Segment Assignment Methods

Readme

This document is culled from a variety of sources used in our Phase I and Phase II work, as well as internal reports and funding applications. As such, figure numbers and references to, e.g., Phase I applications might not make sense. Nevertheless, this is a good starting point to read about other ideas for segment connection and assignment that we have tested but not implemented in SKATE.

Note that SKATE currently uses a single segment assignment method based on geometric distance. This is detailed in the document called 'Phase II final report.'

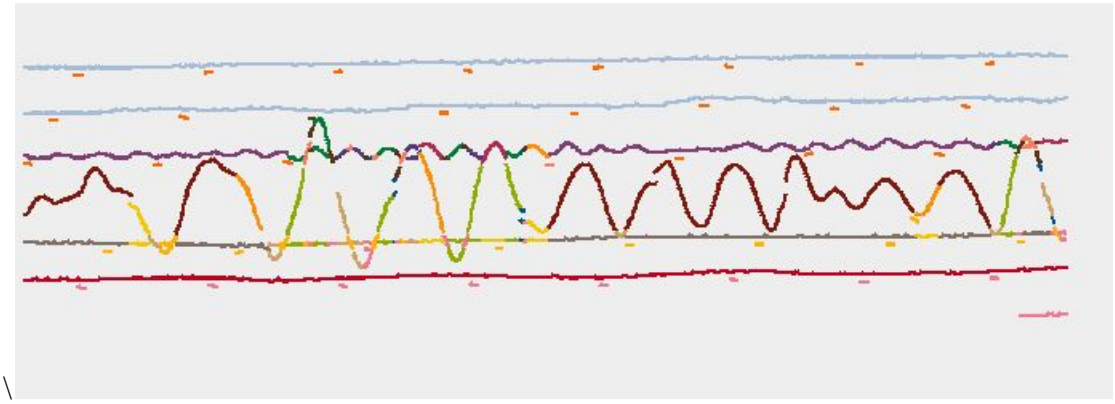
Approaches Based on Distance and Clustering

Below we describe some tests which involve assigning a vector of real number to each segment. Within each test the vectors all have the same number of elements and they are used to define a distance (metric) which measures how close to segments are to each other. A clustering algorithm (Frey Dueck) is then used to find sets of segments which are close to each other in either the frequency domain or the time domain. More precisely the elements of each cluster are closer to each other than they are to the segments not in the same cluster.

The tests described below were all applied to a dataset which had light to moderate seismic activity.

Test 0 Time domain data is used. For each segment a 64 element vector is constructed. If the segment contains less than 64 points, say n-points, the first n-points of the vector are assigned the values of the corresponding values of the segment and the remainder are "zero-filled", that is set to 0.0. If the segment has more than or equal to 64 points, vector elements 1 through 32 are set equal to the first 32 values of the segment and elements 33 through 64 are set equal to the last 32 values of the segment. Note that a long segment will have values that are not represented in the vector. For example values from point 33 to 68 will not occur in the associated vector of a segment of length 100.

The results of the clustering process appear in Figure 0 below. Clusters are color coded. For example the first cluster contains all the segments in the top two traces. Note that the timing marks are all in the same cluster colored orange.



\

Figure 0

Test 1 Frequency domain data is used. In particular the parameters of the FFT of selected points of the segments are computed and used to form the vectors that are used as input to the Frey Dueck clustering algorithm. In more detail, a segment is divided into 32 point subsegments. A remainder subsegment, if any, is zero-filled as in Test 0. The FFT of each subsegment is calculated. The output is a vector of 32 complex numbers. Their real and imaginary parts are stored in order in a vector of length 64. Finally, the mean of these vectors is computed resulting

in a single vector of length 64 for the segment. These vectors were used as input to the clustering method and the results are displayed in Figure 1.

It is important to compute the FFTs first and then average. Averaging the data first would affect the its frequency domain properties.

Note that the quiescent traces are resolved into very few clusters. The more active traces involve more clusters probably because of changes in the frequency of the signals within the trace. In this case as well the timing marks form their own cluster, colored orange.

