Information Processing Letters 110 (2010) 420–423

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Reducing non-determinism of k-NN searching in non-ordered discrete data spaces

Dashiell Kolbe [a,*], Qiang Zhu [b], Sakti Pramanik [a]

[a] Michigan State University, East Lansing, MI, United States
[b] University of Michigan - Dearborn, Dearborn, MI, United States

ARTICLE INFO

Article history:
Received 9 September 2009
Received in revised form 5 March 2010
Accepted 16 March 2010
Available online 21 March 2010
Communicated by S.E. Hambrusch

Keywords:
Algorithms
Databases
Information retrieval
Non-ordered discrete data space
k-nearest neighbor search

ABSTRACT

We propose a generalized version of the Granularity-Enhanced Hamming (GEH) distance for use in k-NN queries in non-ordered discrete data spaces (NDDS). The use of the GEH distance metric improves search semantics by reducing the degree of non-determinism of k-NN queries in NDDSs. The generalized form presented here enables the GEH distance to be used for a much greater variety of scenarios than was possible with the original form.

1. Introduction

Nearest neighbor searches/queries in Non-Ordered Discrete Data Spaces (NDDS) are becoming increasingly useful in areas such as bioinformatics, multimedia, text retrieval, audio and video compression, data-mining, and E-commerce. This form of searching may be stated as the following problem: given a set S of n data points in an NDDS X and a query point $q \in X$, return a set of $k > 0$ objects $A \subseteq X$ where $\forall_{u \in A, v \in S-A}$: $D(q,u) \leqslant D(q,v)$ and $|A| = k$. Examples of such queries are "finding the k closest restaurants to the intersection of Fifth and Main", and "finding the k fixed length words that differ the least from the word 'near'."

Numerous techniques have been proposed in the literature to support efficient nearest neighbor queries in continuous (ordered) data spaces (CDS) and general metric spaces. Excellent surveys of both the theoretical and practical aspects of nearest neighbor searching in such spaces have been presented by Chávez et al. [1] and Hjaltason and Samet [2] respectively. Little work has been reported on supporting efficient similarity searches in NDDSs.

A major problem with k-NN searching in NDDSs is the non-determinism of the solution. That is, there is usually a large number of candidate solutions available which may obscure the result. This is mainly caused by the coarse granularity of the commonly used distance metric, the Hamming distance. An extension to the Hamming distance, termed the Granularity-Enhanced Hamming (GEH) distance, was introduced in [4] as a solution to this problem. We demonstrated that the GEH distance greatly reduced the non-determinism of the solution set, as well as provided performance benefits, while maintaining the semantics of the original Hamming distance [4,5]. However, the GEH distance introduced in [4] was tied directly to data point frequency values in a manner that may not be ideal in all scenarios. Applications/scenarios with other more relevant dataset characteristics (distribution, clustering, etc.) may not experience the same performance benefits seen in [4].

To address this issue, we introduce a generalized form of the GEH distance in this paper. This form may be opti-

* Corresponding author.
E-mail addresses: kolbedas@msu.edu (D. Kolbe), qzhu@umich.edu (Q. Zhu), pramanik@cse.msu.edu (S. Pramanik).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.03.013

mized to a much broader set of applications than the original GEH distance presented in [4]. Conditions/constraints are presented that maintain the necessary distance metric properties to be used as a pruning metric while still preserving the semantics of the original Hamming distance. We show that the original GEH distance is, in fact, an instantiation of this generalized form. Further, we present a new instantiation of the generalized GEH form that demonstrates the benefits of adapting the generalized form for specific scenarios.

The rest of this paper is organized as follows. Section 2 presents some general properties of NDDSs that are necessary for the remaining discussion. Section 3 presents the generalized form of the GEH distance. Section 4 introduces a new ranking based GEH instantiation derived from the generalized form.

