






# Analysing Indexability of Intrinsically High-Dimensional Data Using TriGen

David Bernhauer<sup>1,2</sup>  and Tomáš Skopal<sup>1</sup>  

<sup>1</sup> SIRET RG, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

[bernhdav@fit.cvut.cz](mailto:bernhdav@fit.cvut.cz), [skopal@ksi.mff.cuni.cz](mailto:skopal@ksi.mff.cuni.cz)

<sup>2</sup> Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic

**Abstract.** The TriGen algorithm is a general approach to transform distance spaces in order to provide both exact and approximate similarity search in metric and non-metric spaces. This paper focuses on the reduction of intrinsic dimensionality using TriGen. Besides the well-known intrinsic dimensionality based on distance distribution, we inspect properties of triangles used in metric indexing (the triangularity) as well as properties of quadrilaterals used in ptolemaic indexing (the ptolemaicity). We also show how LAESA with triangle and ptolemaic filtering behaves on several datasets with respect to the proposed indicators.

## 1 Introduction

The real-world datasets for similarity search often exhibit high intrinsic dimensionality manifested by distance distribution with low variance and high mean [5]. The reason could be the high complexity of the similarity model within a given domain (lot of independent features), but often this is just a consequence of automated feature extraction processes, e.g., the inference of deep features [6]. Intrinsically high-dimensional data cannot be used for efficient exact search but, luckily, there have been developed many approximate methods [9] to tackle this problem for the price of a lower retrieval precision. Some of these methods elegantly avoid the direct problem of high intrinsic dimensionality by not indexing actual distances, but just permutations of pivots [4, 7]. These methods enabled competitive application of similarity search in real-world domains where maximal retrieval precision is not as critical as the performance. However, we must keep in mind these methods are limited in tuning the precision at runtime (from query to query) as well as they are restricted to pivot-based indexing schemes.

The TriGen algorithm [11] was proposed as a universal method for fast exact and approximate search in metric and non-metric spaces. So far, it was not analyzed as a method for (intrinsic) dimensionality reduction. In this paper

---

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

we empirically analyze this missing aspect. We also investigate the impact of TriGen modifications on the potential of ptolemaic indexing [8] that achieves better performance than metric indexing (though limited to ptolemaic metrics).

## 2 Background

When indexing data for fast similarity search, we face two fundamental concepts – the data indexability and the indexing model.

### 2.1 Indexability

The indexability generally refers to an ability to search efficiently a dataset  $\mathbb{S} \subset \mathbb{U}$  under a similarity model  $(\mathbb{U}, d)$ , regardless the indexing method used. The key is the distribution of data or, specifically, in case of similarity search it is the distribution of distances  $d(x, y)$  among data objects  $x, y \in \mathbb{S}$ . The classic indexability indicator for a metric space model  $(\mathbb{U}, d)$  is the intrinsic dimensionality [5], defined as the ratio of squared mean and doubled variance of the distance distribution;  $\text{iDim}(\mathbb{S}, d) = \frac{\mu^2}{2\sigma^2}$ . The lower iDim, the better indexability.

Alternatively, the ball overlap factor (BOF) [11] describes the ability to partition the dataset into non-overlapping ball-shaped regions. The BOF counts for how many object pairs will constitute overlapping balls (each ball radius is the distance to the ball center’s  $k$ th nearest neighbor).

### 2.2 TriGen Transformation

The TriGen algorithm [11] transforms the input distance space  $(\mathbb{U}, d)$  by use of triangle-generating or -violating modifiers and a dataset sample  $\mathbb{S}^* \subseteq \mathbb{S} \subset \mathbb{U}$  into a target space  $(\mathbb{U}, f(d))$ . A modifier  $f : R \rightarrow R_0^+$  must be an increasing function with  $f(0) = 0$  to preserve the ordering of distances<sup>1</sup> and thus search results with respect to sequential scan. The triangle-generating (concave) modifiers “inflate” all the triangles in the space to become more equilateral; then the dataset is less indexable as the intrinsic dimensionality increases. The triangle-violating (convex) modifiers have the opposite effect – “squeezing” the triangles and lowering the intrinsic dimensionality. The idea behind the triangle-violating modifiers is that they lower the intrinsic dimensionality (more efficient search) for the price of a retrieval error (some triangles break which shows in incorrect filtering by querying). The indexability indicators, like the intrinsic dimensionality or BOF, together with the T-error measuring the ratio of broken triangles, guide TriGen to determine the right modifier.

