Evolutionary Weighted Mean Based Framework for Generalized Median Computation with Application to Strings

Lucas Franek and Xiaoyi Jiang

Department of Mathematics and Computer Science
University of Münster, Germany
{lucas.franek,xjiang}@uni-muenster.de

Abstract. A new general framework for generalized median approximation is proposed based on the concept of weighted mean of a pair of objects. It can be easily adopted for different application domains like strings, graphs or clusterings, among others. The framework is validated for strings showing its superiority over the state-of-the-art.

1 Introduction

The concept of median is widely used in order to estimate a single representative of a set of objects. Another motivation of the median concept is to eliminate some erroneous objects by averaging over all objects. Further, the median concept is also motivated by the results received from supervised classifier combination: It is well known that by averaging the results of several classifiers a more reliable classification can be achieved [9].

While finding the Euclidean median in vector space was originally posed by Fermat in the 17th century and is referred to as the Fermat-Weber problem, in the last years the median problem was also formulated for more general spaces and objects like strings [7], graphs [3], clusterings [12], and segmentations [11], among others. In most cases, however, the computation of generalized median turns out to be very demanding, partly even of $\mathcal{NP}$-completeness [10,12]. This fact motivates the design of approximate approaches.

There is very little work on general frameworks for generalized median approximation. The embedding approach is based on embedding the objects into the vector space, in which the Weiszfeld algorithm [14] can be applied to find the median point. Then, an inverse transformation to the original object domain is performed by using the weighted mean of a pair of objects (to be discussed later). This framework has been adopted to strings [7] and graphs [3]. However, the transformation in vector space and back into object (string, graph, etc.) domain is not trivial and such an embedding may cause undesired distortions.

In this work we propose a new framework for generalized median approximation. It is formulated for objects in general spaces and can be adopted to different application domains, such as strings, graphs, and clusterings, among

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 70–78, 2012.

others. The proposed framework is motivated by the lower bound for the generalized median [6]. It is observed that in case of a tight lower bound generalized median is received by computing the weighted mean of a pair of objects. This motivates us to formulate an algorithm for generalized median approximation by using the concept of weighted mean. The definition of weighted mean is directly motivated by the weighted mean of two numbers (or vectors) and it has already been adopted to the domain of strings [2], graphs [1] and clusterings [5].

In the experimental part of this work the proposed framework is adopted to the domain of strings. Further, a comparison with the embedded based approach [7] is provided.

The rest of the paper is organized as follows. In the next section we introduce the fundamentals of this work (formal definition of the generalized median problem and weighted mean). Our new general framework for generalized median approximation is presented in Section 3. The application to the domain of strings and experiments are shown in Section 4. We conclude in Section 5.

2 Fundamentals

We first define the problem of generalized median formally for a set of general objects.

Definition 1. *Let $X = \{x_1, \ldots, x_n\}$ be a set of objects in a general space U and $d : U \times U \to \mathbb{R}_0^+$ a distance function defined on U. Then, the generalized median $\hat{x}$ is defined by*

$$\hat{x} \;=\; \arg\min_{x \in U} \sum_{i=1}^{n} d(x, x_i) \;=\; \arg\min_{x \in U} SOD(x), \tag{1}$$

where the summation will be called sum of distances (SOD) of object x.

It intends to infer a representative sample out of the ensemble X. If the minimizer $\hat{x}$ is restricted to be within the ensemble X, then the corresponding solution is called set median.

Further, for the proposed algorithm we need the concept of weighted mean of a pair of objects. Consider two points in the n-dimensional real space, $x, y \in \mathbb{R}^n$. The weighted mean of x and y is defined as

$$z \;=\; \alpha x + (1 - \alpha) y,\; 0 \leq \alpha \leq 1. \tag{2}$$

If $\alpha = \frac{1}{2}$, then z is the (normal) mean of x and y. Clearly, z is a point on the line segment between x and y and the distance between z to x and y is controlled by the parameter α.

Generally, the weighted mean of two objects can be defined as follows.

Definition 2. *Let x_1 and x_2 denote two objects in a space U of all objects, and $d : U \times U \to \mathbb{R}_0^+$ a distance function defined on U which measures the*

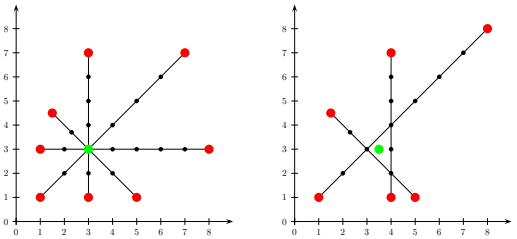

Fig. 1. Generalized median in two-dimensional vector space. Red points: initial points. Small black points: weighted means. Green point: generalized median. (a) Generalized median is located at the intersection point of the line segments of opposite point pairs. (b) Line segments do not intersect in one point. Further iterations are necessary to approach the generalized median.

dissimilarity of two objects. The weighted mean of x_1 *and* x_2 *is an object* x_w *such that*

$$d(x_1, x_w) = \alpha \tag{3}$$
$$d(x_1, x_2) = \alpha + d(x_w, x_2) \tag{4}$$

where α *is a constant with* $0 \leq \alpha \leq d(x_1, x_2)$.

