A Comparison Between Two Similarity Measures Proposed by "Flexible and Efficient Similarity Querying for Time-series Data" by Goldin, et al. and "A Geometrical Solution to Time Series Searching" by Zhou, et al.

Haibei Zhang

(In this report the "Flexible and Efficient Similarity Querying for Time-series Data" paper is referred to as "paper 1", "A Geometrical Solution to Time Series Searching" is referred to as "paper 2".)

Are these two papers trying to solve the same problem?

Yes. They both try to design a similarity measure to compare two sequences (usually time series) such that given a query sequence, the database returns those sequences (represented as vectors) stored in the database that are "similar" to the query sequence. Real world applications of such sequences can be stock prices, DNA sequences, etc.

Do they use the same assumptions?

They use following same assumptions:
The values collected over time are discrete. For example, the closing stock price of each day. These values also have equal interval between them. They also assume that the length of the query sequence equals the length of the stored sequence. Although paper 1 proposes a method to query a sequence (length=n) on a longer sequence (length=m>n) and look for any similar sub-sequences in that longer sequence, it is actually based on equal-length sequence query. The difference is that the equal-length query is executed m-n+1 times to compare the query with all its sub-sequences.

They both assume that amplitude (Y-axis) scaling and shifting are not significant in the problem because they do not affect the shape of the sequence. The target problem cares the shape only. "Shape" is the key feature that determines if two sequences are similar. The effect of scaling and shifting can be eliminated by some normalization method. Any sequences, when transformed into normal forms that are same, are believed equivalent.

Do they use the same techniques?

They use different techniques:
Paper 1:

1. Normalize the data sequence and the query sequence by substituting each element X_i with $(X_i-\alpha)/\sigma$ where α is the mean and σ is the standard deviation.

2. If using traditional Euclidean distance approach and if the dimensionality is high, the database will have to compute each element of the 2 vectors and suffer from rapidly increasing time. Hence paper 1 introduced the DFT function to compact the feature energy of the original vector such that the database only has to compute constant-length short vectors. This step filters our most unmatched sequences. Paper 1 proved that the returned fingerprints lower-bounds the actual result, meaning if the Euclidean distance between two vectors is below tolerance, then the distance between their fingerprints is below tolerance, too. The result ensures no false dismissal but may contain false alarms. (step 1 and 2 are called internal query)

3. Since the result returned by the previous step may contain false alarms, a post-query step is required to filter out false alarms. The actual Euclidean distance is then calculated on the original 2 vectors (the database sequence may be scaled and shifted within user-specified mean and deviation bounds). Since paper 1 proved the accuracy of DFT transformation, i.e. the rate of false alarms is low, most unmatched sequences are not calculated in this step, hence improving time performance.

4. To index a vector, the database stores a vector's fingerprint's Minimal Bounding Rectangle in R*-tree. Upon a query, the database looks for those MBRs that intercept with the query's fingerprint's rectangle, then filters out false alarms from these MBRs, then filters out false alarms from matching fingerprints.

Paper 2:

1. Shifting effect is eliminated by projecting the original vector on the Shifting-Eliminated Hyper Plane (SE-plane). The SE-Plane is perpendicular to the shifting vector N. Scaling can be further eliminated by dividing the Shifting Eliminated vector by the standard deviation, which remains all resulting vector as long as $n^{1/2}$. However, eliminating scaling is not required by this paper in following steps because the main criterion is the angle between the two projected vectors.

2. On the SE-plane, compute the angle between the two Shifting Eliminated vectors. See if the angle is smaller than the tolerance. (Note: on Paper 2 page 14 section 4, the criterion of similarity is "$\cos\theta<\varepsilon$". Is that a typo? Shouldn't it be "$\cos\theta>\varepsilon$" or "$\sin\theta<\varepsilon$" ? When θ goes down, $\cos\theta$ goes up.) Paper 2 allows user to specify a tolerable range of scale factor a, b where a*b=1. $||T_{se}(S)||/||T_{se}(Q)||$ or $||T_{se}(Q)||/||T_{se}(S)||$must fall within [a, b] to satisfy similarity. Unfortunately, paper 2 does not provide a way to specify a tolerable range of shift factor.

3. To index a vector, the database stores the vector's interception point on a hyper cube, slice the cube with balanced load factor, and store these slices in some data structure, e.g. R*-tree. The query vector, along with a tolerable angle, is implemented as a cone. The database retrieves all slices on the hyper cube that intercepts with the cone, and finally filters out false

alarms.

Following is a comparison between the 2 techniques:

	Paper 1	Paper 2
Normalization Transformation	$\hat{S}=T_{1/\sigma,-\alpha/\sigma}(S)$ (substitute each S_i with $(S_i-\alpha)/\sigma$)	Eliminate shifting: $T_{se}(S)=S-(S\cdot N/\|\|N\|\|^2)N=S-\alpha(S)N$ Eliminate scaling: $\hat{S} = (S-\alpha(S)N)/\sigma(S)$
Fingerprint Transformation	$DFT(\hat{S})=n^{-1/2}\sum S_j e^{-j(2\pi i)m/n}$ $F(\hat{S})=\{\alpha(S),\sigma(S),DFT_1(\hat{S})\ldots DFT_m(\hat{S})\}$	None
Compute distance	$D_F(F(Q), F(\hat{S}))\leq\varepsilon$?	$\sin\theta=\|\|Q-S\|\|/T_{se}(Q)$ or $\|\|S-Q\|\|/T_{se}(S)$ $\theta\leq\varepsilon$?
Filter out false alarms from fingerprint matches	$D(Q, a_0S+b_0) \leq\varepsilon$?	None
Indexing	MBR that contains continuous fingerprints	Points grouped by slices on a hyper cube
Retrieving using index	1. Get all MBRs that intercept with the query rectangle 2. Compute fingerprint distance to filter out false alarms in MBR 3. Compute actual distance to filter out false alarms in fingerprint matches	1. Get all slices that intercept with the query cone. 2. Compute actual angle to filter out false alarms in the slice (those vectors that are in the slice but out of the ellipse)

Same Weakness

Only amplitude scaling and shifting with global constant factors are considered "preserving the shape" by these two techniques. Time-scaling, bi-scaling, time-warping, non-uniform amplitude scaling won't be recognized as equivalent or similar, though these transformed vectors preserve the original shape to some extent.