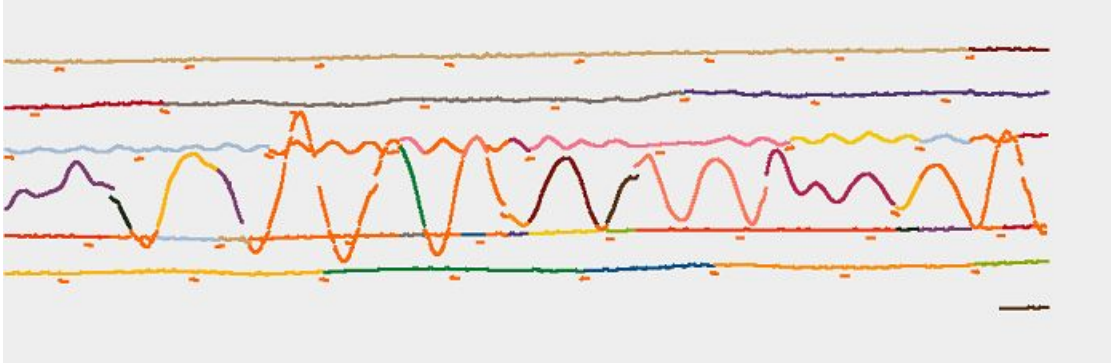


Figure 1

Test 2, *Test 3 and Test 4* These were similar to Test 1 except that in Test 2 the 16 FFT parameters with the largest magnitude (Test 2) or the 8 largest (Test 3) were selected for inclusion in the construction of the vectors and in Test 4 the vectors were normalized so that their mean values were zero. In all of these tests the results were unsatisfactory since almost all segments were assigned to the same cluster.

Test 5 In this test the power spectrum of each segment was computed using a method like that for computing the FFTs in Test 1. That is the segments were divided into subsegments, the power spectra were then computed and averaged. For each segment the average power spectrum across

subsegments was used as the input to the clustering process. The result appears in Figure 5. The vertical lines indicate the boundaries between adjacent segments.

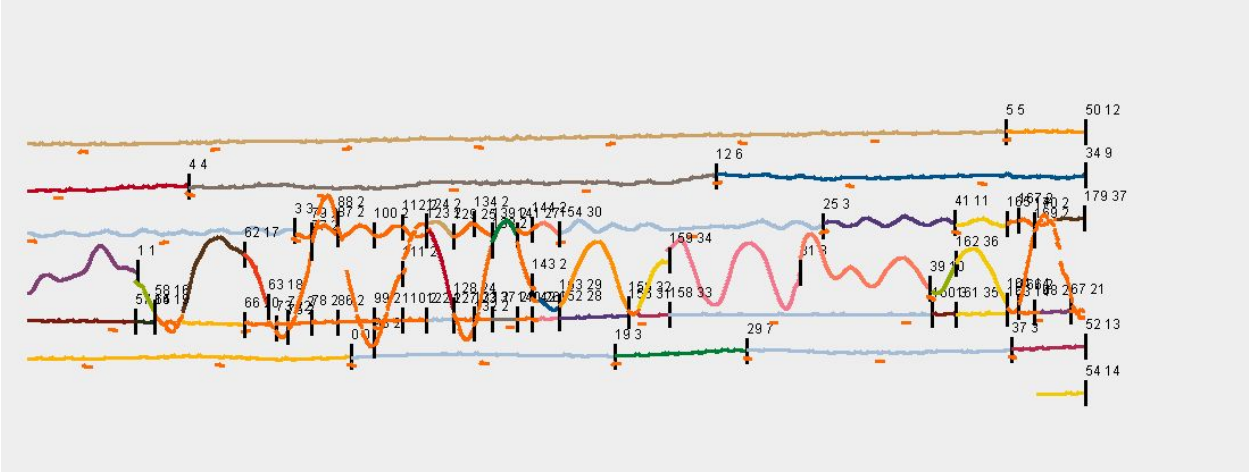


Figure 5

Discussion and Future Work

It was originally hoped that there would be a one-to-one relationship between clusters and traces and that the output of the clustering algorithm would therefore be an assignment of segments to traces. Clearly the situation is more complicated than that. It seems that the solution of the assignment problem will be a search procedure in which ensembles of sets of segments which have the property that there is little or no overlap in their time (i.e. x) ranges of but whose concatenations are close to continuous as signals will provide the final solution. The search procedure will likely benefit from knowledge of the clusters as found in Test 0, Test 1 or Test 5.

In the Frey-Dueck clustering method the exemplars of the clusters emerge from the procedure. This is in contrast to what is done in K-means clustering. In the latter the user specifies a set of exemplars and thereby determines the number of clusters and a target for their mean values.

