and maxima as follows: when a local maximum is reached, it is paired with the local minimum which entered the filtration

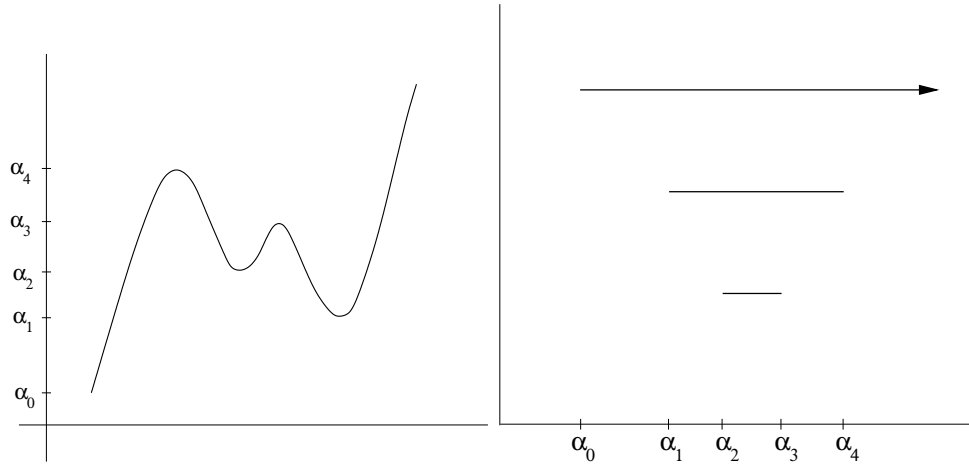


FIGURE 2. A tame function  $f$  and the barcode associated to the sweep filtration.

most recently and is still unpaired. Those local extrema which seem to give the true shape of the graph persist for long periods while those that come from small perturbations persist for short intervals. See Figure 2 for an example.

Now suppose we are given a signal as a time series. Let  $f(t)$  denote the amplitude of the wave at time  $t$ . Since we are given a discrete set of points  $(t_i, f(t_i))$ ,  $0 \leq i \leq n$ , we may view this as a map  $f : K \rightarrow \mathbf{R}$  where  $K$  is the one-dimensional simplicial complex with a vertex  $v_i$ ,  $0 \leq i \leq n$  and an edge joining  $v_i$  with  $v_{i+1}$ . This is simply a triangulation of an interval  $I$ . The function  $f$  is then given as the piecewise linear map interpolating between the values  $f(t_i)$ . If  $\alpha$  is a real number, set  $K_\alpha$  to be the subcomplex of  $K$  generated by  $f^{-1}((-\infty, \alpha])$ . See Figure 3 for an example.

We therefore have a mechanism for assigning a barcode to a signal. Viewing these as points in the barcode space  $\mathcal{B}$ , we may then calculate the distance between barcodes and hopefully distinguish between signals in this way.

Note that the metric  $d$  is unstable in the following sense. Suppose we have a signal  $f$  and its barcode  $B_f$ . A small perturbation  $f^*$  will yield a new barcode  $B_{f^*}$  with many very short bars. One would like to know that  $d(B_f, B_{f^*})$  is small, but it is clear that it is possible to take a very small perturbation  $f^*$  so that the number of extra bars in  $B_{f^*}$  is very large. These new intervals will be unmatched in calculating  $d(B_f, B_{f^*})$ , and even though they are all very short, their lengths could add up to be very large.

There are other metrics available, for example the bottleneck metric of [6]. This is a stable metric in the sense that functions that are close have barcodes that are close in this metric. We have performed the calculations of Section 5 below with this metric, but the results are not as good. That