The concept of weighted mean has been brought to pattern recognition for strings [2], graphs [1], and clusterings [5].

3 Evolutionary Weighted Mean Based Framework for Generalized Median Computation

In order to motivate our new method for generalized median approximation let us first consider the lower bound for the generalized median [6]. Let $\mathcal{P}$ denote a partition of the objects X into $m = \frac{n}{2}$ pairs (n even for convenience):

$$\mathcal{P} = \{(x_{11}, x_{12}), (x_{21}, x_{22}), \ldots, (x_{m1}, x_{m2})\}, \quad x_{i,j} \in X, \bigcup_{i=1,\ldots,m, j=1,2} \{x_{ij}\} = X.$$

Further, $\mathfrak{P}$ denotes the set of all such partitions $\mathcal{P}$. If d is a metric, then it can be shown that a lower bound Γ on the SOD of the generalized median $\hat{x}$, i.e. $0 \leq \Gamma \leq SOD(\hat{x})$, can be computed by estimating the optimal set of pairs $\hat{\mathcal{P}} \in \mathfrak{P}$ such that the sum of distances of the pairs (x_{i1}, x_{i2}) is maximal (see [6] for a proof):

$$\Gamma \;=\; \max_{\mathcal{P} \in \mathfrak{P}} \sum_{i=1}^{m} d\,(x_{i1}, x_{i2})\,. \tag{5}$$

This lower bound formulation motivates an approximation algorithm for the generalized median. By approaching the lower bound the generalized median is

obviously also approached. In the ideal case, where the lower bound is tight, i.e. $SOD(\hat{x}) = \Gamma$, even the true generalized median could be found by approaching Γ. In this case it directly follows from the metric property $d(x_{i1}, \hat{x}) + d(\hat{x}, x_{i2}) = d(x_{i1}, x_{i2})$, i.e. the generalized median is the weighted mean of each pair $(x_{i1}, x_{i2}) \in \hat{\mathcal{P}}$. Since the lower bound is estimated by an optimum partition of pairs, the generalized median is approached by computing the weighted means of such pairs of objects.

For each pair the optimum weight is unknown a priori. But a condition for optimality is obvious: The optimum weighted means of all pairs are estimated such that they match in one point, namely the generalized median.

The idea is depicted in Fig. 1 for a set of points. From a geometrical point of view, the median point in an Euclidean space is ideally located in the exact intersection point between the opposite pairs of points (Fig. 1(a)). Thus, the generalized median is received by computing for each of the pairs a set of weighted means (represented by the smaller points on the line segments). The generalized median is then located at the point where the weighted means match. Note that in general the lower bound will not be tight. To say it another way, in geometric space the weighted means of all pairs will not match in one point as depicted in Fig. 1(b). Thus, we resort to an iterative algorithm:

Step 1: Compute the optimum (opposite) pairs of objects.
Step 2: Estimate for each pair the optimum weighted mean in terms of SOD and add it to the current set of objects.
Step 3: Select the optimum objects in the current set of objects.

These steps are detailed in the following.

Step 1: Optimum Pairs of Objects

The question arises how to estimate the optimum set of pairs of objects such that the sum of distances of pairs (Eq. (5)) is maximized. To handle this problem it is proposed to build a graph, where each object corresponds to a vertex and each edge between two vertices is weighted by the distance between the corresponding objects. Then, finding the optimum pairs is equipollent to solving the maximum weighted graph matching problem. A solution to this problem is provided by [8]. Note that in the situation of an odd number of input objects one vertex remains unmatched (see also [6]). The unmatched point is stored in the current set of objects which is processed in the third step.

