Approximate Search in Dissimilarity Spaces using GA

David Bernhauer
Faculty of Information Technology, Czech Technical University in Prague, Czech Republic
bernhdav@fit.cvut.cz

Tomáš Skopal
Faculty of Mathematics and Physics, Charles University Prague, Czech Republic
skopal@ksi.mff.cuni.cz

ABSTRACT

Nowadays, the metric space properties limit the methods of indexing for content-based similarity search. The target of this paper is a data-driven transformation of a semimetric model to a metric one while keeping the data indexability high. We have proposed a genetic algorithm for evolutionary design of semimetric-to-metric modifiers. The precision of our algorithm is near the specified error threshold and indexability is still good. The paper contribution is a proof of concept showing that genetic algorithms can effectively design semimetric modifiers applicable in similarity search engines.

CCS CONCEPTS

• **Information systems → Search engine indexing**;

KEYWORDS

Genetic algorithm, Similarity search, Content-based retrieval

1 INTRODUCTION

Modern searching engines provide not only standard keyword search but also content-based retrieval. The content-based search aims at finding similar objects, e.g., in multimedia databases and generally in datasets of unstructured data. Similarity function (distance, respectively) measures the similarity (dissimilarity) of two database objects. To achieve applicability in different domains, the search engines have to be able to employ various similarity models.

The efficiency is the most crucial part of every search engine today. The naïve algorithm provides precise results, but the sequential scan is time-consuming. As the size of databases is growing, a suitable indexing is necessary. Some generic algorithms, such as LAESA [4], have additional requirements on the distance function δ used. Most of them require distance functions satisfying the metric properties. Metric properties, especially the triangle inequality, are an essential part of the indexing process. The basic idea is a construction of computationally cheap lower bound to the original expensive δ using the triangle inequality and some pre-selected objects P_i called pivots. As we know the distances $\delta(Q, P_i)$ (computed) and $\delta(X, P_i)$ (fetched from index), where Q is a query object and X is a database object, we can compute lower-bound distance as $\delta_{LB}(Q, X) = |\delta(Q, P_i) - \delta(X, P_i)|$. If $\delta_{LB}(Q, X) \geq r_Q$, then we can say that X is not within query range r_Q. However, many distance functions, *semimetrics*, do not satisfy the triangle inequality.

In this paper, we present an alternative approach for approximate search in dissimilarity spaces using a genetic algorithm. We took the idea of TriGen [5] algorithm and extended it to a new way of generating metric distance from semimetric. We also propose new methods of triplet sampling necessary for our algorithm.

2 RELATED WORK

One group of non-metric approaches use a different kind of lower-bounding instead of triangle inequality. In [2] authors define relaxed triangle inequality as $\rho\left(\delta(O_i, O_j) + \delta(O_j, O_k)\right) \geq \delta(O_i, O_k)$ which can be used as normal triangle inequality. In [3] usage of Ptolemy's inequality for lower bounds is presented. In [1] the authors present an evolutionary algorithm to generate artificial inequalities for a particular database and a distance function.

The TriGen algorithm [5] idea is to modify a semimetric distance to satisfy the triangle inequality. The assumption is that the original distance can be normed to $[0, 1]$. From the database, there is chosen a sample of objects to form triplets. The *triplets* are then distances between triplet objects that form a triangle in metric space. TriGen is implemented as a binary search to find the best triangle-generating (TG) modifier with parameter w (concavity). The best TG modifier should satisfy the triangle inequality for at least ω triplets and should have the as low intrinsic dimensionality ($\text{iDim} = \mu^2/2\sigma^2$) as possible (the indexability indicator).

3 EXPERIMENT

We have proposed the new kind of semimetric-to-metric modifier and designed a genetic algorithm to learn its parameters. Our Point Modifier (PMod) is represented as a sorted vector $\vec{P}$ of real values from $[0, 1]$. The $\vec{P}$ dimension D is variable, but for purpose of our experiment we used $D = 10$. The $\vec{P}$ values represent PMod as piecewise linear function. The $\vec{P}$ values must be strictly increasing as we want to preserve similarity orderings [5].

As the exact computation of real error and efficiency is time-consuming, we take over TriGen validation of modifier using the triplet error ϵ_t (ratio of triplets not satisfying the triangle inequality) and iDim as the indexability indicator.

3.1 Genetic Algorithm

Our genetic algorithm consists of the basic cycle that selects $2pop$ candidates for recombination and creates a new population of size $pop = 100$. We have implemented Tournament selector with adaptive K parameter that is increasing in time, represented as the percentage of a population from 1/3 to 2/3 at the end of the algorithm. The algorithm ends after 1000 iterations or after $Cat_m = 15$

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321907

consecutive catastrophic operator (see below) invocation. We have implemented the following operators.

Crossover, that randomly selects a point in Point Modifier representation and split both modifiers into two pieces. The first piece from the first modifier and the second piece from the second modifier are merged again in sorted sequence. *Mutation*, with probability $p_m = 0.05$ randomly choose a point in Point Modifier and move them up or down without breaking the order of parameters. *Catastrophic* operator, with adaptive tournament selector solves the problem with early population degeneration. The catastrophic operator is invocated after $Cat_n = 10$ generations without fitness improvement. In that case, the best 10% of the population is kept, and the rest of the population is newly randomly generated.