2. Non-ordered discrete data space

In this section, we provide a more thorough description of an NDDS. A discrete space is based upon the concept of all objects in the universe of discourse X being discrete and unordered along each dimensional domain $I^d(X)$. Ordered discrete objects typically demonstrate the same properties as CDS objects and thus are not considered here. Examples of such non-ordered discrete data are multimedia objects, profession, gender, bioinformatics, and user-defined types. Each of these examples may be represented as a feature vector in a d-dimensional space. Consider, in genome sequence databases such as GenBank, sequences with alphabet $A = \{a, g, t, c\}$ are broken into substrings of fixed-length d for similarity searches [3,6]. Each substring can be considered as a vector in d-dimensional space. For example, substring *aggctttgcaaggctttgcagcact* is a vector in the 25-dimensional data space, where the ith character is a letter chosen from alphabet A in the ith dimension. In this example, 'a' is no closer to 'c' than it is to 't' and so forth. Thus, mapping a discrete space into a continuous space by applying a form of ordering changes the semantics of the dataset. Formal definitions of an NDDS have been presented in [7,5].

The inability to be ordered along an axis renders standard forms of distance measurement, such as Euclidean or Manhattan, inapplicable. In turn, a common way of calculating the distance between two discrete objects is to apply the Hamming measurement. Essentially, this measurement represents the number of dimensions between two d-dimensional vectors that contain different elements. This is described formally as follows:

$$D_{Hamm}(\alpha, \beta) = \sum_{i=1}^{d} \left\{ \begin{array}{ll} 0 & \text{if } \alpha[i] = \beta[i] \\ 1 & \text{otherwise} \end{array} \right\}. \tag{1}$$

This distance is useful in discrete spaces due to its non-reliance upon dataset semantics, particularly for equality measurements. However, its usefulness declines rapidly when applied to other operations, such as grouping, due to its limited cardinality. The cardinality of an NDDS for a d-dimensional dataset X with an alphabet size $|A_i|$ for each dimension i in d, is calculated as the product of the alphabet sizes from each dimension, or $\prod_{i=1}^{d} |A_i|$. Using the aforementioned genome sequence example, a 25-dimensional dataset with an alphabet size of 4 for each dimension would have a cardinality of 1,125,889,906,842,624: that is, there are over 10^{15} possible distinct objects in the dataset. However, if the Hamming distance formula is used to calculate the distance between the objects, there are only $d+1$ (26) different possible distances between any two objects. As presented in our earlier work, this leads to a large degree of non-determinism which can obscure the result of k-NN searches. For example, in [5] we demonstrated that when searching a 10-dimensional NDDS dataset of $2M$ vectors for 10 neighbors, there were over $45M$ possible solutions when using the Hamming distance formula.

To address this issue, we introduced the Granularity-Enhanced Hamming (GEH) distance in [4,5]. This distance expanded upon the Hamming distance to provide more granularity while maintaining all of the semantics of the Hamming distance. This was accomplished by adding an adjustment value to the Hamming distance between two vectors based upon their matching elements. The form proposed in [4,5] is as follows:

$$D_{GEH}(\alpha, \beta) = \sum_{i=1}^{d} \left\{ \begin{array}{ll} 1 & \text{if } \alpha[i] \neq \beta[i] \\ \frac{1}{d} f(\alpha[i]) & \text{otherwise} \end{array} \right\}, \tag{2}$$

where

$$f\big(\alpha[i]\big) = 1 - f_g\big(\alpha[i]\big).$$

The value of $f_g(\alpha[i])$ is the number of occurrences of $\alpha[i]$ in the ith dimension of the dataset, divided by the number of vectors in the dataset; essentially, a global frequency value. While this does provide a dramatic increase in the determinism of result sets when used in a similarity search, this distance metric may not provide an ideal distance semantic for all applications. Eq. (2) is limited to applications where the global frequency of elements has some significance in the dataset. Applications where other dataset characteristics provide a better semantic may not be able to benefit from using Eq. (2) to the same degree as the results shown in [4]. To address this issue, we propose a generalization of the GEH distance that may be optimized to a much broader set of applications.