Step 2: Optimum Weighted Mean in Terms of SOD

After having computed the optimum set of pairs the weighted mean for each pair of objects is computed. In this situation, however, the problem arises that the weight may vary for each pair in the range $[0, 1]$ (after normalization) and the optimum weight yielding the generalized median is unknown a priori. To find the optimum weight a search procedure is applied. First, several weighted

means are computed and in the next step it is decided which weighted means are suitable and which are not, i.e. a kind of fitness function is needed. Considering that the generalized median aims at minimizing the SOD in Eq. (1) it is proposed to use this SOD as fitness function. To estimate the best weight α the whole range is sampled a fixed number of times in an equidistant way. The weighted means are evaluated by the SOD and the weighted mean with the lowest SOD is selected. Note that this kind of search procedure is a linear search. Further search procedures could be also adopted. Optionally, the two weighted means with the lowest SOD may be selected. This is discussed in the next step.

Step 3: Selecting the Optimum Set of Objects

In an ideal situation the optimum weighted mean for each pair of objects in terms of SOD (as computed in Step 2) would be equal to the generalized median of the initial set. In this case the algorithm should terminate. In a non-ideal situation, however, it cannot be expected that the computed optimum weighted means of different pairs are equal. Probably, some weighted means will be more suitable than other ones. Again, it is proposed to resort to the SOD as a fitness function in order to distinguish between suitable weighted means and less suitable ones.

More specifically, in the proposed algorithm the optimum weighted mean for each pair of objects is added to the current set of objects. Then, the optimum set of objects is estimated by selecting the best n_{max} objects from the current set of objects, where n_{max} is a parameter of the algorithm. Hereby, the best objects are again selected by evaluating their SOD with respect to the input set of objects and selecting the objects with the lowest SOD. The parameter n_{max} is fixed such that the size of the set of optimum objects is limited during the iteration process. Note that optionally, the second best weighted mean from Step 2 may also be added to the set of objects because in a non-ideal case it may contain valuable information as well.

The process is now iterated beginning with the first step using the current optimum set of objects. The algorithm may finish when either the lower bound is reached or when it converges to some solution.

Evolutionary Weighted Mean Based Framework

The proposed framework can now be formulated as follows. Given a set of objects $O = \{o_1, \ldots, o_n\}$.

1. Consider all pairs of objects and compute their weights by the distance between the corresponding objects. Save them into the distance matrix D.
2. Determine the optimal set of pairs using maximum weighted graph matching on D. Let $E = (e_1, \ldots, e_m)$ denote the corresponding optimum set of edges.
3. For each edge $e_i \in E$ consider the corresponding pair (o_{i1}, o_{i2}) and:
   (a) Compute w weighted means by using $\alpha = \frac{i \cdot d(o_{i1}, o_{i2})}{w+1}, i = 1, \ldots, w$.
   (b) Evaluate the w weighted means by the fitness function SOD and select the best weighted mean o^*. Update the current set of objects by adding the best weighted mean: $O = O \cup \{o^*\}$.

4. Evaluate all objects in O by SOD and delete the worst instances such that the resulting current set O consists of a maximum number $n_{\max}$ of objects.
5. It is checked if the lower bound is matched or if convergence is achieved. Otherwise, the procedure starts from step 1 with the current O. A maximum number of iterations I_{max} prevents the algorithm from getting inefficient.

Obviously, the algorithm is convergent, because the values of the fitness function (SOD) can only decrease. However, I_{max} is introduced in the last step for efficiency reasons.

It is emphasized that the only requirement for this framework is that the weighted mean is well defined for the space under consideration. In the next section this framework will be adopted and validated for the median string problem.

4 Application to Strings

Strings are a fundamental representation in structural pattern recognition. Here, our framework is adopted to the domain of strings. As a comparison method the embedding based generalized median computation proposed by Jiang et al. [7] will be used, which has been demonstrated to outperform the related algorithms from the literature.

In order to be able to apply our proposed approach to the domain of strings two requirements have to be fulfilled. First, a suitable distance function is needed in order to compare strings. Secondly, based on this distance function the weighted mean has to be defined. Here, we use the popular Levenshtein edit distance [13].

The weighted mean of a pair of strings $(\mathfrak{S}_1, \mathfrak{S}_2)$ for the edit distance was introduced in [2]. It is defined analogously to Definition 2 as a string $\mathfrak{S}_w$ with

$$d(\mathfrak{S}_1,\ \mathfrak{S}_w) = \alpha,\ d(\mathfrak{S}_1, \mathfrak{S}_2) = \alpha + d(\mathfrak{S}_w, \mathfrak{S}_2),\ 0 \leq \alpha \leq d(\mathfrak{S}_1, \mathfrak{S}_2).$$

$\mathfrak{S}_w$ is constructed by selecting a subsequence of all edit operations used for transforming $\mathfrak{S}_1$ into $\mathfrak{S}_2$, such that $d(\mathfrak{S}_1, \mathfrak{S}_\mathfrak{w}) = \alpha$. Applying this subsequence to $\mathfrak{S}_1$ yields $\mathfrak{S}_w$.

