Penalty-based aggregation of strings

Raúl Pérez-Fernández[1,2] and Bernard De Baets[1]

[1] KERMIT, Department of Data Analysis and Mathematical Modelling,
Ghent University, Coupure links 653, 9000 Gent, Belgium
{raul.perezfernandez,bernard.debaets}@ugent.be
[2] Department of Statistics and O.R. and Mathematics Didactics,
University of Oviedo, Oviedo, Spain
perezfernandez@uniovi.es

Abstract. Whereas the field of aggregation theory has historically studied aggregation on bounded posets (mainly the aggregation of real numbers), different aggregation processes have been analysed in different fields of application. In particular, the aggregation of strings has been a popular topic in many fields featuring computer science and bioinformatics. In this conference paper, we discuss different examples of aggregation of strings and position them within the framework of penalty-based data aggregation.

Keywords: Aggregation, Strings, Penalty functions.

1 Introduction

Binary strings are ubiquitous in computer science [1], whereas DNA sequences are a prominent type of string arising naturally in the field of bioinformatics [2]. It is no surprise then that the aggregation of strings has been extensively studied. The computation of median strings and center strings, which respectively minimize the sum of the distances and the maximum distance to the strings to be aggregated, surely represents the core problem in the aggregation of strings [3]. Different distance metrics have been considered for defining median and center strings [4]. The two most prominent examples are the Hamming distance metric [5], popular in coding theory, and the Levenshtein distance metric [6], popular for code correction.

It is nonetheless surprising that there has been little interest in this topic from the field of aggregation theory, especially bearing in mind that the study of median strings and center strings certainly resembles some classical problems for aggregation theorists. This is probably due to the fact that there is no meaningful order when dealing with strings, letting aside the literature-oriented alphabetic/lexicographic order, and the field of aggregation theory has historically been linked to processes that aggregate elements of a bounded poset. In a recent work [7], the present authors introduced a new framework for penalty-based data aggregation that does not restrict to the aggregation on ordered structures and that embraces the aggregation on stuctures equipped with a betweenness relation. In this paper, we position the search for median and center strings with respect to different distance metrics within this framework. We discuss both cases in which the strings are or are not restricted to have a fixed length.

2 The framework of penalty-based data aggregation

Penalty functions have been used for decades in the context of aggregation theory [8]. Mostly confined to the aggregation of real numbers [9], the current understanding of a penalty function is more or less as follows [10] (up to a positive additive constant).

Definition 1. *Consider $n \in \mathbb{N}$ and a closed interval $I \subseteq \mathbb{R}$. A function $P : I \times I^n \to \mathbb{R}^+$ is called a penalty function if:*

(i) $P(y; \mathbf{x}) \geq 0$*, for any* $y \in I$ *and any* $\mathbf{x} \in I^n$*;*
(ii) $P(y; \mathbf{x}) = 0$ *if and only if* $\mathbf{x} = (y, \ldots, y)$*;*
(iii) $P(\cdot; \mathbf{x})$ *is quasi-convex*[3] *and lower semi-continuous*[4] *for any* $\mathbf{x} \in I^n$*.*

In a recent paper [7], the present authors proposed a generalization of the definition of a penalty function based on the compatibility with a betweenness relation.

Definition 2. *A ternary relation B on a non-empty set X is called a betweenness relation if it satisfies the following three properties:*

(i) Symmetry in the end points: for any $x, y, z \in X$, it holds that

$$(x, y, z) \in B \Leftrightarrow (z, y, x) \in B\,.$$

(ii) Closure: for any $x, y, z \in X$, it holds that

$$\big((x, y, z) \in B \wedge (x, z, y) \in B\big) \Leftrightarrow y = z\,.$$

[3] Consider a closed interval $I \subseteq \mathbb{R}$. A function $f : I \to \mathbb{R}$ is called quasi-convex if, for any $u, v \in I$ and any $\lambda \in]0, 1[$, it holds that $f(\lambda u + (1 - \lambda)v) \leq \max(f(u), f(v))$.

[4] Consider a closed interval $I \subseteq \mathbb{R}$. A function $f : I \to \mathbb{R}$ is called lower semi-continuous if, for any $u \in I$, it holds that $\liminf\limits_{v \to u} f(v) = f(u)$.

(iii) End-point transitivity: for any $o, x, y, z \in X$, it holds that

$$\big((o,x,y) \in B \wedge (o,y,z) \in B\big) \Rightarrow (o,x,z) \in B\,.$$

Two betweenness relations of importance to this paper are the betweenness relation induced by a given distance metric and the product betweenness relation.

