Extending the Edit Distance Using Frequencies of Common Characters

Muhammad Marwan Muhammad Fuad and Pierre-François Marteau

VALORIA, Université de Bretagne Sud
BP. 573, 56017 Vannes, France
{marwan.fuad, pierre-francois.marteau}@univ-ubs.fr

Abstract. Similarity search of time series has attracted many researchers recently. In this scope, reducing the dimensionality of data is required to scale up the similarity search. Symbolic representation is a promising technique of dimensionality reduction, since it allows researchers to benefit from the richness of algorithms used for textual databases. To improve the effectiveness of similarity search we propose in this paper an extension to the edit distance that we call the extended edit distance. This new distance is applied to symbolic sequential data objects, and we test it on time series data bases in classification task experiments. We also prove that our distance is a metric.

Keywords: Time Series, Symbolic Representation, the Edit Distance.

1 Introduction

Similarity search is an important problem in computer science, and it has a large number of applications. Research in this area has focused on its different aspects. One of these aspects is the distance metric used to measure the similarity between two data objects. Another aspect of this problem is the so called "dimensionality curse". One of the best solutions to deal with dimensionality curse is to utilize a dimensionality reduction technique to reduce dimensionality, then to utilize a suitable indexing structure on the reduced data objects. There have been different suggestions to represent time series, to mention a few; *DFT* [1] and [2], *DWT* [03], *SVD*[7], *APCA* [6],*PAA* [5] and [11], *PLA* [9]...etc.. However, among dimensionality reduction techniques, symbolic representation has many interesting advantages; it allows using text-retrieval algorithms and techniques [6]. In the beginning distance measures available for symbolic data processing were restricted to data structures whose representation is naturally symbolic (DNA and protein sequences, textual data…etc). But later these symbolic measures were also applied to other data structures that can be transformed into strings by using some symbolic representation techniques. There are quite a few distance metrics that apply to symbolically represented data. One of these measures is the edit distance (*ED*) [10], which is defined as the minimum number of delete, insert, and substitute (change) operations needed to transform string *S* into string *T*. Other measures for sequence alignment were proposed. The edit distance has a main drawback; it penalizes all change operations in the same way, without taking into account

S.S. Bhowmick, J. Küng, and R. Wagner (Eds.): DEXA 2008, LNCS 5181, pp. 150 – 157, 2008.

the character that is used in the change operation. In order to overcome this drawback we could predefine cost functions that gave the costs of all possible change operations. But this approach is inflexible and highly dependent on the application and corresponding alphabet.

In this paper we propose a new general distance metric that applies to strings. We call it "The Extended Edit Distance" (*EED*). This distance adds new features to the well-known edit distance by adding an additional term to it. The new distance has a main advantage over the edit distance in that it deals with the above mentioned problem straightforwardly, since there is no need to predefine a cost function for the change operation. This distance can, by itself, detect if the change operations use characters that are "familiar" or "unfamiliar" to the two strings concerned.

The rest of this paper is organized as follows: in section 2 we present a motivating example followed by the *EED*. Section 3 contains the experiments that we conducted, we discuss the results in section 4, and conclude in section 5 with some perspectives.

2 EED

2.1 Motivating Example

Given the string $S_1 = marwan$, by performing two change operations in the first and fifth positions we obtain the string $S_2 = aarwin$. By calculating their edit distance we get; $ED(S_1, S_2) = 2$. Let NDC be the number of distinct characters that two strings contain, i.e. $NDC(S_1, S_2) = \left|\{ch(S_1)\} \cup \{ch(S_2)\}\right|$, where *ch( )* is the set of characters that a string consists of. In our example we have; $NDC(S_1, S_2) = 6$. Now if we change the same positions in S_1 with different characters b, e we obtain the string: $S_3 = barwen$. By calculating the edit distance we get; $ED(S_1, S_3) = 2$ (which is the same as $ED(S_1, S_2)$). But we notice that $NDC(S_1, S_3) = 7$. This means that one change operation used a character that is more "familiar" to the two strings in the first case than in the second case, in other words, S_2 is closer to S_1 than S_3. But the edit distance couldn't recognize this, since the edit distance was the same in both cases.