The evolutionary weighted mean algorithm can now be directly applied to the domain of strings. In the implementation we set $w = 3$ and $n_{max} = 10$, i.e. the set of strings will consist of a maximum number of 10 strings after the first iteration. Further, the iteration is stopped after $I_{max} = 5$ iterations.

4.1 Experimental Settings

In order to be able to compare the proposed approach with the embedding based approach the experimental settings from [7] are used. A synthetic dataset is generated for test purpose by distorting an initial string p times. Hereby, each symbol of the initial string is distorted with a fixed probability $p_{distort}$. If a symbol is distorted, then the three elementary operations substitution, deletion, and insertion are chosen by a fixed probability. Five strings (Scotland, Birmingham, Philadelphia, TristanDaCunha, WesternPatagonia) are used. 100 datasets are generated for each initial string and the average performance measures are reported. The same parameter values as in [7] are used:

- $p = 40$: Number of strings in the initial set $\mathcal{S}$.
- $p_{distort} = 12\%$: Distortion probability for each symbol.
- The probabilities of the three basic operations: $p_{substitute} = 87\%$, $p_{delete} = 9\%$, $p_{insert} = 4\%$.
- $c(s \to \bar{s}) = c(\epsilon \to s) = c(s \to \epsilon) = 1$: Equal costs for the edit operations.

We also summarize some settings of the embedding based method [7]. It uses prototype selection in order to reduce computational efforts. In our experiments the K-medians prototype selector is used as it was proposed in [7]. The number of prototypes is chosen to be 25% of the original set size. A comparison with all variants of the embedding based approach (concerning the inverse transformation to the original object domain) is provided, namely linear, triangulation, and recursive. Additionally, set median is also included into the comparison.

4.2 Experimental Results

In order to evaluate the obtained median strings it has to be taken into account that both the generalized median $\hat{\mathfrak{S}}$ and its $\mathrm{SOD}(\hat{\mathfrak{S}})$ are unknown. Consequently, we have to resort to the lower bound Γ (see Eq. (5)). Then, the quality of the obtained median $\bar{\mathfrak{S}}$ is evaluated by $\Delta = \mathrm{SOD}(\bar{\mathfrak{S}})/\Gamma$. If $\Delta \approx 1$, it is a strong hint that $\bar{\mathfrak{S}}$ is an accurate approximation of the generalized median.

Probability of symbol distortion. In the first experiment the robustness against distortions in the input strings is investigated (see Fig. 2). The distortion probability is varied ($p_{distort} \in [2, 50]$). As observed in [7], our results confirm that the recursive approach dominates the other embedding based approaches up to 20%. For higher distortion probabilities the deviation Δ increases. The proposed evolutionary weighted mean algorithm clearly outperforms all variants for all distortion probabilities. Even for a distortion probability of 35% the deviation Δ is less than 1.05, making our method suitable also for high distortion levels.

Number of strings. Now we study the algorithm behavior with respect to the number of strings in the initial set of strings, i.e. we vary $p = 10, 12, \ldots, 40$. The results are shown in Fig. 3. While for example the recursive approach needs 40 initial strings in order to yield a deviation $\Delta \leq 1.05$ the evolutionary weighted mean algorithm performs very well already for more than 10 input strings with a deviation $\Delta \leq 1.01$.

Length of initial string. In Fig. 4 the performance is plotted for different string lengths. The evolutionary weighted mean algorithm clearly outperforms the embedding based approach for all string lengths. The SOD of the obtained median is very close to the lower bound indicating that the obtained median of the proposed algorithm is very close to the generalized median.

Time complexity. The time complexity for one iteration depends on the number of weighted means computed for one pair of strings as well as on the maximum