Paper 1's Strength (Comparatively, paper 2's weakness)

1. Better performance
   Paper 1 proposes that DFT transformation be used to compact the feature energy of a vector around its head, then use the mean, standard deviation and a few elements in the head of the DFT vector to assemble a short, constant-length fingerprint. The fingerprint keeps the vector's shape information, while still ensures validity (no false dismissal), accuracy (low false alarm rate, for most real world data). Comparing fingerprints significantly improves

performance when dimensionality is high. Although this technique introduces some overhead to do transformation and post-processing (false alarm filtering), the performance gain is more significant than the trade-offs since most unsimilar vectors are filtered out by fingerprints. Paper 1 provided proof of validity (no false dismissal) and accuracy (few false alarms) to show the technique's correctness and high efficiency. The proof of continuity (constant time to compute a neighboring fingerprint) shows that this technique is especially efficient in matching the query with continuous sub-subsequences of a longer sequence.

Paper 2's technique does not reduce dimensionality by using fingerprints. For any input vector, it computes the vector's projection on the SE-plane from scratch and with all elements of the vector. If dimensionality is high, this technique will suffer from a performance degradation. Especially when the query sequence tries to match sub-sequences of the data sequence, more efficient algorithms could be developed to process continuous sequences.

2. Setting of shifting bounds
   Paper 1 allows user to specify a distance tolerance, a shifting factor tolerance as well as a scaling factor tolerance, thus it offers more flexibility. When internal query is executed, the fingerprints' distance is checked against the distance tolerance; the fingerprint's mean and standard deviation are also checked against tolerance values. When external query is executed, the Euclidean distance is checked against the distance tolerance; the data vector's mean and standard deviation are checked against scaling and shifting factor tolerance.

   Paper 2 cares about bounds for scaling factor only. The bounds for shifting factor cannot be set. The author believes that the similarity becomes meaningless if the vector's scaling factor is very small (in which case the vector almost reaches the X-axis). In the normalization step, when vectors are projected onto the SE-plane, the shifting factor is completely eliminated. The paper does not develop any method to set bounds for shifting factor.

3. Detailed proof
   Paper 1 provides detailed evidence to prove that the fingerprint approach is valid, accurate, continuous and updateable.

   Paper 2 does not come with much evidence on some of the author's statements. For example, the author claims "the most obvious difference (dissimilarity) between two radials is the angle between them." Therefore, the author decided to use the angle instead of minimal attainable distance. However, the author did not provide any specific and proven reason why the angle is better than the minimal attainable distance.

Paper 2's Strength (Comparatively, paper 1's weakness)

1. Flexibility in choosing different distance metrics
   While the technique in paper 1 normalizes all query or data vectors and uses Euclidean distance as the only distance metric, paper 2 discusses several different distance measures. One important notion is the Minimal Attainable Distance. Paper 2 shows an example that the minimal attainable distance between two vectors is smaller than the distance between their normalized vectors. In [1], as cited by paper 2, the author adopts an asymmetric method to compute a minimal attainable distance from one vector to another. Then the authors of [2] propose to use the smaller value of the two minimum distances as the similarity measure. This guarantees that the minimal attainable distance is obtained. In later sections, paper 2 developed a few different formulae to measure the distance in different ways. For example, if minimal attainable distance is desired, then simply choose the shorter transformed vector on the SE-plane and multiply it by $\sin\theta$. If the angle between 2 transformed vectors is desired, then just use θ directly.

2. Detecting mirror-image similarity
   Paper 2 shows that if two SE vectors' angle is between $\pi/2$ and π, then these two vectors are similar to each other's mirror image to some extent.

   Since paper 1's technique is distance-based, the distance value is always positive. It cannot detect mirror-image similarity.

3. No restriction of sequence data
   Paper 1's DFT transformation requires that the sequence data be some kind of color noises, meaning that each vector element is somehow consistent with its adjacent elements. If the sequence elements are completely random, then the DFT transformation does not compact any feature energy hence is not able to produce shorter, constant-length fingerprints. In paper 2, since no optimization is implemented, the technique treats any sequence equally.

Is there a way to combine their strength into a unified approach that is better than either of them separately?

Paper 2 claims that angle-based metric is better than distance-based metric. However, it does not show much evidence in terms of validity, accuracy, continuity, updateability. Further work is needed to prove that angle-based metric is at least as powerful as distance-based metric.

Paper 2 does not address any optimization strategy. Since real-world data sequences are usually somehow consistent, it is important to reduce dimensionality (though at the cost of increasing levels of transformation) and improve performance. A combined design may be: using distance-based metric and fingerprint approach to rapidly filter out unqualifying vectors, then applying angle-based metric within a smaller set of vectors to get the most precise result.

References

[11] Kelvin Kam Wing Chu and Man Hon Wong. Fast Time-Series Searching with Scaling and Shifting. In Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania, pages 237–248. ACM Press, 1999.

[2] T. Kahveci, A. Singh, and A. Gurel. Shift and scale invariant search of multi-attribute time sequences. Technical report, UCSB, 2001.