Unlike other methods that map the source distance space into the Euclidean space, the TriGen model is based solely on transformation of distances, hence there is no need for an expensive and static embedding of metric objects into vectors. In consequence, once a modifier is computed for a particular problem, its

<sup>1</sup> Ranking of objects  $x_i \in \mathbb{U}$  based on  $d(q, x_i)$  is the same as based on  $f(d(q, x_i))$ .

change (e.g., a precision guarantee) can be easily recomputed and the already created index just updated (no change in descriptors). This allows to switch between several TriGen modifiers at query time, providing thus flexible exact-to-approximate search (e.g., the NM-tree [10]). Other TriGen follow-ups include extensions to non-symmetric distances [3], and the genetic TriGen variants [1, 2].

In this paper, we inspect the TriGen in the role of dimensionality reduction method. In high-dimensional datasets (as measured by intrinsic dimensionality), all of the non-trivial triangles tend to be almost-equilateral. Then application of TriGen with triangle-violating modifiers could act as a lossless dimensionality reduction method by squeezing the triangles without the violation of triangle inequality (breaking the triangles by squeezing them too much). Our hypothesis is, the higher the (intrinsic) dimensionality of data is, the more almost-equilateral the triangles are, and so the more aggressive modifier could be applied while still keeping the triangles unbroken. Simply said, we analyze the question if TriGen could “cancel” the curse of dimensionality (to some extent) in similarity search.

### 2.3 Metric and Ptolemaic Indexing

The metric access methods (metric indexes) [5] use some construction of lower bounds using the triangle inequality. In the simplest case of pivot tables (aka LAESA), the three objects in the triangle are the query object  $q$ , a dataset object  $x$ , and a pivot  $p$  (i.e.,  $LB_{\Delta}(q, x) = |d(q, p) - d(p, x)|$ ). If the triangle is equilateral,  $LB_{\Delta}(q, x) = 0$  and so the dataset object  $x$  cannot be filtered by the lower bound. On the other hand, if the triangle is (squeezed to) a line segment, the lower bound gets maximal (i.e.,  $LB_{\Delta}(q, x, p) = d(q, x)$ ) and so it is “super-effective” for filtering.

Similarly, ptolemaic access methods (ptolemaic indexes) [8] use some construction of lower bounds using the Ptolemy’s inequality that operates on quadrilaterals (quadruplets, respectively). In the simplest (LAESA) case there are four objects in the quadrilaterals: the query object  $q$ , a dataset object  $x$ , and two pivots  $p_1, p_2$ , while a lower bound can be derived as

$$LB_{\text{pt}}(q, x, p_1, p_2) = \frac{|d(q, p_1) \cdot d(x, p_2) - d(q, p_2) \cdot d(x, p_1)|}{d(p_1, p_2)} \quad (1)$$

As the quadrilaterals are more complex than triangles, there is not a single best or worst quadrilateral example for the lower bound construction. Also the inflating and squeezing effect of TriGen modifiers is not clear in case of quadrilaterals, and so for ptolemaic indexing.

## 3 Triangle and Quadrilateral Distribution

The intrinsic dimensionality, as an indexability indicator, considers only distances themselves but does not consider that some distance combinations cannot be present in triangles at the same time, which is important for the filtering by metric access methods. The BOF compensates this issue, but it cannot be easily

generalized for Ptolemaic inequality or non-metric cases. Therefore, we define the *triangularity* to quantify the shape of triangle on a real-value scale from equilateral triangle, through line segment to broken triangle. Similarly, we define *Ptolemaicity* to quantify the shape of quadrilateral on a scale from tetrahedron, through line segment to broken equilateral.

Hence, we need to aggregate three distances forming a triangle into one number, with extremes for equilateral triangles and line segments. We could adopt the TriGen criteria (presented in [5]) used for determining the number of triangles that do not satisfy the triangle inequality. The *triangularity* is defined for a triangle  $a = d(x, y)$ ,  $b = d(y, z)$ ,  $c = d(x, z)$  by Eq. 2 – this ratio determines how “equilateralish” (or “inflated”) a triangle is. The triangularity is 1 for equilateral triangle, 1/2 means the triangle forms line segment (“squeezed”), and for values below 1/2 the triangle is broken (does not satisfy the triangle inequality).

$$\text{Triangularity}(a, b, c) = \frac{a + b}{2c}, \text{ where } a \leq b \leq c \quad (2)$$

After TriGen preprocessing, we expect the distribution will be shifted to line segments (“squeezed”) instead of almost-equilateral triangles. Knowledge of this common property makes the *triangularity* a good indicator of datasets with statistically high probability to exhibit bad indexability.