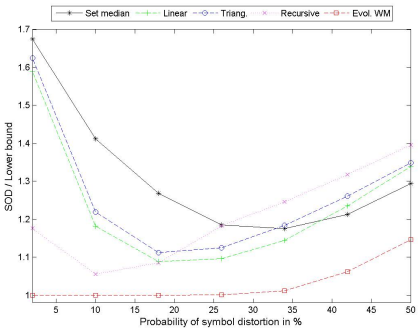

Fig. 2. Performance as a function of distortion probability

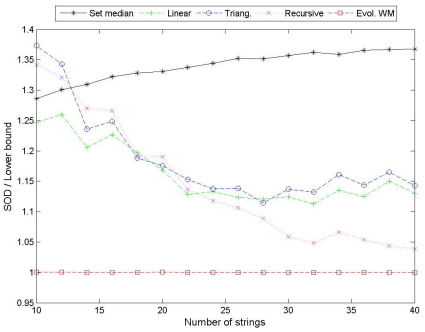

Fig. 3. Performance as a function of number of strings

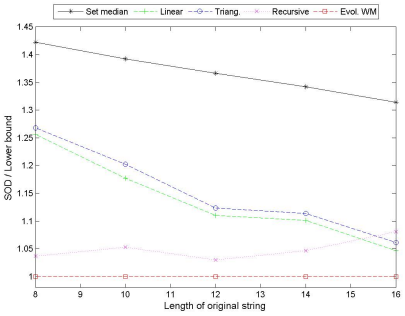

Fig. 4. Performance as a function of string length

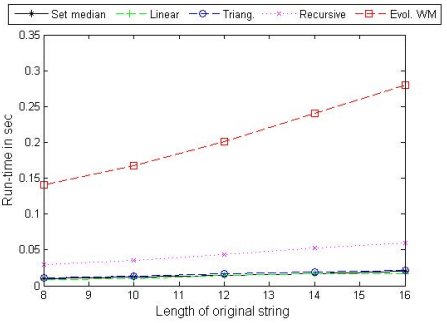

Fig. 5. Computational time for test series "String length"

number of pairs in the current set of objects, i.e. $\mathcal{O}(w \cdot \frac{n_{max}}{2})$. Consequently, for a maximum number of iterations I_{max} the overall time complexity is $\mathcal{O}(I_{max} \cdot w \cdot \frac{n_{max}}{2})$. The computational time was measured on an Intel Core i7 2.80 GHz with 6 GB RAM. The result with respect to the string length is shown in Fig. 5. Note that the proposed approach is slightly more complex than the embedding based approach because of a higher number of necessary weighted mean computations. Nevertheless, the computational time is absolutely negligible (less than half a second in all cases).

Discussion. The performed experiments have shown that the evolutionary algorithm clearly outperforms the embedding based approach in all cases. It can handle high distortion levels very well and it works also quite well for a small number of initial input strings, whereas for example the embedding based recursive approach needs a significantly higher number of input strings in order to yield comparable results. Moreover, the results have shown that in many cases the performance is less than 2-3% compared to the lower bound, indicating that the obtained result is very close to the unknown generalized median.

5 Conclusion

A new algorithm for generalized median computation was formulated based on the concept of weighted mean. Experimental results were shown for strings. The proposed algorithm clearly outperforms the embedding based approach (and thus the related algorithms from the literature). The main advantage of the proposed framework is that it can be easily adopted for every application domain, in which the weighted mean is defined. Recently, the framework was adopted for ensemble clustering and its superiority with respect to several state-of-the-art ensemble clustering methods has been shown [4]. In future we will consider the application of the framework to further domains such as graphs and image segmentation.

References

1. Bunke, H., Günter, S.: Weighted mean of a pair of graphs. Computing 67(3), 209–224 (2001)
2. Bunke, H., Jiang, X., Abegglen, K., Kandel, A.: On the weighted mean of a pair of strings. Pattern Anal. Appl. 5(1), 23–30 (2002)
3. Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., Bunke, H.: Generalized median graph computation by means of graph embedding in vector spaces. Pattern Recognition 43(4), 1642–1655 (2010)
4. Franek, L.: Ensemble Algorithms with Applications to Clustering and Image Segmentation. Ph.D. thesis, University of Münster (2012)
5. Franek, L., Jiang, X.: Weighted mean of a pair of clusterings. Pattern Anal. Appl. (under revision)
6. Jiang, X., Münger, A., Bunke, H.: On median graphs: Properties, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. 23(10), 1144–1151 (2001)
7. Jiang, X., Wentker, J., Ferrer, M.: Generalized median string computation by means of string embedding in vector spaces. Pattern Recognition Letters 33(7), 842–852 (2012)
8. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society of Industrial and Applied Mathematics 5(1), 32–38 (1957)
9. Rokach, L.: Pattern classification using ensemble methods. World Scientific Pub. Co. Inc. (2010)
10. Sim, J.S., Park, K.: The consensus string problem for a metric is NP-complete. J. Discrete Algorithms 1(1), 111–117 (2003)
11. Singh, V., Mukherjee, L., Peng, J., Xu, J.: Ensemble clustering using semidefinite programming with applications. Mach. Learn. 79(1-2), 177–200 (2010)
12. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. Int. J. Pattern Recognition and Artificial Intelligence 25(3), 337–372 (2011)
13. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM 21(1), 168–173 (1974)
14. Weiszfeld, E., Plastria, F.: On the point for which the sum of the distances to n given points is minimum. Annals of Operations Research 167, 7–41 (2009)