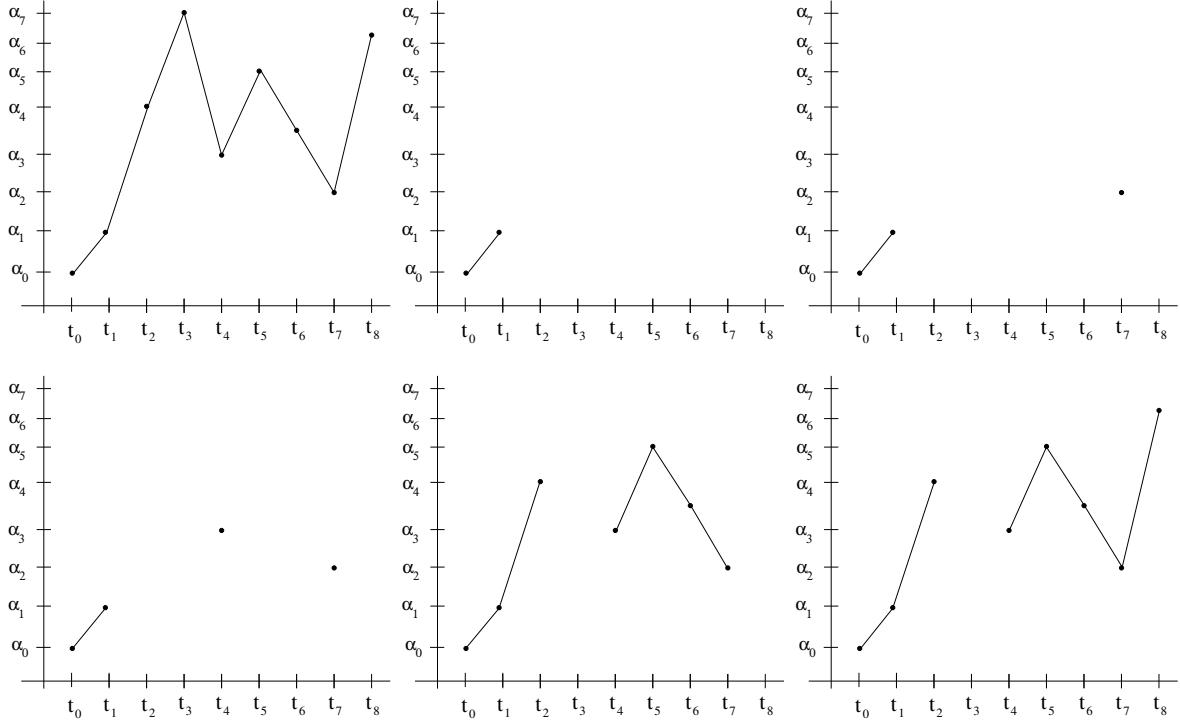


FIGURE 3. A signal as a time series and the associated subcomplexes  $K_{\alpha}$ ; the top row has the signal,  $K_{\alpha_1}$ , and  $K_{\alpha_2}$ , while the bottom row has  $K_{\alpha_3}$ ,  $K_{\alpha_5}$ , and  $K_{\alpha_6}$ . The values of  $\beta_0$  are 1, 1, 2, 3, 2, 2, respectively.

is, we obtained sharper clustering with the metric  $d$ . The reasons for this are unclear, but we will present an analysis in [2].

## 5. POINT CLOUDS IN BARCODE SPACE FROM HUMAN SPEECH SIGNALS

We obtained a collection of speech signals from three individuals, labeled by their initials: *cm*, *if*, *mc*. There were eight different sounds: three vowels (*aa* as in the Spanish word *casa*, *ae* as in *safe*, and *eh* as in *bed*), two nasals (*m* and *n*), and three “noisy” sounds (*f*, *sh*, and *z*). The sample rate was 22050 samples per second. For each sound, we selected 100 segments. For the periodic sounds (vowels and nasals) each segment consisted of a single period (approximately 130 sample points) and for the noisy sounds we selected segments of length equal to the average of the lengths of the periodic sounds. Examples of the segments for speaker *cm* are shown in Figure 4. For each segment, we calculated the  $\beta_0$ -barcode obtained by filtering the signal by the sweeping procedure described in Section 4.