Proposition 1. *Consider a distance metric d on a set X. The ternary relation B_d on X defined as*

$$B_d = \big\{(x,y,z) \in X^3 \mid d(x,z) = d(x,y) + d(y,z)\big\}\ ,$$

is a betweenness relation on X, called the betweenness relation induced by d.

Proposition 2. *Consider $n \in \mathbb{N}$ and a betweenness relation B on a set X. The ternary relation $B^{(n)}$ on X^n defined as*

$$B^{(n)} = \big\{(\mathbf{x},\mathbf{y},\mathbf{z}) \in (X^n)^3 \mid (\forall i \in \{1,\dots,n\})((x_i,y_i,z_i) \in B)\big\}\ ,$$

is a betweenness relation on X^n, called the product betweenness relation.

Given a betweenness relation, we have proposed a definition of a penalty function in which the original third property aiming at providing some desirable semantics to the penalty is substituted by the requirement of the set of minimizers to be non-empty and by the compatibility with a betweenness relation aiming again at providing the penalty with some desirable semantics.

Definition 3. *Consider $n \in \mathbb{N}$, a set X and a betweenness relation B on X^n. A function $P : X \times X^n \to \mathbb{R}^+$ is called a penalty function (compatible with B) if the following four properties hold:*

(P1) $P(y;\mathbf{x}) \geq 0$, *for any $y \in X$ and any $\mathbf{x} \in X^n$;*
(P2) $P(y;\mathbf{x}) = 0$ *if and only if* $\mathbf{x} = (y,\dots,y)$;
(P3) The set of minimizers of $P(\cdot;\mathbf{x})$ is non-empty, for any $\mathbf{x} \in X^n$.
(P4) $P(y;\mathbf{x}) \leq P(y;\mathbf{x}')$, *for any $y \in X$ and any $\mathbf{x},\mathbf{x}' \in X^n$ such that $((y,\dots,y),\mathbf{x},\mathbf{x}') \in B$.*

Remark 1. If a betweenness relation on X is given instead of a betweenness relation on X^n, it is assumed that the product betweenness relation is considered.

The process of aggregation is then understood as a process of minimizing a penalty function given the list of objects to be aggregated. Different existing procedures coming from different fields are dicussed in [7] and are shown to fit within this framework.

Definition 4. *Consider $n \in \mathbb{N}$, a set X, a betweenness relation B on X^n and a penalty function $P : X \times X^n \to \mathbb{R}^+$ compatible with B. The function $f : X^n \to \mathcal{P}(X)$ defined by*

$$f(\mathbf{x}) = \arg\min_{y \in X} P(y; \mathbf{x}) \,,$$

for any $\mathbf{x} \in X^n$, is called the penalty-based function associated with P.

It is important to note that any aggregation process characterized as above is idempotent, i.e., the result of aggregating a list of n times the same object needs to be this very object. Additionally, one should note that more than one minimizer could be obtained. This is often the case in the setting of this paper in which we deal with the aggregation of strings.

The two most common examples of penalty-based functions are defined by means of the sum of distances or the maximum distance to the objects to be aggregated [7].

Corollary 1. *Consider $n \in \mathbb{N}$, and a metric space (X, d). The function $P : X \times X^n \to \mathbb{R}^+$ defined by*

$$P(y; \mathbf{x}) = \sum_{i=1}^{n} d(y, x_i) \,,$$

for any $(y; \mathbf{x}) \in X^{n+1}$, is a penalty function (compatible with $B_d^{(n)}$).

Corollary 2. *Consider $n \in \mathbb{N}$, and a metric space (X, d). The function $P : X \times X^n \to \mathbb{R}^+$ defined by*

$$P(y; \mathbf{x}) = \max_{i=1}^{n} d(y, x_i) \,,$$

for any $(y; \mathbf{x}) \in X^{n+1}$, is a penalty function (compatible with $B_d^{(n)}$).

Functions of the former type are usually referred to as *medians* and functions of the latter type are usually referred to as *centers*[5]. For some types of object such as real numbers and real vectors, *centroids* (which minimize the sum of squared distances to the objects to be aggregated) have also been extensively studied. However, centroid strings are to the best of our knowledge way less popular than median and center strings. For this very reason, centroid strings will not be discussed in this paper, although they would perfectly fit within the framework of penalty-based aggregation.