2.2 Definition-The Extended Edit Distance

Let Σ be a finite alphabet, and let Σ^* be the set of strings on Σ. Let $f_a^{(S)}, f_a^{(T)}$ be the frequency of the character a in S and T, respectively. Where S, T are two strings in Σ^*. The extended edit distance (*EED*) is defined as;

$$EED(S,T) = ED(S,T) + \lambda\left[|S| + |T| - 2\sum_{a \in \Sigma} \min(f_a^{(S)}, f_a^{(T)})\right]$$

Where $|S|, |T|$ are the lengths of the two strings S, T respectively, and where $\lambda \geq 0$ ($\lambda \in R$). We call λ the co-occurrence frequency factor.

Revisiting the example presented in section 2.1 we see that $EED(S_1, S_2) = 4$, $EED(S_1, S_3) = 6$; which is what we expected, since, according to the concept of similarity we presented in section 2.1, S_2 is more similar to S_1 than S_3.

2.3 Proposition (P1): *EED* Is a Distance Metric

Let D be a set of objects. A function $d : D \times D \rightarrow \mathbb{R}^+$ is called a distance metric if the following holds $\forall x, y, z \in D$:

(p1) $d(x, y) = d(y, x)$, (p2) $x = y \Leftrightarrow d(x, y) = 0$, (p3) $d(x, z) \leq d(x, y) + d(y, z)$.

We prove below that *EED* is a metric.

(p1): $EED(S,T) = EED(T,S)$ (this is obvious).

(p2): Since for all S in Σ^* we have $|S| = \sum_{a \in \Sigma} f_a^{(S)}$. we can easily verify that:

$$\lambda \left[|S| + |T| - 2 \sum_{a \in \Sigma} \min(f_a^{(S)}, f_a^{(T)}) \right] \geq 0 \qquad \forall S,T \tag{1}$$

Let's prove first that $EED(S,T) = 0 \Rightarrow S = T$:

If $EED(S,T) = 0$, and taking into account (1), we get the two following relations:

$$\lambda \left[|S| + |T| - 2 \sum_{a \in \Sigma} \min(f_a^{(S)}, f_a^{(T)}) \right] = 0 \tag{2}$$

$$ED(S,T) = 0 \tag{3}$$

From (3), and since *ED* is metric we get: $S = T$. The backward proposition $S = T \Rightarrow EED(S,T) = 0$ is obvious.

(p3): $EED(S,T) \leq EED(S,R) + EED(R,T)$

$\forall S, T, R$ in Σ^*. Since *ED* is metric, we have: $ED(S,T) \leq ED(S,R) + ED(R,T)$ (4)

Let $D(S,T) = |S| + |T| - 2 \sum_{a \in \Sigma} \min(f_a^{(S)}, f_a^{(T)})$. We have to show that for all S, T, R in Σ^*:

$$\lambda \cdot D(S,T) \leq \lambda \cdot D(S,R) + \lambda \cdot D(R,T) \tag{5}$$

First, we note that the following equivalences hold:

$$\lambda \cdot D(S,T) \leq \lambda \cdot D(S,R) + \lambda \cdot D(R,T) \Leftrightarrow |S| + |T| - 2 \cdot \sum_{a \in \Sigma} Min(f_a^{(S)}, f_a^{(T)})$$

$$\leq |S| + |R| - 2 \cdot \sum_{a \in \Sigma} Min(f_a^{(S)}, f_a^{(R)} + |R| + |T| - 2 \cdot \sum_{a \in \Sigma} Min(f_a^{(R)}, f_a^{(T)})$$

$$\Leftrightarrow \sum_{a \in \Sigma} Min(f_a^{(S)}, f_a^{(R)}) + \sum_{a \in \Sigma} Min(f_a^{(R)}, f_a^{(T)}) \leq |R| + \sum_{a \in \Sigma} Min(f_a^{(S)}, f_a^{(T)}$$

Since $|R| = \sum_{a\in\Sigma} f_a^{(R)}$, proving (5) is equivalent to proving (6):

$$\sum_{a\in\Sigma} Min(f_a^{(S)}, f_a^{(R)}) + \sum_{a\in\Sigma} Min(f_a^{(R)}, f_a^{(T)}) \le \sum_{a\in\Sigma} f_a^{(R)} + \sum_{a\in\Sigma} Min(f_a^{(S)}, f_a^{(T)}) \tag{6}$$

Second, we note that for all a in Σ we have:

$$Min(f_a^{(S)}, f_a^{(R)}) \le f_a^{(R)} \quad \text{and} \quad Min(f_a^{(T)}, f_a^{(R)}) \le f_a^{(R)}$$