3. Generalized GEH distance

We observe that the Hamming distance assumes that a worst case match (i.e. a non-match) between two elements is represented by a distance of 1, while all other matches are represented by a distance of 0. We expand upon this by adding more granularity to the values assigned to different types of matches. We propose the following generalized form of the GEH distance to accomplish this goal:

$$D_{GEH}(\alpha, \beta) = D_{Hamm}(\alpha, \beta) + \frac{1}{C} \sum_{i=1}^{d} f_{geh}\big(\alpha[i], \beta[i]\big), \tag{3}$$

where

Constraint 1: $\forall_{\alpha,\beta}$: $C \geqslant d - D_{Hamm}(\alpha, \beta)$.

Constraint 2: $\forall_{\alpha[i],\beta[i]}$: $0 < f_{geh}(\alpha[i], \beta[i]) < 1$.

Constraint 3: $\forall_{\alpha[i],\beta[i]}$: $f_{geh}(\alpha[i], \beta[i]) = f_{geh}(\beta[i], \alpha[i])$.

Constraint 4: $\forall_{\alpha[i],\beta[i]}$: $\alpha[i] \neq \beta[i] \rightarrow f_{geh}(\alpha[i], \beta[i]) = 0$.

Here, f_{geh} represents some function, chosen by an application expert, that will provide an adjustment to the Hamming distance for each dimension. The variable C is a pseudo-constant[1] used to guarantee the adjustment values of f_{geh} do not become more dominant than the original Hamming distance. Constraint 1 indicates that the value of C must be greater than or equal to the number of matching dimensions between two vectors. Constraint 2 indicates that the result of function f_{geh} for the ith element of the two vectors being considered must be in the range of $(0, 1)$ non-inclusive. Constraint 3 indicates that function f_{geh} must be symmetric. Constraint 4 indicates that the result of f_{geh} for the ith elements of the two vectors being considered must equal 0 if the these two elements do not match.[2]

From Eq. (3), we can see that, if $m \leqslant D_{GEH}(\alpha, \beta) < m+1$ $(m = 0, 1, \ldots, d)$, then vectors α and β mis-match on m dimensions (i.e. match on $d - m$ dimensions), therefore preserving the original semantics of the Hamming distance.[3] Further, the four provided constraints allow the generalized GEH distance to maintain the metric properties necessary for use as a pruning metric in similarity searches as described in the following lemmas:

Lemma 1. *The generalized GEH distance maintains the Positiveness property (i.e.* $\forall_{x,y}$: $D_{GEH}(x, y) \geqslant 0$*).*

Proof. By maintaining the Hamming distance within the GEH distance, we are guaranteed a positive value if any elements between the two vectors do not match. Condition 2 indicates that all values resulting from matching elements will have non-negative values. □

Lemma 2. *The generalized GEH distance maintains the Strict Positiveness property (i.e.* $\forall_{x,y}$: $x \neq y \rightarrow D_{GEH}(x, y) > 0$*).*

Proof. This property is inherited by maintaining the Hamming distance within the generalized GEH distance, whereby any two vectors that are not equal will have a distance greater than '0' based upon a '1' being added to the distance for each non-matching dimension. Constraint 2 guarantees that the values added from function f_{geh} will all be non-negative. □

Lemma 3. *The generalized GEH distance maintains the Symmetry property (i.e.* $\forall_{x,y}$: $D_{GEH}(x, y) = D_{GEH}(y, x)$*).*

Proof. The Hamming distance is known to maintain symmetry between vectors. In addition, Constraint 3 guarantees that the values provided by the function f_{geh} will maintain symmetry as well. □

Lemma 4. *The generalized GEH distance maintains the Pseudo-Reflexivity property (i.e.* $\forall_{x,y}$: $D_{GEH}(x, x) < 1 \wedge x \neq y \rightarrow D_{GEH}(x, y) \geqslant 1$*).*[4]

Proof. This property is maintained due to Constraints 1 and 2, which stipulate that the additional value added to the Hamming distance will always be in the range of $(0, 1)$, non-inclusive. Thus the distance between two vectors that exactly match will have a distance value less than '1'. Any vectors that are different in one or more dimensions will have a distance greater than or equal to '1'. □