For each speaker, this process yielded a point cloud in barcode space consisting of 800 points. We then computed the  $800 \times 800$  matrix of distances between pairs of these points. This was computationally expensive—speaker



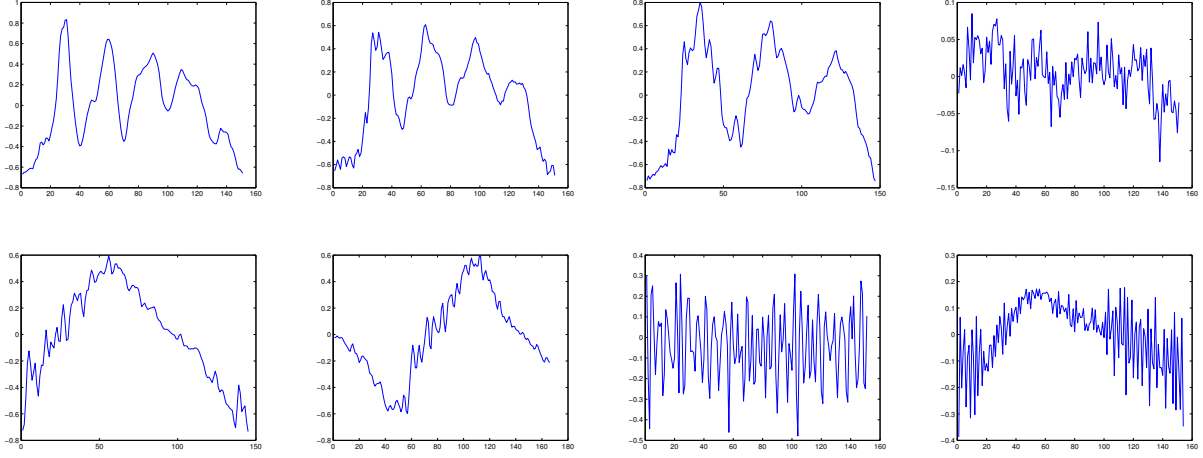


FIGURE 4. Segments of the signals from speaker *cm*. Top row: *aa*, *ae*, *eh*, *f*; Bottom row: *m*, *n*, *sh*, *z*.

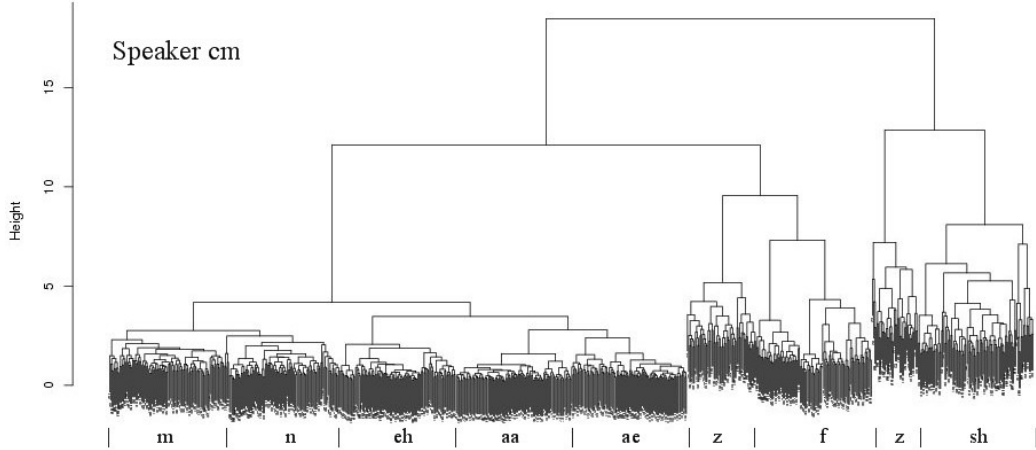
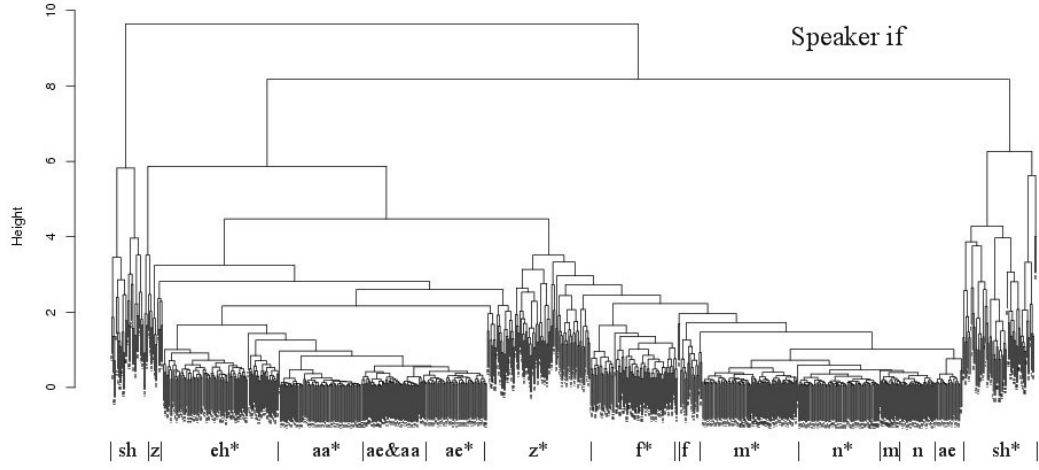
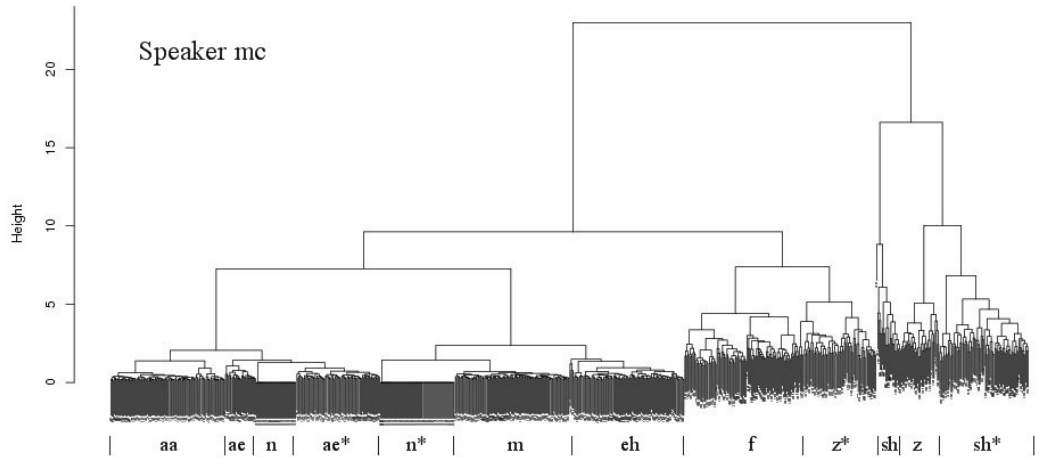