Furthermore, for all a in $\sum$we have: Either

$f_a^{(R)} = Min(f_a^{(R)}, f_a^{(S)}, f_a^{(T)}) \Rightarrow Min(f_a^{(R)}, f_a^{(T)}) \le Min(f_a^{(S)}, f_a^{(T)}) \Rightarrow$

$Min(f_a^{(S)}, f_a^{(R)}) + Min(f_a^{(R)}, f_a^{(T)}) \le f_a^{(R)} + Min(f_a^{(S)}, f_a^{(T)})$

Or $f_a^{(S)} = Min(f_a^{(R)}, f_a^{(S)}, f_a^{(T)}) \Rightarrow Min(f_a^{(S)}, f_a^{(R)}) = Min(f_a^{(S)}, f_a^{(T)}) \Rightarrow$

$Min(f_a^{(S)}, f_a^{(R)}) + Min(f_a^{(R)}, f_a^{(T)}) \le f_a^{(R)} + Min(f_a^{(S)}, f_a^{(T)})$

Or $f_a^{(T)} = Min(f_a^{(R)}, f_a^{(S)}, f_a^{(T)}) \Rightarrow Min(f_a^{(R)}, f_a^{(T)}) = Min(f_a^{(S)}, f_a^{(T)}) \Rightarrow$

$Min(f_a^{(S)}, f_a^{(R)}) + Min(f_a^{(R)}, f_a^{(T)}) \le f_a^{(R)} + Min(f_a^{(S)}, f_a^{(T)})$

This shows that, for all a in $\sum$, the following inequality holds:

$$Min(f_a^{(S)}, f_a^{(R)}) + Min(f_a^{(R)}, f_a^{(T)}) \le f_a^{(R)} + Min(f_a^{(S)}, f_a^{(T)})$$

Summing over all a in $\sum$ we get a proof for proposition (6) and consequently a proof for proposition (5). Adding (4) and (5) side to side we get (p3): $EED(S,T) \le EED(S,R) + EED(R,T)$. From (p1), (p2), and (p3) we get (P1) and conclude that *EED* is a metric.

3 Empirical Evaluation

We conducted four experiments of times series classification task based on the 1-NN rule on the datasets available at *UCR* [12]. We used leaving-one-out cross validation. As mentioned earlier, our new distance is applied to data structures which are represented symbolically. Time series are not naturally represented symbolically, but more and more studies focus on symbolic representation of time series. One of the most famous methods in the literature is *SAX* [4]. *SAX*, in simple words, consists of three steps; 1-Reducing the dimensionality of the time series by using piecewise aggregate approximation *PAA* 2-Discretization the *PAA* to get a discrete representation of the times series 3-Using the *MINDIST* measure. To test *EED* (or *ED*) we proceeded in the same way for steps 1 and 2 above to get a symbolic representation of time series, then in step 3 we compared *EED* with *ED* and with the distance measure defined in *SAX*, all applied to the resulting strings.

3.1 The First Experiment

The aim of the this experiment is to make a direct comparison among *ED*,*EED* and *SAX* For this experiment, we used the same compression ratio that was used to test *SAX* (i.e. *1* to *4*). We also used the same range of alphabet size (*3-10*).

For each dataset we tune the parameters on the training set to get the optimal values of these parameters; i.e. the values that minimize the error. Then we utilize these optimal values on the testing set to get the error rate for each method and for each dataset. As for parameter λ , for simplicity, and in all the experiments we conducted, we optimized it in the interval *[0, 1]* only (*step=0.25*), except in the cases where there was strong evidence that the error was decreasing monotonously as λ increased.

For this experiment, we chose, at random, *4* datasets from the *20* datasets of *UCR* [12]. The chosen datasets were *CBF*, *Trace*, *Two_Patterns*, *Yoga*. After optimizing the parameters on the training sets, we used these parameters on the testing sets of these datasets; we got the results shown in Table. 1. (The best method is highlighted)

Table 1. The error rate of *ED*, *EED*, *SAX* on the testing sets of *CBF*, *Trace*, *Two Patterns*, and *Yoga*. The parameters used in the calculations are those that give optimal results on the training sets, the alphabet size was chosen from the interval *[3, 10]*.The compression ratio is (*1:4*).