Since in a given seismogram the number of traces will be known it seems reasonable that an exemplar could be constructed by sampling the mean line at a rate consistent with the sampling rate of the signals. It is important that if it were possible to correctly assign segments to mean lines it is then relatively easy to construct the complete signal. This is because of the assumption that segments do not overlap except perhaps slightly at the margins. Again, the output of a successful K-means clustering procedure could guide the search algorithm mentioned above.

Bayesian Methods

We will use probabilistic methods to determine the likelihood of any particular segment connection coming out of an intersection or gap being correct. To date we have investigated one specific intersection type, as represented in Figure 9, using real data. In this case, two segments come into an intersection, and two exit it. We compare the various connection possibilities 1 to 3, 2 to 4; 1 to 4 and 2 to 3 using Bayesian methods and a fitting model for the segments.

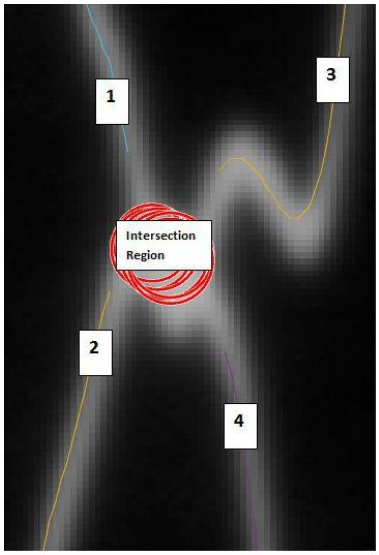


Fig. 9 Typical intersection investigated with Bayesian methods.

The Bayesian method first computes the ratio of the probabilities of the first possible set of connections in the time forward direction, to the second, alternate, set of connections in the time forward direction. This forward ratio is represented as:

P(1 >3 AND 2> 4)/P(1>4 AND 2>3)

The same is done in the time backwards direction, computing the backwards ratio as:

((3>1 AND 4>2)/P(4>1 AND 32)

We used a Kalman filter to model the segments. The advantage of this probabilistic method is that any number of filters and fits can be tried, in order to best suit seismic data[1]. An extrapolation of the modeled path returns a distribution on the predicted points in the path. Typically this is Gaussian, but can be altered depending on requirements. From this distribution a probability for each path is determined.

Using this information, Figure 10 shows a plot of odds ratios, using a log scale as is convention. What is desired for good connections is either high ratios both forward & backwards or low ratios both forward & backwards. This corresponds to wanting things in the upper right quadrant or lower left quadrant, and indicates either the connection in the numerator or denominator, respectively, are most likely.

Figure 10 shows a plot obtained using data from 57 connections, using data from SKATE. There are lines on the plots at x=+1/1 and y=+1/1 to better identify the quadrants. This corresponds to odds ratios > 2 or < 0.5 which is somewhat arbitrary but on inspection represents a good boundary value[2]. The cross-shaped region enclosed by these boundaries represent 'bad' regions, meaning a failure for either forward or backward paths to be likely.

The plot shows all the intersections that were tested, with most residing and clustered in the upper right quadrant, indicating not only a good fit, but also validating the approach[3]. In other words, if a significant number of intersections resided in the $1 < (x,y) < 1$ 'bad' region, that would have represented an inability of the method to discriminate between the choices of paths.

---

[1] For example, we tried financial filters and had less success because, unlike seismic data, financial data does not revert to the mean over time.

[2] When intersection type didn't fit the idealized model, which occasionally happened for various reasons, the method invariably failed. Note that this wasn't a failure of the method, only of the assumed intersection type that we were modeling.

[3] We would expect to see some in the lower left as well, but believe that that the spatial ordering of the segment data as outputted by SKATE skewed results towards Quadrant I.