FIGURE 5. Clustering dendrogram for speaker *cm*.

*mc*'s matrix required approximately 36 hours of CPU time. We note, however, that the calculation was done in MATLAB on a laptop running Windows XP. Calculations in [2], done in C++, are significantly faster. The experiment is then to study the structure of this point cloud in  $\mathcal{B}$ . One would hope that it consists of eight components, one for each sound.

**5.1. The components of speech data point clouds.** To analyze the point clouds in  $\mathcal{B}$ , we use standard hierarchical clustering. The associated dendrograms are shown in Figures 5, 6, and 7 for speakers *cm*, *if*, and *mc*, respectively.

FIGURE 6. Clustering dendrogram for speaker *if*.FIGURE 7. Clustering dendrogram for speaker *mc*.

Let us examine these dendrograms. For speaker *cm*, we have 9 very distinct components, with sound *z* breaking into 2 clusters. The larger of these has 60 points in it. Note, however, that the nasal sounds (*m* and *n*) are quite close together, as are the three vowel clusters. Furthermore, the clusters for *f*, *sh*, and *z* also join together before merging with other components. So we see a sharp divide between the three classes of sounds, and within each class, for speaker *cm* at least, definite clustering occurs for each sound.

Consider next speaker *if*. Those clusters labeled with an asterisk are the largest for a particular sound. Note again that the three classes of sounds

	aa	ae	eh	f	m	n	sh	z
<i>cm</i>	.01	.32	.93	.90	.52	.27	.48	.81
<i>if</i>	.98	.98	.63	.86	.51	0	.47	.83
<i>mc</i>	.99	.71	.76	.32	.30	.82	.68	.83

TABLE 1. Error rates for the proposed recognition algorithm.

tend to bunch together, and inside each of these, individual sounds form large clusters of their own. Observe also that sound *sh* forms two rather large clusters away from the other sounds.