Moreover, we try TriGen for Ptolemaic indexing, though the TriGen modifiers were originally proposed for indexing using the lower bounds based on triangle inequality and not the Ptolemy’s inequality (Eq. 3). We would like to find out how the Ptolemy’s inequality holds in comparison with the triangle inequality. We define *ptolemaicity* of a quadrilateral as Eq. 4, where  $d(w, x)d(y, z), d(w, y)d(x, z) \leq d(w, z)d(x, y)$ . The greatest ptolemaicity value is 1, which represents regular tetrahedron and results in bad indexability. ptolemaicity 1/2 represents a line segment and for values below 1/2 the equilateral is broken (does not satisfy Ptolemy’s inequality).

$$(\forall w, x, y, z \in \mathbb{U}) d(w, x)d(y, z) + d(w, y)d(x, z) \geq d(w, z)d(x, y) \quad (3)$$

$$\text{Ptolemaicity}(w, x, y, z) = \frac{d(w, x)d(y, z) + d(w, y)d(x, z)}{2d(w, z)d(x, y)} \quad (4)$$

## 4 Analysis of High-Dimensional Data

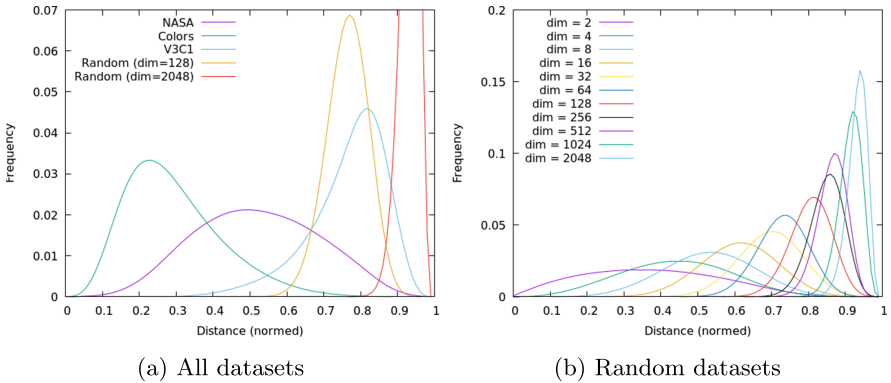
We have analyzed several datasets and looked at the intrinsic dimensionality and the retrieval efficiency (using the LAESA algorithm). Two low-dimensional datasets are from SISAP datasets: the 20-dimensional NASA dataset, and the 112-dimensional Colors dataset. As high-dimensional datasets we used a sample of the 2048-dimensional AlexNet image (V3C1) dataset, and several artificial datasets of dimensionality 2 to 2048 (randomly generated vectors). For all datasets we have used the Euclidean space, which is both metric and ptolemaic.

**Table 1.** Datasets statistics (iDim, distance computations with metric LAESA).

Dataset (dim)	without TriGen		with TriGen (zero error)	
	iDim	Dist. Comp.	iDim	Dist. Comp.
NASA (20)	$5.184 \pm 0.007$	2.12%	$4.593 \pm 0.007$	1.15%
Colors (112)	$2.742 \pm 0.003$	2.63%	$2.553 \pm 0.003$	2.08%
Random (128)	$181.328 \pm 0.304$	100%	$28.663 \pm 0.022$	95.78%
Random (2048)	$1967.66 \pm 184.295$	100%	$37.035 \pm 0.175$	99.3%
V3C1 (2048)	$30 \pm 0.050$	86.65%	$9.215 \pm 0.012$	45.39%

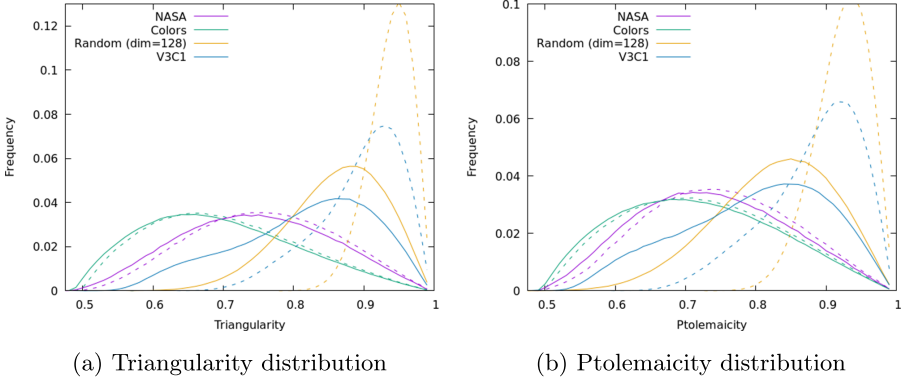
In Table 1 on the left, we present intrinsic dimensionality comparison and efficiency improvement of the metric LAESA (with randomly chosen 50 pivots) against sequential search. The iDim of Colors dataset is lower than iDim NASA dataset, however, LAESA performs better on NASA. Note the embedding dimensionality and iDim are dramatically different in case of V3C1 and Colors. Figure 1 shows distance distribution histograms for all datasets.