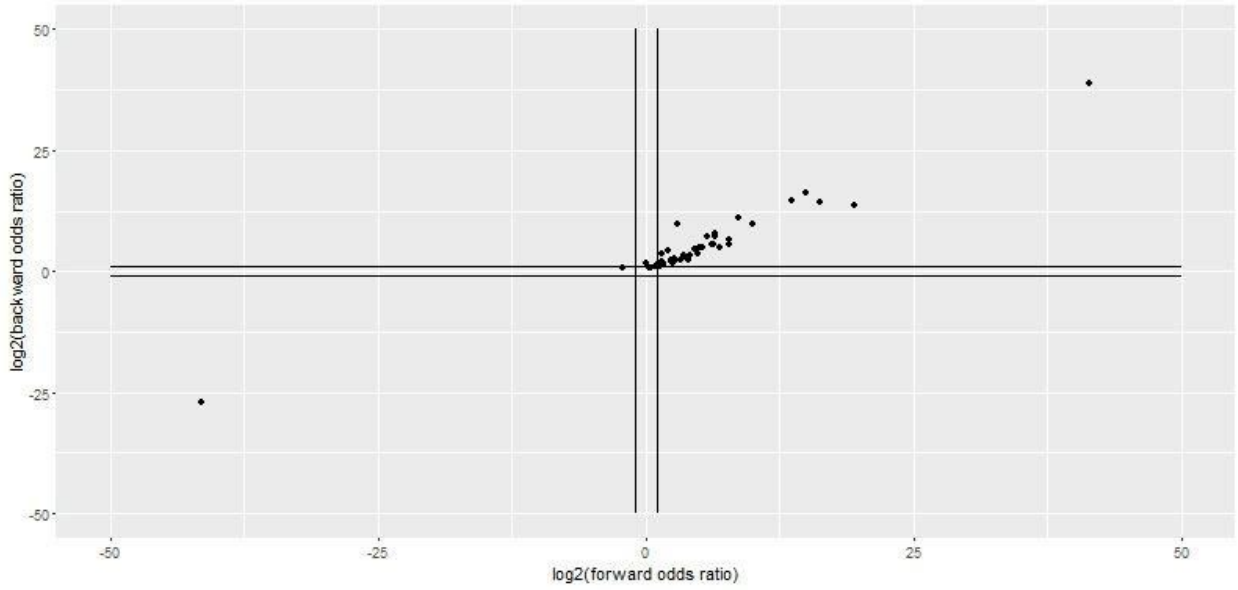


Fig. 10 Probability ratios for possible connections.

This Phase I work will use this approach on at least the subset of questionable intersections, if not all intersections and gaps. Computationally it is very fast, particularly compared to connection methods that rely on calculations in image space. This again highlights the advantage of working in data space vs. image space; it is significantly faster and therefore much less expensive. We will generalize the code to handle N-segment intersections (where N will typically be a maximum of 6). We will investigate the weighting of the path choices based on the odds ratio and use this value (weighted or not) in the overall path selection criteria as discussed in Work Plan 4c..

MTSP (Multiple Traveling Salesman Problem)

represents all segments in a seismogram as nodes on a graph that need to be connected. We will use an ‘ant algorithm’ on the graph to cast this as a multiple traveling salesman problem with the number of salesmen equal to the number of meanlines. We will represent each segment as a node with a domain

over which it exists. An advantage of this technique is that it doesn't require any intersection information; it works with all segments regardless of their start and end points.

The nodes/segments have internal properties of period, amplitude and phase, gray scale and more, which we already obtain and store for use as needed. One important constraint to leverage is that two nodes with overlapping time domains are not connectable and can not be on the same trace line. Another constraint is the requirement that the total integral of the path with respect to its mean, or zero-energy line, is equal to zero.

The nodes are randomly connected to other nodes within some given time and amplitude distance threshold (limiting the search field), creating a graph of multiple paths from one side of the seismogram to the other. Any path can only be traversed once, and can only be claimed by a single agent (ant) at a time. A solution/tour is judged first on whether all agents make it across the whole seismogram which will immediately eliminate many bad or impossible tours and which will reinforce certain edges/routes. Next, tours will be judged and their paths reinforced based on shortest distance traveled. "Distance" can include not just pixel distance but also closeness of fit in, e.g., fourier space, intensity values, and other metrics. By running the algorithm multiple times, those most likely paths will become reinforced, and unlikely or impossible paths such as those that end up associated with a different mean line will be eliminated. Figure 3 shows the results of one such set of multiple tours, with potential paths shown in white. (No tours are on the bottom half because the seismogram is quiescent there.) One advantage of this technique is computational speed, as simple matrix calculations drive the path; we can run this hundreds or thousands of times very quickly. We also anticipate that this will help to eliminate noise by not requiring every segment (meaning a segment could be noise and not real data) to have an assignment.