	The Edit Distance (ED)	**The Extended Edit Distance (EED)**	**SAX**
CBF	0.029 α* =10	**0.026** **α =3, λ =0.75**	0.104 α =10
Trace	0.11 α =10	**0.07** **α =6, λ ≥ 1.25**	0.42 α =10
Two_Patterns	**0.015** **α =3**	**0.015** **α =3, λ =0**	0.081 α =10
Yoga	**0.155** **α =7**	**0.155** **α =7, λ =0**	0.199 α =10
MEAN	0.077	**0.066**	0.201
STD	0.067	**0.064**	0.155

(*: α is the alphabet size)

The results obtained show that *EED* was always better, or equal, to the other methods. Its average error is the smallest. The results also show that of all the three tested methods *EED* has the minimum standard deviation

3.2 The Second Experiment

This experiment is an extension of the first experiment; we didn't compare our new distance with *ED* and *SAX* only, but we also compared it with other distances that are applied for non–compressed time series. We chose the two most famous distances; *Dynamic Time Warping* (*DTW*) [7] and *Euclidean distance*. We chose randomly *4* datasets of the remaining datasets in *UCR* [12]. These were *Gun_Point*, *OSU Leaf*, *50words*, and *Fish*. We used the same compression ratio and the same range of alphabet size that we used with in the first experiment. We proceeded in the same way. We obtained the results shown in Table. 2.

Table 2. The error rate of *ED*, *EED*, *SAX*, *DTW* together with the *Euclidean distance* on the testing sets of *Gun_Point*, *OSU Leaf*, *50words*, and *Fish*. The alphabet size was chosen from the interval *[3, 10]*. The compression ratio is (*1:4*).

	Euclidean Distance	**DTW**	**ED**	**EED**	**SAX**
Gun-Point	0.087	0.093	0.073 α =4	**0.06** **α =4, λ =0.25**	0.233 α =10
OSULeaf	0.483	0.409	0.318 α =5	**0.293** **α =5, λ =0.75**	0.475 α =9
50words	0.369	0.310	**0.266** **α =7**	**0.266** **α =7, λ =0**	0.327 α =9
Fish	0.217	0.267	**0.149** **α =10**	**0.149** **α =10, λ =0**	0.514 α =10
MEAN	0.289	0.270	0.201	**0.192**	0.387
STD	0.173	0.132	0.111	**0.108**	0.131

The results of this experiment show that *EED* is superior to the other distances.

3.3 The Third Experiment

This experiment aims at studying the impact of using a wider range of alphabet size; *[3, 20]*, we proceed in the same way we did before; we randomly chose *7* datasets of the remaining datasets. The *7* chosen datasets were *Coffee*, *Beef*, *Adiac*, *ECG200*, *Wafer*, *Swedish Leaf*, *Face (all)*. The compression ratio is the same as before (*1:4*).

Table 3. The error rate of *ED*, *EED*, *SAX* on the testing sets of *Coffee*, *Beef*, *Adiac*, *ECG200*, *Wafer*, *Swedish Leaf*, and *Face (all)*. The alphabet size was chosen from the interval *[3,20]*.The compression ratio is (*1:4*).

	The Edit Distance (ED)	**The Extended Edit Distance (EED)**	**SAX**
Coffee	0.071 α =12,13	**0.0** **α =14, λ =0.25**	0.143 α =20
Beef	0.467 α =17	**0.4** **α =4, λ= 0.75**	0.433 α =20
Adiac	0.555 α =18	**0.524** **α =19, λ =1**	0.867 α =18
ECG200	0.23 α =13	0.19 α =5, λ =0.25	**0.13** **α =16**
Wafer	0.008 α =4	0.008 α =4, λ =0	**0.004** **α =19**
Swedish Leaf	0.344 α =4	0.365 α =7, λ =0.25	**0.253** **α =20**
Face (all)	0.324 α =7	0.324 α =7, λ =0	**0.305** **α =19**
MEAN	0.286	**0.257**	0.305
STD	**0.199**	0.200	0.284

EED was compared with *ED* and *SAX*. The final results on the testing sets are shown in Table. 3.

The results of this experiment show that for this range of alphabet size, the average error of the *EED* is the smallest. The standard deviation for *ED* for this range size is the smallest. However, it's very close to that of *EED.*

3.4 The Fourth Experiment

This experiment is designated to study the impact of using a different compression ratio. We conducted it on the rest of the datasets in *UCR* [12].The compression ratio of this experiment is (*1:5*). The alphabet range is *[3, 10]*. After proceeding in the same way that we used for the other experiments we got the results shown in Table. 4

Table 4. The error rate of *ED*, *EED*, *SAX* on the testing sets of *Lighting2*, *Lighting7*, *Synthetic Control*, *Face (four),Trace*, and *Olive Oil* the alphabet size was chosen from the interval *[3,10]*.The compression ratio is (*1:5*)