Lemma 5. *The generalized GEH distance possesses the Triangular Inequality property (i.e.* $\forall_{x,y,z}$: $D_{GEH}(x, y) + D_{GEH}(y, z) \geqslant D_{GEH}(x, z)$*).*

Proof. We first consider the Hamming portion of the generalized GEH distance. For any dimension $i \in [1, d]$, if $x_i \neq z_i$ then either $x_i \neq y_i \oplus z_i \neq y_i$ or $x_i \neq y_i \wedge z_i \neq y_i$. Thus for each dimension i where the right side of the inequality (i.e. $D_{GEH}(x, z)$) would be incremented by an integer value of '1', the left side of the inequality (i.e. $D_{GEH}(x, y) + D_{GEH}(y, z)$) would be incremented by an integer value of either '1' or '2', thus maintaining the inequality. Next, we consider the adjustment portion of the GEH distance (i.e. $\frac{1}{C} f_{geh}()$). For any dimension $i \in [1, d]$, if $x_i = z_i$ then either $x_i = y_i \wedge z_i = y_i$ or $x_i \neq y_i \wedge z_i \neq y_i$. Thus, due to Constraints 2 and 3, for all dimensions where this is the case, the left side will either be incremented by twice as much as the right side or be incremented by an integer value of '2' while the right side is incremented by some value less than '1'. Constraint 4 indicates that no additions will be made if the values in the dimension match, leaving the Hamming component to be dominant. Thus the adjustment values maintain the inequality. □

4. Ranking based GEH instantiation

As described in [1], many search algorithms demonstrate improved performance when the distances between data points are distributed evenly throughout the distance range. We note that the original GEH distance, Eq. (2), is likely to result in a heavily skewed distribution of possible distances.[5] As the alphabet size grows, the likely values

[1] The term *pseudo-constant* is used to indicate that C is not strictly a constant, and may vary as long as Constraint 1 of Eq. (3) is maintained.

[2] Note that both variables α_i and β_i are passed to f_{geh}. This enables f_{geh} to be fully expressed whereby Constraint 4 may be verified.

[3] Many application specific solutions such as BLOSUM, employed in bioinformatics, reduce the non-determinism of solution sets by utilizing a cost matrix as a direct form of distance measurement. This is similar in theory to utilizing f_{geh} as a distance measure directly. Unfortunately, these methods do not preserve the semantics of the original Hamming distance and thus lose a level of portability between application environments. However, solutions such as BLOSUM may be incorporated into Eq. (3) by utilizing the diagonal of the cost matrix for f_{geh}.

[4] Note that the traditional property of Reflexivity (i.e. $\forall_x$: $D(x, x) = 0$) is replaced by the property of Pseudo-Reflexivity. This is a reasonable substitution in an NDDS due to two vectors exactly matching each other still being identifiable from all other pairs of vectors based only upon the distance between them.

[5] Eq. (2) may be derived from Eq. (3), where $f_{geh}(\alpha[i], \beta[i]) = \{1 - f_g(\alpha[i])$ if $\alpha[i] = \beta[i]$, 0 otherwise$\}$ and $C = d$.

Table 1
Varying dimensionality.

	Hamm.	Freq.	Rank
$d = 5$	36	7	6
$d = 10$	968	472	480
$d = 15$	5914	4591	4675
$d = 20$	8294	8228	8232

Table 2
Varying zipf distribution.

