Get something working

- The main idea: use story curves to prime users so they can remember data better.

1. Parse data

a. Segmentation: Make piecewise linear approximation of data

- Bottom-up Piecewise Linear Approximation

- Start from the finest approximation, then segments are iteratively merged.

- First read about in Esling and Agon, 2011?

- Read Keogh and Pazzani, 1998 about this. KEOGH, E. AND PAZZANI, M. 1998. An enhanced representation of time series which allows fast and accurate classification,

clustering and relevance feedback. In Proceedings of the 4th International Conference of Knowledge Discovery and

Data Mining. AAAI Press, 239–241.

- Read Himberg et al., 2001 for fast greedy algorithms for this. HIMBERG, J., KORPIAHO, K., TIKANMAKI, J., AND TOIVONEN, H. 2001. Time series segmentation for context recognition

in mobile devices. In Proceedings of the 1st IEEE International Conference on Data Mining. 203–210.

HIMBERG, J., MANTYJARVI, J., AND KORPIPAA, P. 2001. Using PCA and ICA for exploratory data analysis in situation

awareness. In Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent

Systems. 127–131.

Himberg et al. Time series segmentation for context recognition in mobile devices. 2001.

- Dynamic programming solution for the k-segmentation problem could solve this, but “is computationally too hard for long sequences” due to the complexity of the algorithm.

- Global Iterative Replacement “gives approximately optimal results in a fraction of the time required by dynamic programming.” And is much faster than dynamic programming.

- Note: as the number of segments grows, the average relative error of the method grows as well; sort of. Seems to get bad starting at ~15. Length of sequence actually makes error go down

- This is error compared to the optimal, which is the dynamic programming algorithm.

- “The computational complexity of the greedy methods is linear in the size of the input data.” Of order O(KN), where K is the number of segments and N the size of the input.

The algorithm itself:

Input: Number *k* of segments. Set of datapoints *D* = p0 through pn, where pm = (xm, ym).

1. Set initial segmentation S0 s.t. there are *k* segments. Results in a set of breakpoints b0 through bk+1, including the start point b0 = p0 and the end point bk+1 = pn. For now, spaces initial breakpoints evenly.

2. Select a break point (randomly or sequentially, will do sequentially for now) br where r /= 0 or k+1, remove it, and concatenate the two consecutive segments. Call this new segmentation S0,r. At this point we have one less break point, k instead of k + 1.

3. Find a new optimal location for a break point anywhere in the sequence. Go through each segment in S0,r, find the optimal 2-segmentation (optimal split of the segment, linear time), and compute the potential savings for each segment (e.g. cost(old segment) – (cost(new segment 1) + cost(new segment 2)).

- The cost function of any single segment is “the internal heterogeneity” of a segment. The aim is to minimize this, e.g. increase the internal homogeneity of the segment.

- Cost function, F, that they used is “the sum of the variances of the components of the segment.”



n = b – a + 1 and d is the number of dimensions (2 for our data, x and y)

x(1), x(2), …, x(N) are points in a time series of N samples.

s(a, b) is a segment consisting of samples x(a), x(a+1), …, x(b).

- So, in the above equation, you’re doing, for both the x and y dimensions, sums from the first point in the segment (point a) to the last point in the segment (point b).

- e.g., x1(a) = the first x-point in the segment. x2(a) = the first y-point in the segment.

4. Choose the split with the largest savings, and set a new breakpoint at the optimal place within that split.

5. Renumerate the segments (e.g. reset them w/ the additional breakpoint added in step 4) into the new segmentation S1.

6. Steps 2 through 5 are repeated until a stopping criterion is met (when total cost cannot be decreased any further, or when a certain number of iterations has been reached).

- Read Vasko and Toivonen, 2002 for statistical method for choosing # of segments

b. Use frequency scanning methods to find interesting moments (a la weather paper)?

Story curves aren’t the only way of using story techniques

* Need to somehow have a mechanism to decide where to match, whether to match, etc. for applying story curves.
* Still need a conceptual example of what the whole system will be like when it’s done.
* Need to start pilot testing.