Finally, we study speaker *mc*. Again, we see that the noisy sounds tend to cluster together, with each sound forming its own component (more or less). The nasals tend to cluster near each other, as do *aa* and *ae*. The component for *eh*, however, is closer to the nasals, and merges with them before the other vowels.

In all three cases, though, we see clear evidence of clustering among the individual sounds.

**5.2. A proposed algorithm.** The existence of 8 large components in the point clouds suggests the following procedure for speech recognition.

- (1) Given a signal, select a short sample (one period of a vowel, for example).
- (2) Compute the barcode  $S$  associated to the sweep filtration of the signal.
- (3) Choose a representative barcode from the largest cluster in  $\mathcal{B}$  for each sound and calculate the distance from  $S$  to each of these.
- (4) The barcode closest to  $S$  is likely the same sound.

We ran experiments to test the error rate for this procedure. For each speaker and each sound, we selected a representative at random from each large cluster. Given a sound, we selected 100 samples and computed the distances from each sample to each of the representative signals. The results are shown in Table 1.

Of course, this is discouraging news. The best results were for sound *aa* for speaker *cm* and sound *n* for speaker *if*. Otherwise, the algorithm almost always fails. The good news is that when a sound is incorrectly identified, it is usually mistaken for a sound in its same class; that is, a vowel is close to another vowel, nasals may be mistaken for each other, and noisy sounds may be garbled.

One remedy for this is to choose more representatives from each cluster so that a particular signal has a chance to be close to a correct sound. The downside of this is that more storage space is required, and the computation time to identify a sound will increase. Perhaps a more judicious choice of representative barcodes would provide a fix. And, while we assert that the metric  $d$  provides more recognizable clusters, it may be better to use another metric (such as the bottleneck distance) to test the algorithm.

## ACKNOWLEDGMENTS

The author is grateful to Sundar Srinivasan for providing the sound files and for helpful conversations. Kenny Brown provided valuable assistance with coding. The persistent homology calculations were performed in MATLAB using Vin de Silva and Patrick Perry's Plex package, which includes the persistent homology package written by Afra Zomorodian and Lutz Kettner. Plex is available for download at <http://comptop.stanford.edu/>. The clustering calculations were performed in the statistics package R; thanks go to Russ Stocker for R-help.

## REFERENCES

- [1] K. Brown and K. Knudson, *Nonlinear statistics of human speech data*, Internat. J. Bifur. Chaos Appl. Sci. Engrg. **19** (2009), 2307–2319.
- [2] K. Knudson, *Metrics in barcode space*, in preparation.
- [3] G. Carlsson, T. Ishnakov, V. de Silva, and A. Zomorodian, *On the local behavior of spaces of natural images*, International Journal of Computer Vision **76** (2008), 1–12.
- [4] G. Carlsson, A. Zomorodian, A. Collins, and L. J. Guibas, *Persistence barcodes for shapes*, International Journal of Shape Modeling **11** (2005), 149–187.
- [5] F. Chazal and S. Oudot, *Towards persistence-based reconstruction in Euclidean spaces*, Proc. 24th ACM Sympos. Comput. Geom. (2008), 232–241.
- [6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, *Stability of persistence diagrams*, Proc. 21st Ann. Sympos. Comput. Geom. (2005), 263–271.
- [7] V. de Silva and G. Carlsson, *Topological estimation using witness complexes*, in Eurographics Symposium on Point-Based Graphics (2004).
- [8] H. Edelsbrunner, D. Letscher, and A. Zomorodian, *Topological persistence and simplification*, Disc. Comp. Geom. **28** (2002), 511–533.
- [9] S. Emrani, T. Gentimis, and H. Krim, *Persistent homology of delay embeddings*, preprint (2013), arXiv:1305.3879.
- [10] H. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly **2** (1955), 83–97.
- [11] A. Zomorodian and G. Carlsson, *Computing persistent homology*, Disc. Comp. Geom. **33** (2005), 249–274.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF FLORIDA, GAINESVILLE, FL 32611.  
*E-mail address:* [kknudson@ufl.edu](mailto:kknudson@ufl.edu)