[5] In the context of the aggregation of strings, both terms ‘center string’ and ‘closest string’ are found to carry the same meaning.

In case the set X is finite, there is a large literature in the field of operations research on how to compute the minimizers of $P(\cdot; \mathbf{x})$ (for any given $\mathbf{x} \in X^n$). More specifically, we refer to the minimum facility location problem for the computation of medians and to the minmax facility location problem for the computation of centers [11].

3 Strings of the same length

Given an alphabet (set of characters) Σ, any list of m elements of Σ is called a string of length m. The set of all strings of length m (on an alphabet Σ) is denoted by Σ_m. For any $S \in \Sigma_m$ and any $j \in \{1, \ldots, m\}$, we denote by $S(j)$ the j-th element of S. In this section, we fix the value of m and we discuss four natural examples of distance metrics on the set of strings of the same length (m).

3.1 The discrete distance metric

The first and most trivial example of distance metric on the set of strings of length m is the discrete distance metric, defined as $\delta(S, S') = 0$, if $S = S'$, and $\delta(S, S') = 1$, otherwise. This distance metric induces the betweenness relation B_δ on Σ_m illustrated in Figure 1, known as the minimal betweenness relation. Note that the minimal betweenness relation is contained in any possible betweenness relation [7]. For this reason, the semantics brought to any penalty function by the betweenness relation B_δ are negligible.

Fig. 1. Illustration of the strings (in grey) that are in between the strings *cat* and *dog* (in red) according to the betweenness relation B_δ for the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$. Please note that there is no string in grey.

Still, one could think of identifying the median string(s) and the center string(s) of a given list of strings $\mathbf{S}$ with respect to the discrete distance metric. A median string of $\mathbf{S}$ is characterized as a string appearing with the highest frequency in $\mathbf{S}$, whereas any possible string in Σ_m is a center string of $\mathbf{S}$ unless there exists a string $S \in \Sigma_m$ such that $\mathbf{S} = (S, \ldots, S)$, a case in which S is the unique center string (due to the idempotence of any penalty-based function).

Example 1. Consider the following list of strings of length 3 on the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$: $\mathbf{S} = \{cat, dog, cat, dot, cog\}$. The unique median string is *cat*, whereas any possible string in Σ_3 is a center string.

3.2 The Hamming distance metric

The Hamming distance is a popular distance metric for strings of the same length [5]. Intuitively, this distance metric assigns to each couple of strings of the same length m the number of positions at which both strings differ, i.e.,

$$H(S, S') = |\{j \in \{1, \ldots, m\} \mid S(j) \neq S'(j)\}| \,.$$

This distance metric induces a betweenness relation B_H on Σ_m, illustrated in Figure 2. As can be seen, the semantics induced by any penalty function compatible with B_H is richer than that induced by any penalty function compatible with B_δ.

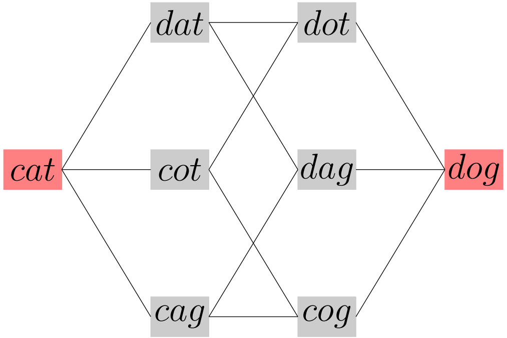

Fig. 2. Illustration of the strings (in grey) that are in between the strings *cat* and *dog* (in red) according to the betweenness relation B_H for the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$.

The search for the median string(s) and the center string(s) of a given list of strings $\mathbf{S} = (S_1, \ldots, S_n)$ with respect to the Hamming distance metric is a popular topic in computer science. Median strings with respect to the Hamming distance metric are easily characterized as the strings S such that $S(j)$ appears with the highest frequency in $(S_1(j), \ldots, S_n(j))$ for all $j \in \{1, \ldots, m\}$. Unfortunately, center strings with respect to the Hamming distance metric are not characterizable and they are known to be NP-hard to compute [1]. The latter problem of finding the center string (usually referred to as the closest string problem) is of interest to many scientific disciplines such as molecular biology and coding theory [12] and thus has led to many works aiming at finding an efficient algorithm/approximation for the closest string problem [13].