In Fig. 2a (dashed), we present triangularity distribution. As we expected, the distribution is shifted to the right side for high-dimensional datasets. This is the main assumption for transforming metric space using the TriGen into a more indexable one. Similarly, we have visualized the ptolemaicity distribution in Fig. 2b (dashed), which displays the same properties.

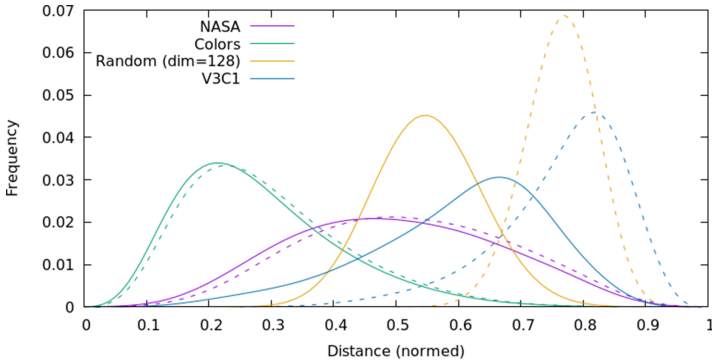


**Fig. 1.** Distance distribution comparison

Both triangularity and ptolemaicity distributions are similar, which means TriGen could be used for modification of Ptolemaic space, too. If the TriGen transforms both spaces consistently then, based on figures, Ptolemy’s inequality is violated earlier, because there is a higher number of line segments.



**Fig. 2.** Distribution of triangularity or ptolemaicity in datasets before (dashed) and after (solid) TriGen modifications.



**Fig. 3.** Dist. distribution before (dashed) and after (solid) TriGen modifications.

### 4.1 TriGen Modifications

In the first part of our experiment, we have configured TriGen to zero error tolerance. The measured retrieval error (as defined in [11]) was also zero, hence, we achieved faster and still exact search. Figure 3 shows the change of distance distributions after TriGen modifications were made.

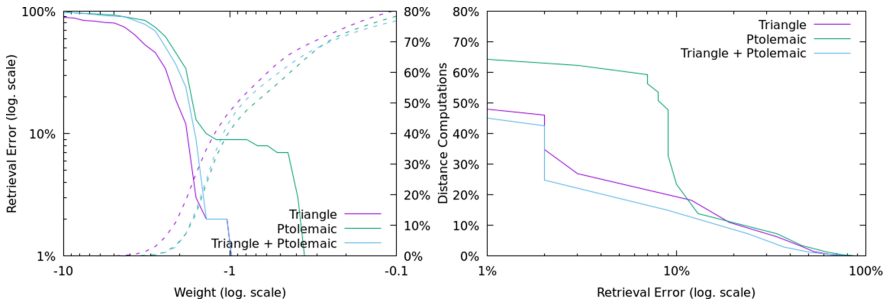
Table 1 on the right describes basic indicators after TriGen modifications, and we observe that triangle-violating modifications reduced the intrinsic dimensionality. The retrieval efficiency improved for all datasets (for some only slightly, but two times for NASA and V3C1). It indicates the presence of an inner structure beyond all conventional indicators, except for Random (2048) that is not indexable for exact search. However, TriGen can still transform a seemingly not-indexable dataset (V3C1, Random(128)) into partially indexable even for exact search.

Both triangularity (Fig. 2a) and ptolemaicity (Fig. 2b) distributions are flatter and shifted to the left as we expected. The ptolemaicity distribution is flatter than triangularity distribution, which means that Ptolemy’s inequality is more prone to a violation when used with TriGen.

### 4.2 Comparison of Real Performance

The TriGen algorithm controls the ratio of triangles satisfying the triangle inequality (so-called T-error tolerance) by a weight parameter that determines the convexity/concavity of the modifier. In the previous experiments we set T-error tolerance = 0 that (empirically) guarantees zero retrieval error. In Fig. 4a, we can see the dependence of distance computations and retrieval error on the weight (V3C1 dataset). We used just the triangle-violating (squeezing) modifications where  $-10$  weight is heavy squeezing and  $-0.1$  weight is almost no squeezing. We used LAESA with 50 randomly chosen pivots utilizing metric filtering, ptolemaic filtering, or both, and compared it with the sequential search.