The most important part of our genetic algorithm is the fitness function (Eq. 1). During the pre-experiments, we have found that functions with few inflection points show better precision compared to function with a large number of inflection points. Based on that observation we have proposed the ConFactor($\vec{P}$) as the weighted ratio $\frac{\max\{2|C^+|,|C^-|\}}{2|C^+|+|C^-|}$ that prefers concave function and less inflection points. The C^+ (C^-, resp.) is the number of points in $\vec{P}$ where PMod is concave (convex).

$$f(\vec{P}) = \begin{cases} 1 - \epsilon_T(\vec{P}), & \text{for } \epsilon_T(\vec{P}) > \epsilon_t, \\ 1 + \frac{\text{ConFactor}(\vec{P})}{\sqrt{\text{iDim}(\vec{P})}}, & \text{otherwise.} \end{cases} \quad (1)$$

3.2 Database & Distance

We have tested our algorithm on Minkowski fractional L_p distances (semimetrics) with p coefficients $p_1 = 0.25$, $p_2 = 0.125$ and $p_3 = 0.0625$. As the database, we have used NASA vector database[1] with 40150 entries described as 20-dim. vectors. As the search index we have used LAESA with 5 randomly chosen pivots.

For learning the parameters we sampled 1000 entries of the original database and generated 25000 triplets. The TriGen algorithm [5] prefers anomalous (corner case) triplets in the sample, while such triplets have high variance in order to almost break the triangle inequality (i.e., $\min\{a+b/c\}$, where c is the biggest side of a triangle). In TriGen only top $p_a\%$ triplets with the biggest c value are considered. However, in experiments it was observed that our approach break triangle inequality in lower c values. So, we have proposed uniform anomalous triplet sampling, i.e., we sample $p_a^u\% = 0.05$ triplets of uniformly distributed c values (so we have smaller triangles). The remaining triplets are generated randomly.

3.3 Validation

We have compared our proposed genetic algorithm method with iDim-based TriGen algorithm, where as TG-modifiers the Fractional Point (FPMod) modifier and several Rational Bézier Quadratic modifiers (RBQ) modifiers were used. Since the RBQ modifier performed similarly to FPMod, we considered only the FP modifier. Both algorithms used triplet error ϵ_t and iDim for learning its parameters.

The effectivity of methods was evaluated by the query error function $\epsilon_{QE}(E, O) = |E \cap O| / \max\{|E|, |O|\}$ where E are expected objects (a result of sequential search), and O are observed objects (LAESA result). The efficiency was measured as the number of distance computations (ratio). Expected error threshold was $\theta = 0.1$.

[1] http://www.sisap.org/library/metricSpaces/dbs/vectors/nasa.tar

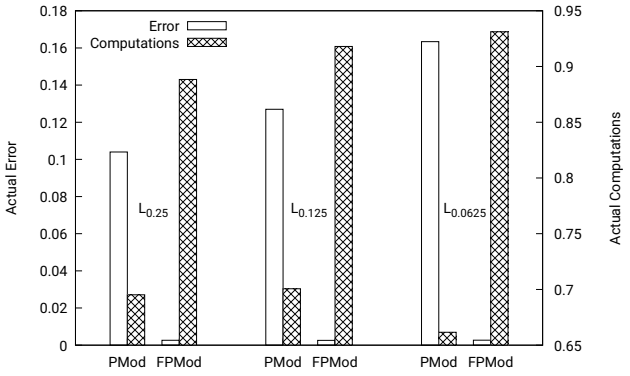

Figure 1: Comparison of average precision (left bar) and efficiency (right bar) of PMod (GA) and FPMod (TriGen).

Validation queries were range queries generated randomly for different ranges $\mu = \{0.1, 0.2, \ldots, 0.9\}$. For each query range $0.25\% \approx 100$ query objects were sampled from the original database.

4 RESULTS

We have ran our GA algorithm five times and took the average error and efficiency. During the tests, the FPMod (TriGen) was far below the threshold, but the ratio of distance computation was high. In Figure 1 see that our proposed PMod (GA) is nearly at the error threshold θ while the computation ratio is lower than using FPMod. We observe that different PMods have different query range optima that indicate we can construct PMods per group of queries.

5 CONCLUSIONS

We have shown that there is a possibility to generate triangle generating functions by genetic algorithms. PMod can be effectively used in the approximated similarity search. GA trained PMod needs just 75% of distance computation against TriGen. As the fitness function does not directly correspond with final results, the future work should aim to tune the fitness function. Also, the experiments with other non-metric distances are needed to confirm our hypothesis, in particular similarities on Linked Data datasets and their contexts.

Acknowledgments. This research has been supported by Czech Science Foundation (GAČR) project Nr. 19-01641S.

REFERENCES

[1] Tomáš Bartoš, Tomáš Skopal, and Juraj Moško. 2013. Efficient Indexing of Similarity Models with Inequality Symbolic Regression. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*. ACM, New York, NY, USA, 901–908.

[2] Ronald Fagin and Larry Stockmeyer. 1998. Relaxing the Triangle Inequality in Pattern Matching. *International Journal of Computer Vision* 30, 3 (01 Dec 1998), 219–231.

[3] Magnus Lie Hetland, Tomáš Skopal, Jakub Lokoč, and Christian Beecks. 2013. Ptolemaic access methods: Challenging the reign of the metric space model. *Information Systems* 38, 7 (2013), 989 – 1006.

[4] Francisco Moreno, Luisa Mico, and Jose Oncina. 2002. Extending LAESA Fast Nearest Neighbour Algorithm to Find the k Nearest Neighbours. 718–724.

[5] Tomáš Skopal. 2007. Unified Framework for Fast Exact and Approximate Search in Dissimilarity Spaces. *ACM Trans. Database Syst.* 32, 4, Article 29 (Nov. 2007).