Information Sciences 348 (2016) 153–162

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Generalized quasi-metric on strings

Fagner Santana[a], Regivan Santiago[a,*], Benjamin Bedregal[a], Daniel Paternain[b], Humberto Bustince[b]

[a] Federal University of Rio Grande do Norte, Central Campus, Natal Postal Code 59072-970, Brazil
[b] Universidad Publica de Navarra, Campus Arrosadia, Pamplona Postal Code 31006, Spain

ARTICLE INFO

Article history:
Received 4 August 2014
Revised 1 December 2015
Accepted 26 January 2016
Available online 12 February 2016

Keywords:
Strings
Generalized distance
Subsequence
Edit operations
Handwriting digit classification

ABSTRACT

In this paper, we propose a generalized quasi-metric in spaces of strings, which is based on edit operations (insertion and deletions) and taking values as pairs of non-negative integers. We show that with such a generalization is possible to carry more information about similarity between strings than in the usual case where the distance between the strings is a simple real number. An algorithm for the calculation of this quasi-metric is presented and as well as an illustrative example of the application of this quasi-metric in handwritten digit classification. We also show some relations of this quasi-metric with the concept of subsequence of a string.

1. Introduction

The problem of evaluating similarities between strings has important applications in many areas of computing. In general, the functions used for that purpose are called distance function, more specifically, metrics [3,15]. The metrics for strings are functions of the type $d : \sum^* \times \sum^* \longrightarrow \mathbb{R}$ satisfying:

1. $d(x, y) \geq 0$;
2. $d(x, y) = d(y, x)$;
3. $d(x, y) = 0$ iff $x = y$;
4. $d(x, z) \leq d(x, y) + d(y, z)$ (*triangle inequality*).

In this case, given two strings w and v, the value of $d(w, v)$ establishes a similarity degree (actually dissimilarity) between two strings, so that the greater the value of $d(w, v)$ smaller is the similarity between w and v. One of the first and most widely used metrics in spaces of strings is called the edit or Levenshtein distance [7]. This distance is defined as the lowest number of operations of insertion, removal and substitution of characters required to transform one string into another. This distance (or some reformulations of it) has been used in various applications, such as: information extraction, object identity, data mining, biological sequence analysis, on-line signature authentication, bioinformatics, etc. (see [8,12]). Most of these distance measures, such as generalized Levenshtein distance [14], are also functions of type $d : \sum^* \times \sum^* \longrightarrow \mathbb{R}$, i.e., $d(w, v)$ is a real number. Thus, the information of similarity between strings, which is codified in the value of distances between the two strings, is restricted to what a real number is able to inform. In [5] it was proposed a generalization of the

* Corresponding author. Tel.: +55 84 8723 8891; fax: +55 84 3215 3813.
E-mail address: regivan@dimap.ufrn.br, regivan.santiago@gmail.com (R. Santiago).

http://dx.doi.org/10.1016/j.ins.2016.01.097
0020-0255/

mathematical concept of distance in which the distance was a function of type $d : \sum^* \times \sum^* \longrightarrow V$, where V is an abelian and ordered monoid. So, $d(w, v)$ may be something more general than a real number and thus, can carry more information about the similarity between v and w. In this paper, we propose a generalized distance (quasi-metric) in a space of strings: μ_s. It is inspired by the Levenshtein distance, but with values as pairs of non-negative integers. This approach overcomes some deficiencies of the original method in the sense that the resulting pairs are able to inform which kind of operations (removals and insertions) were done during the process of measurement. This enables us, for example, to decide when a string is a subsequence of another. An algorithm which implements μ_s is provided. An application for handwritten digit recognition is made in order to show the viability of μ_s, since some type of image can be easily represented as strings.

This paper is organized in the following way: Section 2 presents the Levenshtein distance and the theory of generalized distances. Section 3 presents the function μ_s, some properties are proved and the algorithm which implements μ_s is also provided. Section 4 states the relation between μ_s and the notion of subsequences of strings and some advantages over Levenshtein approach. Section 5 presents the application of μ_s for handwritten digit recognition and compares the usual Levenshtein distance with μ_s. Finally, section 6 provides our final remarks.