Example 2. Consider again the list of strings of length 3 on the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$: $\mathbf{S} = \{cat, dog, cat, dot, cog\}$. The unique median string is *cot* since c appears with the highest frequency (three) at the first position, o appears with the highest frequency (three) at the second position and t appears with the highest frequency (three) at the third position. Center strings are *dag* and all strings of the form $co*$ or $*ot$, where $*$ represents any element in Σ.

3.3 The lexicographic distance metric

The use of lexicographic orders arises naturally when dealing with strings [14]. In this setting, the most common lexicographic order is the widely-known alphabetical order. Because of its popularity, both terms are often used interchangeably when talking about strings. Obviously, the set of strings of length m is linearly ordered if we consider the lexicographic order $\leq$. Thus, as in every linearly ordered set, we can define the lexicographic distance metric:

$$L(S, S') = |\{S'' \in \Sigma_m \mid \min(S, S') \leq S'' \leq \max(S, S')\}| - 1 \,.$$

This distance metric induces the betweenness relation B_L on Σ_m (which amounts to $B_{\leq}$ if we consider the betweenness relation induced by the alphabetical order $\leq$ – for more details see [7]), illustrated in Figure 3. Note that this betweenness relation B_L carries a totally different semantics than B_H. For this very reason, a penalty-based function compatible with B_L will almost surely lead to quite different results than a penalty-based function compatible with B_H.

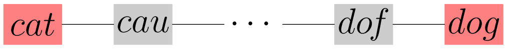

Fig. 3. Illustration of the strings (in grey) that are in between the strings *cat* and *dog* (in red) according to the betweenness relation B_L for the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$.

The search for the median string(s) and the center string(s) of a given list of strings $\mathbf{S}$ with respect to the lexicographic distance metric is similar to the search for the median(s) and center(s) of a list of real numbers. Actually, it suffices to identify each string with its position in the alphabetical order and compute the median of these positions for identifying the median string(s) (as usual, only assured to be unique if the number of strings is odd). If, instead of the median of the positions, we compute the arithmetic mean of the smallest position and the greatest position, then we would obtain the center string (in case this arithmetic mean of the smallest position and the greatest position is not a natural number, there will be two center strings identified by the floor and the ceiling of said value).

Example 3. Consider again the list of strings of length 3 on the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$: $\mathbf{S} = \{cat, dog, cat, dot, cog\}$. We identify the string *cat* with the position $2 \cdot 26^2 + 0 \cdot 26 + 19 \cdot 1 + 1 = 1372$, the string *dog* with the position $3 \cdot 26^2 + 14 \cdot 26 + 6 \cdot 1 + 1 = 2399$, the string *dot* with the position $3 \cdot 26^2 + 14 \cdot 26 + 19 \cdot 1 + 1 = 2412$ and the string *cog* with the position $2 \cdot 26^2 + 14 \cdot 26 + 6 \cdot 1 + 1 = 1723$. The median string is the string identified with the median of $\{1372, 2399, 1372, 2412, 1723\}$, which is 1723 – thus being the median string *cog*. The center string is the string identified with the position $\frac{1372+2412}{2} = 1892$. Since $1892 = 2 \cdot 26^2 + 20 \cdot 26 + 19 \cdot 1 + 1$, the center string is *cut*.

In the particular case in which we are dealing with existent words and not with just any possible string, both the median string(s) and the center string(s) could be nonexistent words even though all strings to be aggregated are perfectly fine existent words (for the case of the median string this is only possible if an even number of strings is to be aggregated). A potential way of solving this issue would require to compute medoid [15] (or set median) strings: instead of considering the set of minimizers of some certain $P(\cdot; \mathbf{S})$ in the whole Σ_m, we would just restrict our attention to the minimizers among those strings in $\mathbf{S}$.

Example 4. Consider the list of strings of length 3 on the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$: $\mathbf{S} = \{cat, car\}$. The center string is *cas*. Since this word does not appear in the English dictionary, one might think of computing a medoid-like center string (sometimes called set center string). We would thus obtain both *cat* and *car* as solutions.

3.4 The Baire distance metric

The Baire distance metric [16] (also referred to as Generalized Cantor distance metric) is an ultrametric[6] on the set of strings of length m that is defined as $B(S, S') = 0$, if $S = S'$, and $B(S, S') = \alpha^r$, otherwise, where $\alpha \in]0, 1[$ is a parameter to be fixed and r is the first position at which S and S' differ.