	Hamm.	Freq.	Rank
zipf 0.0	968	472	480
zipf 0.5	693	399	301
zipf 1.0	381	233	126
zipf 1.5	105	73	30

of $f_g(\alpha[i])$ trend closer to '0' leading to a clumping of distance values close to the floor value. Additionally, setting $C = d$ results in C having a dominant role in the distance value as the dimensionality of the dataset grows larger. To address these issues, we propose a new GEH distance instantiation:

$$f_{geh}\big(\alpha[i], \beta[i]\big) = \frac{\text{rank}_i(\alpha[i])}{|A_i| + 1},$$
$$C = d - D_{Hamm}(\alpha, \beta) + 1. \tag{4}$$

Here, the term $\text{rank}_i(\alpha[i])$ indicates the global rank of element $\alpha[i]$ among the alphabet set in dimension i. The ranking mechanism employed should be set by an application expert on the condition that it results in the different elements of the alphabet receiving ranking values of $[1, |A|]$ inclusive. The value of C tracks to the number of matching dimensions between vectors α and β. As an example ranking mechanism, we consider the frequency of elements within a dimension, applying a higher rank (lower integer value) to elements that occur more frequently, and a lower rank (higher integer value) to elements that occur less frequently. For example, if the alphabet set in dimension i consists of $\{a, b, c\}$, where a appears in 20% of the vectors, b appears in 50% of the vectors, and c appears in 30% of the vectors in dimension i, the rank of each of the elements in dimension i would be as follows: $a \to 3$, $b \to 1$, and $c \to 2$. Although an element's frequency within a dimension still plays a role in the determination of the GEH distance (in this example), the ranking mechanism maintains a uniform distribution of distance values over varying alphabet sizes. Additionally, having the value of C track to the number of matching dimensions rather than the dimensionality of the dataset reduces the dominance of C as the dimensionality of the dataset grows larger.

To evaluate the benefits of an adaptable distance metric, we performed a series of k-NN queries utilizing the GEH distance implementations in Eqs. (2) and (4) as well as the Hamming distance. Table 1 shows a comparison of I/O results while searching uniformly distributed datasets of varying dimensionality. These results demonstrate a scenario where the frequency based GEH implementation provides slightly better search performance than the rank based GEH implementation. Further, our results agree with those in [8] linking a decreasing performance with increasing dimensionality.[6] Table 2 shows a comparison of the I/O results while searching 10-dimensional datasets of varying zipf distribution. For this scenario, use of the new ranking based GEH implementation provides a strong performance improvement over the frequency based GEH distance implementation. This is in agreement with the results shown in [1] concerning search performance and distance value distribution. These results highlight scenarios where Eqs. (2) and (4) provide search performance improvements specific to each case, thus demonstrating the benefits of an adaptable distance metric.

References

[1] E. Chávez, B. Navarro, R. Baeza-Yates, J. Marroquín, Searching in metric spaces, ACM Computing Surveys 33 (2001) 273–321.
[2] G. Hjaltason, H. Samet, Index-driven similarity search in metric spaces, ACM Transactions on Database Systems 28 (2003) 517–580.
[3] W.J. Kent, BLAT-the BLAST-like alignment tool, Genome Resources 12 (2002) 656–664.
[4] D. Kolbe, Q. Zhu, S. Pramanik, On k-nearest neighbor searching in non-ordered discrete data spaces, in: International Conference on Data Engineering, 2007, pp. 426–435.
[5] D. Kolbe, Q. Zhu, S. Pramanik, Efficient k-nearest neighbor searching in non-ordered discrete data spaces, ACM Transactions on Information Systems 28 (2) (2010).
[6] G. Qian, Principles and applications for supporting similarity queries in non-ordered-discrete and continuous data spaces, PhD thesis, Michigan State University, East Lansing, 2004.
[7] G. Qian, Q. Zhu, Q. Xue, S. Pramanik, Dynamic indexing for multidimensional non-ordered discrete data spaces using a data-partitioning approach, ACM Transactions on Database Systems 31 (2006) 439–484.
[8] R. Weber, H.-J. Schek, S. Blott, A Quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in: 24th International Conference on Very Large Data Bases, 1998, pp. 194–205.

[6] Note that for the largest dimensionality tested, $d = 20$, the I/O results when using both ranking based and frequency based GEH implementations begin to approach each other. We attribute this to the dimensionality of the dataset playing a less dominant role in Eq. (4) than in Eq. (2).