The important observation is the ptolemaic filtering<sup>2</sup> has a similar pattern as the metric filtering. The general difference is in the shift of the ptolemaic curves to the right. The combination of triangle and ptolemaic filtering utilizes the benefits of both approaches. Triangle filtering deals with retrieval error caused by the Ptolemy’s inequality violation and the Ptolemy’s filtering deals with better efficiency, because of its ability to create better lower bounds.



(a) Efficiency (dashed) and retrieval error (solid) based on TriGen weight. (b) Efficiency per error (solid) based on TriGen weight.

**Fig. 4.** Efficiency and retrieval error (LAESA with 50 pivots on V3C1 dataset).

Another point of view is presented in Fig. 4a, where pairs of efficiency and retrieval error values from Fig. 4b are aggregated into single efficiency per error value. So, we get rid of the TriGen weight parameter and only observe how the real efficiency is dependent on real retrieval error, obtaining more readable results than when depicted individually.

<sup>2</sup> We used simple random selection of pivot pairs in ptolemaic filtering instead the better but slower Balanced heuristic [8].

### 4.3 Discussion

The intrinsic dimensionality is not always sufficient to predict the real efficiency of an indexing algorithm. First, because of some inner structure that can hardly be described by a single number. Second, the high number of low distances, triangularities, or ptolemaicities does not imply better indexability. A good example can be randomly generated vectors with one outlier, which will shift the whole histogram to the left.

The TriGen can be used for both precise and approximate search. The combination of both filtering inequalities improves not only efficiency but also lowers the retrieval error. There is a possibility in the future to try other kinds of inequalities and their ability to scale with TriGen.

## 5 Conclusions

We have introduced structure-sensitive empirical measures for the analysis of metric and Ptolemaic spaces and defined the triangularity and the ptolemaicity as the quantifiers of triangle and quadrilateral shapes. Analysis of high-dimensional data shows that it is possible to use TriGen as dimensionality reduction method that improves the efficiency of similarity search.

Although the TriGen was designed for transforming non-metric spaces into metric ones, we have shown that the inverse application on high-dimensional data is possible as well and efficient for both exact and approximate search. Moreover, experiments indicate that TriGen could be used with different types of filtering inequalities (like Ptolemy's). The combination of several filtering inequalities synergically deals with the advantages (better efficiency) and disadvantages (worse precision) of the individual methods.

## References

1. Bernhauer, D., Skopal, T.: Approximate search in dissimilarity spaces using GA. In: GECCO, pp. 279–280. ACM (2019)
2. Bernhauer, D., Skopal, T.: Non-metric similarity search using genetic TriGen. In: Amato, G., Gennaro, C., Oria, V., Radovanović, M. (eds.) SISAP 2019. LNCS, vol. 11807, pp. 86–93. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-32047-8\\_8](https://doi.org/10.1007/978-3-030-32047-8_8)
3. Boytsov, L., Nyberg, E.: Pruning algorithms for low-dimensional non-metric k-NN search: a case study. In: Amato, G., Gennaro, C., Oria, V., Radovanović, M. (eds.) SISAP 2019. LNCS, vol. 11807, pp. 72–85. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-32047-8\\_7](https://doi.org/10.1007/978-3-030-32047-8_7)
4. Chavez, E., Figueroa, K., Navarro, G.: Effective proximity retrieval by ordering permutations. *IEEE TPAMI* **30**(9), 1647–1658 (2008)
5. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L.: Searching in metric spaces. *ACM Comput. Surv.* **33**(3), 273–321 (2001)
6. Donahue, J., et al.: DeCAF: a deep convolutional activation feature for generic visual recognition. In: ICML, pp. I-647–I-655. JMLR.org (2014)



7. Esuli, A.: Use of permutation prefixes for efficient and scalable approximate similarity search. *Inf. Process. Manag.* **48**(5), 889–902 (2012)
8. Hetland, M.L., Skopal, T., Lokoč, J., Beecks, C.: Ptolemaic access methods: challenging the reign of the metric space model. *Inf. Syst.* **38**(7), 989–1006 (2013)
9. Patella, M., Ciaccia, P.: Approximate similarity search: a multi-faceted problem. *J. Discret. Algorithms* **7**(1), 36–48 (2009)
10. Skopal, T., Lokoč, J.: NM-tree: flexible approximate similarity search in metric and non-metric spaces. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) *DEXA 2008*. LNCS, vol. 5181, pp. 312–325. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-85654-2\\_30](https://doi.org/10.1007/978-3-540-85654-2_30)
11. Skopal, T.: Unified framework for fast exact and approximate search in dissimilarity spaces. *ACM Trans. Database Syst.* **32**(4), 29-es (2007)