Like the discrete distance metric, the betweenness relation induced by the Baire distance metric is the minimal betweenness relation. This is actually a common result for all ultrametrics. Suppose that three pairwisely different elements x, y, z of the ultrametric space are such that $d(x, y) + d(y, z) = d(x, z)$. A contradiction then arises from the fact that $d(x, z) \leq \max(d(x, y), d(y, z))$ and being both $d(x, y)$ and $d(y, z)$ greater than zero.

Nevertheless, the fact that the induced betweenness relation is the minimal one (and thus does not bring any interesting semantics to the penalty) does not mean that penalty-based functions compatible with the betweenness relation induced by an ultrametric are not interesting. In this case, both the search for

[6] We recall that an ultametric on X is a distance metric on X satisfying that $d(x, z) \leq \max(d(x, y), d(y, z))$, for any $x, y, z \in X$.

the median string(s) and the search for the center string(s) are of interest. The former is easily characterizable when a small enough value of α is considered. In particular, if $\alpha < \frac{1}{n-1}$ (where n is the number of strings to be aggregated), the median string(s) is(are) obtained in an iterative manner by first computing the most frequent first element and eliminating all the strings with a different first element, subsequently, among the remaining strings computing the most frequent second element and eliminating all the strings with a different second element, and so on. The computation of the center string(s) is easier. Independently of the value of α, we compute the longest common prefix among all strings to be aggregated. Any string starting with this prefix is a center string.

Example 5. Consider again the list of strings of length 3 on the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$: $\mathbf{S} = \{cat, dog, cat, dot, cog\}$. The most common first element is c, thus we consider the strings *cat*, *cat* and *cog*. Among those three strings, the most common second element is a, thus we consider the strings *cat* and *cat*. Among those two strings, the most common third element is t. Thus, the median string is *cat*. Since there is no common prefix among all strings in $\mathbf{S}$, all strings in Σ_3 are center strings.

4 Strings of different length

Given an alphabet Σ, we denote by Σ_* the set of all possible strings of any length $m \in \mathbb{N}$, i.e., $\Sigma_* = \cup_{i=0}^{\infty} \Sigma_i$ ($\Sigma_0 = \{\emptyset\}$).

An edit operation is a basic change that allows to transform one string into another. Typically, edit operations are related to spelling errors. According to Damerau [17], the most common spelling errors (and thus examples of edit operations) are addition of a single character, removal of a single character, substitution of a single character and transposition of two consecutive characters, which amount to more than 95 percent of the spelling errors found in different texts. Other possible edit operations could include inversion of the whole string or, when dealing with proper words, substitution by a homophonous word.

Edit distance metrics [4] are defined relative to a set of edit operations $\mathbb{E}$ and a cost function $c : \mathbb{E} \to [0, \infty[$:

$$E_{\mathbb{E},c}(S, S') = \min_{\mathbf{E} \in \mathbb{E}(S,S')} \sum_{e \in \mathbf{E}} c(e) \,,$$

where $\mathbb{E}(S, S')$ represents the set of all lists of edit operations in $\mathbb{E}$ that turn S into S'. Undoubtedly, the most common edit distance – which is almost considered a standard – is the Levenshtein distance metric [6] (just denoted by E, without subscripts). For the Levenshtein distance metric, the set of edit operations is formed by addition of a single character, removal of a single character and substitution of a character into another one, and the cost function is any constant function (e.g., $c(e) = 1$, for any $e \in \mathbb{E}$). Note that the Hamming distance metric is a special case of edit distance metric in which the unique edit operation allowed is substitution.

The betweenness relation B_E on Σ_* (illustrated in Figure 4) is quite interesting for error detection. Note that we use two different strings compared to the previous cases, otherwise we would obtain the same diagram as displayed in Figure 2.

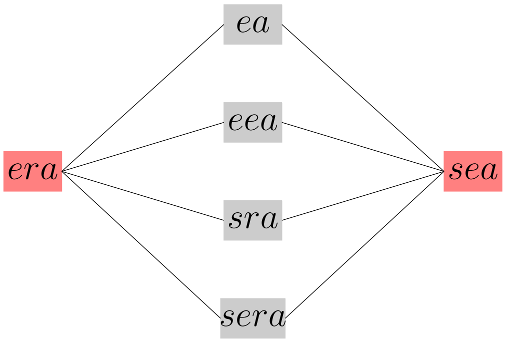

Fig. 4. Illustration of the strings (in grey) that are in between the strings *era* and *sea* (in red) according to the betweenness relation B_E for the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$.