	The Edit Distance (ED)	**The Extended Edit Distance (EED)**	**SAX**
Lighting2	**0.311** **α =5**	**0.311** **α =5, λ =0,0.75**	0.377 α =3
Lighting7	**0.247** **α =5**	**0.247** **α =5, λ= 0**	0.479 α =7
Trace	0.11 α =10	**0.09** **α =8, λ =0.75**	0.36 α =10
Synthetic Control	0.077 α =8	0.05 α =6, λ =0.25	**0.03** **α =10**
Face (four)	**0.045** **α =5,6**	**0.045** **α =5,6, λ =0**	0.182 α =9
Olive Oil	**0.267** **α =7**	**0.267** **α =7, λ =0,...,1**	0.833 $\forall\alpha$
MEAN	0.176	**0.168**	0.377
STD	**0.112**	0.120	0.275

The results obtained show that *EED* was the best in almost all the datasets used in this experiment. The average error of *EED* is the smallest. However, the standard deviation for *ED* for this compression ratio is the smallest.

4 Discussion

In the experiments we conducted we had to use time series of equal lengths for comparison reasons only, since *SAX* can be applied only to strings of equal lengths. But *EED* (and *ED*, too) can be applied to strings of different lengths. We also didn't conduct experiments for alphabet size=2 because *SAX* is not applicable in this case (when *alphabet size =2* then the distance between any two strings will be zero with *SAX*, and for any dataset). However, it's important to mention that comparing *EED* or *ED,* with *SAX* was only used as an indicator of performance. In fact, *SAX* is faster than any of

EED or *ED*, even though the error it produces is greater in most cases than that of *EED* or *ED*.

In order to represent the time series symbolically, we had to use a technique prepared for *SAX* for comparison purposes. Nonetheless, a representation technique prepared specifically for *EED* may even give better results.

The main property of the *EED* over *ED* is that it is more precise, since it considers a global level of similarity that *ED* doesn't consider

5 Conclusion and Perspectives

In this paper we presented a new distance metric applied to strings. The main feature of this distance is that it considers the frequency of characters, which is something other distance measures do not consider. Another important feature of this distance is that it's metric. We tested this new distance on a time series classification task, and we compared it to other distances. We showed that our distance gave better results in most cases. The main drawback of this distance is that it uses the parameter λ , which is heuristic, it also increases the training phase. The future work concerns the elimination of this parameter.

References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Proceedings of the 4th Conf. on Foundations of Data Organization and Algorithms (1993)
2. Agrawal, R., Lin, K.I., Sawhney, H.S., Shim, K.,: Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In: Proceedings of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, pp. 490–501 (1995)
3. Chan, K., Fu, A.W.: Efficient Time Series Matching by Wavelets. In: Proc. of the 15th IEEE Int'l Conf. on Data Engineering, Sydney, Australia, March 23-26, 1999, pp. 126–133 (1999)
4. Lin, J., Keogh, E.J., Lonardi, S., Chiu, B.Y.-c.: A symbolic representation of time series, with implications for streaming algorithms. DMKD 2003, 2–11 (2003)
5. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra: Dimensionality reduction for fast similarity search in large time series databases. J. of Know. and Inform. Sys. (2000)
6. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra: Locally adaptive dimensionality reduction for similarity search in large time series databases. SIGMOD, 151–162 (2001)
7. Keogh, E.: Exact indexing of dynamic time warping. In: Proc. 28th Int. Conf. on Very Large Data Bases, pp. 406–417 (2002)
8. Korn, F., Jagadish, H., Faloutsos, C.: Efficiently supporting ad hoc queries in large datasets of time sequences. In: Proceedings of SIGMOD 1997, Tucson, AZ, pp. 289–300 (1997)
9. Morinaka, Y., Yoshikawa, M., Amagasa, T., Uemura, S.: The L-index: An indexing structure for efficient subsequence matching in time sequence databases. In: Proc. 5th PacifiCAisa Conf. on Knowledge Discovery and Data Mining, pp. 51–60 (2001)
10. Wagner, R.A., Fischer, M.J.: The String-to-String Correction Problem. Journal of the Association for Computing Machinery 21(I), 168–173 (1974)
11. Yi, B., K.: Fast time sequence indexing for arbitrary Lp norms. In: Proceedings of the 26st International Conference on Very Large Databases, Cairo, Egypt (2000)
12. UCR Time Series datasets, `http://www.cs.ucr.edu/~eamonn/time_series_data/`