All likely connections will be assigned a weighting value, whose value will be determined in part by the outcome of the MTSP tour itself[4]. These results will be used for the overall path selection criteria in multi criteria decision making.

[4] Initially all tours will have equal weights until enough outcomes give information for weighting schemes.

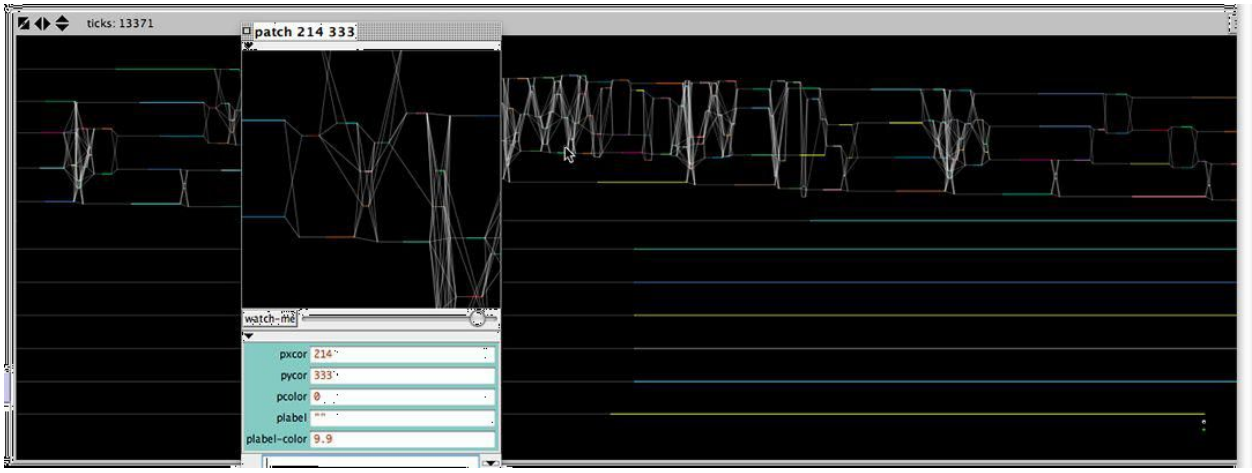


Fig. 3 Possible paths an agent can take traversing a seismogram. All possible paths are shown.

Multi criteria decision making

All of the connection metrics result in a list of possible paths for all segments that then need to be combined into a single combined solution. A multi criteria decision matrix will be constructed that collates all N segments into a single N x N matrix.

Figure 4 is an idealization of this matrix. The rows will be all segments, 1...N, and represent the end of a segment that needs a connection forward in time. The columns are again all segments, 1… N, that represent the beginnings of segments that need a connection backwards in time. Each possible connection will have a score assigned to it based on a summation of scores. We will investigate weighting of connection scores, but will start with equal weighting until methods suggested by the data become available.

The table has '**X**' values for those impossible connections, i.e., those connecting backwards in time. The locations listed as 'not allowed' represent some value in time forward space where it is unlikely that a connection will be found. Once all possible segment connections are made we will assign all connected segments to the nearest mean line; domains for large numbers of connected segments will be easier to locate. Unconnected segments will occur; comparison with empty domains, comparison against noise values, and iteration of this solution with any combination of connection algorithms are several of the ways to check the solution.

We will output the data into a .csv or other similar type file, utilizing our current standard of columnar x-y data for each meanline. This output will then be used in the final time series as discussed in Work Plan 2c.

	1	2	3	4	5	6	7	8	9	…	N
1	X	Value 1a	Value 2a	Value 3a	Value 4a	Value 5a	Not allowed	Not allowed	Not allowed	Not allowed	Not allowed
2	X	X	Value 1b	Value 2b	Value 3b	Value 4b	Value 5b	Not allowed	Not allowed	Not allowed	Not allowed
3	X	X	X	Value 1c	Value 2c	Value 3c	Value 4c	Value 5c	Not allowed	Not allowed	Not allowed
4	X	X	X	X	Value 1d	Value 2d	Value 3d	Value 4d	Value 5d	Not allowed	Not allowed
5	X	X	X	X	X	Value 1e	Value 2e	Value 3e	Value 4e	…	Not allowed
etc	…	…	..	…	…	…	…	…	…	…	…

Fig. 4 Multi-criteria decision making metric. Values are sums of (possibly) weighted inputs.