2. Levenshtein distance and generalized distances

The Levenshtein distance [7] between the strings w_1 and w_2, $d_L(w_1, w_2)$, is defined as the smallest amount of edit operations (deletions and insertions) to transform w_1 into w_2 and vice-versa. For example, consider the alphabet $\Sigma = \{a, c, t, g\}$ and the strings: *acctg*, *cactga* and *cag*. $d_L(acctg, cactga) = d_L(cactga, acctg) = 3$ (the minimum is two insertions and one deletion) and $d_L(cag, cactga) = 3$ (three deletions). In both cases the values of d_L are the same, although the applied operations are completely different. In other words, the application of Levenshtein distance does not reflect the involved edit operations applied during the processes of measurement. This kind of information is enough, for example, to determine if one string is a subsequence of another. For example, *cag* is a subsequence of *cactga* but *acctg* is not. In this section we present the notion of generalized metric (and quasi-metrics), they will be applied in the next section generalizing the Levenshtein Distance in order to endow this distance with the information about the edit operations involved during the processes of measurement.

Definition 1. A set V in which are defined an associative, commutative binary operation $\oplus$ with identity element $0 \in A$ is called abelian monoid. If A is endowed with a partial order $\leq$ such that $\oplus$ and $\leq$ are compatible, i.e., $0 \leq u, \forall u \in A$ and $u_1 \oplus v_1 \leq u_2 \oplus v_2$ whenever $u_1 \leq u_2$ and $v_1 \leq v_2$, so $(A, \leq, \oplus, 0)$ is called abelian ordered monoid.

Definition 2. Let $(A, \leq, +, 0)$ be an abelian ordered monoid. A generalized metric on a non-empty set M is a function $d : M \times M \longrightarrow A$ satisfying:

1. $d(x, y) = 0$ iff $x = y$;
2. $d(x, y) = d(y, x)$;
3. $d(x, y) \leq d(x, z) \oplus d(z, y)$.

The pair (M, d) is called generalized metric space.

This definition of generalized metric is very similar to the definition of usual metrics (see [3]). The concept of generalized quasi-metric, which was not presented in [5], is formulated based on the concept of generalized metric, just like the usual case.

Definition 3. Let $(A, \leq, \oplus, 0)$ be an abelian ordered monoid. A generalized quasi-metric on a non-empty set M is a function $d : M \times M \longrightarrow A$ satisfying:

1. $d(x, y) = d(y, x) = 0$ iff $x = y$;
2. $d(x, y) \leq d(x, z) \oplus d(z, y)$.

The pair (M, d) is called generalized quasi-metric space.

In next section we will consider a specific abelian ordered monoid which will be the basis to define a generalized quasi-metric on the set of strings.

3. Function μ_s

Consider a finite alphabet $\sum = \{a_1, a_2, \ldots, a_n\}$ (every element of Σ is called a symbol) and let Σ^* be the set of all strings over Σ. Given $w \in \sum^*$, we will denote by w_c the set of symbols in w and the length of w by $|w|$. The empty string will be denoted by ε.

Example 1. If $\sum = \{a_1, a_2, a_3\}$, then $a_1a_2 \neq a_2a_1$, $|a_1a_1a_1a_1a_1| = 5$, and $(a_1a_1a_2a_2)_c = \{a_1, a_2\}$.

Every function $f : \sum^* \longrightarrow \sum^*$ will be called an edit operation.

Example 2. Let Σ be an alphabet. The concatenation of two strings over Σ^* is defined by: if $w = a_{r_1} \ldots a_{r_m}$ and $t = a_{s_1} .. a_{s_n}$, then the concatenation of w and t is the string $wt = a_{r_1} \ldots a_{r_m} a_{s_1} .. a_{s_n}$. For fixed $w \in \sum^*$ the function $f : \sum^* \longrightarrow \sum^*$ defined by $f(t) = wt$ is an edit operation over Σ^*.