Obtaining the median string(s) and the center string(s) with respect to the Levenshtein distance metric is known to be an NP-complete problem [3]. Some approximation techniques have been proposed. For instance, Kohonen [18] proposed to compute the set median string (or medoid string), which is straightforward, and then proceeding in a hill-climbing-like style of making small changes until (hopefully) the median is found.

Example 6. Consider the list of strings on the English alphabet $\Sigma = \{a, b, c, \ldots, z\}$: $\mathbf{S} = \{era, sea\}$. Note that any string S such that $(era, S, sea) \in B_E$ is a median string. This means that all among era, ea, eea, sra, $sera$ and sea are median strings. Obviously, only era and sea are medoid strings. Finally, center strings are only ea, eea, sra and $sera$.

5 Conclusions

In this paper, we have discussed the problem of finding median and center strings for five popular distance metrics on the set of strings within the framework of penalty-based data aggregation. Some other popular distance metrics such as the Damerau-Levenshtein distance metric (in which all edit operations proposed by Damerau [17] instead of those proposed by Levenshtein [6] are considered), the Jaccard distance metrix (based on the Jaccard index [19]), the Jaro distance metric [20] and the Jaro-Winkler distance metric [21] have also been used for measuring distances between strings – thus they could potentially be considered for searching for median and center strings. Future research concerns a more in-depth study of the aggregation of strings in which all these different distance metrics are analysed. The study of centroid strings and an analysis of the property of monotonicity from a betweenness-based (instead of order-based) perspective are also highlighted as interesting future research subjects.

Acknowledgments

Raúl Pérez-Fernández acknowledges the support of the Research Foundation of Flanders (FWO17/PDO/160) and the Spanish MINECO (TIN2017-87600-P).

References

1. Lanctot, J.K., Li, M., Ma, B., Wang, S., Zhang, L.: Distinguishing string selection problems. Information and Computation **185** (2003) 41–55
2. Gusfield, D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge (1997)
3. Nicolas, F., Rivals, E.: Complexities of the centre and median string problems. In: Proceedings of the 14th Annual Conference on Combinatorial Pattern Matching, Berlin, Heidelberg, Springer-Verlag (2003) 315–327
4. Gagolewski, M.: Data Fusion. Theory, Methods and Applications. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2015)
5. Hamming, R.W.: Error detecting and error correcting codes. The Bell System Technical Journal **29**(2) (1950) 147–160
6. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady **10**(8) (1966) 707–710
7. Pérez-Fernández, R., De Baets, B.: On the role of monometrics in penalty-based data aggregation. IEEE Transactions on Fuzzy Systems, in press, DOI: 10.1109/TFUZZ.2018.2880716
8. Yager, R.R.: Toward a general theory of information aggregation. Information Sciences **68** (1993) 191–206
9. Calvo, T., Beliakov, G.: Aggregation functions based on penalties. Fuzzy Sets and Systems **161** (2010) 1420–1436

10. Bustince, H., Beliakov, G., Dimuro, G.P., Bedregal, B., Mesiar, R.: On the definition of penalty functions in data aggregation. Fuzzy Sets and Systems **323** (2017) 1–18
11. Owen, S.H., Daskin, M.S.: Strategic facility location: A review. European Journal of Operational Research **111** (1998) 423–447
12. Li, M., Ma, B., Wang, L.: On the closest string and substring problems. Journal of the ACM **49**(2) (2002) 157–171
13. Ma, B., Sun, X.: More efficient algorithms for closest string and substring problems. In Vingron, M., Wong, L., eds.: Research in Computational Molecular Biology, Berlin, Heidelberg, Springer Berlin Heidelberg (2008) 396–409
14. Fishburn, P.C.: Lexicographic orders, utilities and decision rules: A survey. Management Science **20**(11) (1974) 1442–1471
15. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, New York (2009)
16. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer-Verlag Berlin Heidelberg, Berlin (2009)
17. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM **7**(3) (1964) 171–176
18. Kohonen, T.: Median strings. Pattern Recognition Letters **3** (1985) 309–313
19. Jaccard, P.: The distribution of the flora in the Alpine zone. The New Phytologist **11**(2) (1912) 37–50
20. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association **84** (1989) 414–420
21. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: Proceedings of the Section on Survey Research Methods of the American Statistical Association. (1990) 354–359