Pattern Recognition Letters
journal homepage: www.elsevier.com

On the Application of Convex Transforms to Metric Search

Richard Connor[a,**], Alan Dearle[a], Vladimir Mic[b], Pavel Zezula[b]

[a] University of St. Andrews, Jack Cole Building, North Haugh, St Andrews, Fife KY16 9SX, Scotland, UK
[b] Masaryk University, Botanicka 68a, Brno 602 00, Czech Republic

ABSTRACT

Scalable similarity search in metric spaces relies on using the mathematical properties of the space in order to allow efficient querying. Most important in this context is the *triangle inequality* property, which can allow the majority of individual similarity comparisons to be avoided for a given query. However many important metric spaces, typically those with high *dimensionality*, are not amenable to such techniques. In the past *convex transforms* have been studied as a pragmatic mechanism which can overcome this effect; however the problem with this approach is that the metric properties may be lost, leading to loss of accuracy. Here, we study the underlying properties of such transforms and their effect on metric indexing mechanisms. We show there are some spaces where certain transforms may be applied without loss of accuracy, and further spaces where we can understand the engineering tradeoffs between accuracy and efficiency. We back these observations with experimental analysis. To highlight the value of the approach, we show three large spaces deriving from practical domains whose dimensionality prevents normal indexing techniques, but where the transforms applied give scalable access with a relatively small loss of accuracy.

1. Introduction

Search based on similarity has become a common activity in the modern data processing landscape. Probably the most generic approach to similarity searching constrains the search space to be a *metric space* (Kelly (1955)), which encompasses a wide variety of different data models. Given a domain of objects D, a metric space is a pair (D, d) where $d : D \times D \rightarrow \mathbb{R}_0^+$ is a distance function which quantifies the dissimilarity of objects. This function must be *positive*, *symmetric*, and satisfy the *triangle inequality* property: that is, for any $x_1, x_2, x_3 \in D, d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$. Metric similarity searching is popular due to its wide applicability, and many metric indexes have been proposed (Zezula et al. (2006)).

In a similarity search, an object (the *query*) q from the space D is presented, and the task is to find similar objects from a given large, finite space $X \subseteq D$. Typically, the cost of assessing the pairwise dissimilarity $d(q, x_i), x_i \in X$ is high.

The intent of metric *indexing* is to organise the data set during a pre-processing phase so that the majority of objects which are dissimilar to a subsequent query can be excluded from a search, whilst maintaining perfect accuracy of results. This can be achieved in spaces with low dimensionality, but becomes increasingly difficult to achieve as dimensionality increases. The simplest mechanism which can be used is known as *pivot exclusion*, as explained in Section 2.1, although others exist. The contribution of this paper is towards the application of transforms to the distance function which increase the probability of exclusion occurring, but without a major loss of accuracy. The technique extends to any metric space.

The two most common types of query are known as *range* and *nearest neighbour* (NN). In a range query, for some query $q \in X$ and distance t, the solution set is defined as $\{x \in X | d(q, x) \leq t\}$. NN queries return, for some k, the k closest objects to q. The two are strongly related and, in a continuous space, there always exists a threshold value t which will return the *kNN* for any k; in the dialogue we consider queries with a fixed threshold, but the discussion applies to both types.

Our approach is based on the use of convex transforms applied to metric spaces. For a given transform $C : \mathbb{R} \rightarrow \mathbb{R}$ our technique maps a metric space (D, d) into another space $(D, C \circ d)$. The tradeoff is that $(D, C \circ d)$ will have better indexing properties, but may lose the metric property of triangle

** Corresponding author
e-mail: rchc@st-andrews.ac.uk (Richard Connor)

inequality, and thus the mathematical basis of pivot exclusion. This can improve the performance of metric indexing but in general can also result in a loss of perfect accuracy.

Unlike other efficiency improvments such as dimensionality reduction, the transform C is applied to the distance function, rather than the domain, of the original space. This means that the techniques shown may be applied to any metric space.

1.1. Related Work

1.1.1. Transforms on Metric Spaces

The technique of using convex transforms to speed-up the similarity search was first proposed and investigated by Skopal (2007), who focused on the definition of suitable convex transforms for a given dataset (Skopal (2007)). Bernhauer and Skopal (2019) investigate both convex and concave transforms that can be utilised to transform non-metric spaces to almost precise metrics to facilitate their indexing. Skopal and Lokoc (2008) propose the *NM-tree* for efficient similarity search. It enhances the *M-tree* (Ciaccia et al. (1997)) with the convex or concave transform proposed by Skopal (2007).

Our work enhances this previous analysis with a deeper examination of the geometric effects of transforms within spaces, and in particular shows the effect of purely convex transforms over a particular class of high dimensional space. We do not suggest ways of finding appropriate transforms, nor specialised indexing mechanisms which may take advantage of them.

1.1.2. Metric Search Techniques

In Section 5 we demonstrate the efficacy of using convex transforms with two well-known pivot-based metric approaches, *LAESA* (Micó et al. (1994)) and *Vantage Point Trees* (VPT) (Yianilos (1993).) The first of these performs filtering, while the latter performs indexing.

LAESA uses a set of pivots $p_i \in D$, and pre-computes all distances $d(x, p_i), x \in X$ in advance. Having a query object $q \in D$, its distances to pivots $d(q, p_i)$ are evaluated to check the lower bounds for $d(q, x)$ given by the rule of the triangle inequality for each $x \in X$. If the condition: $\exists p_i : |d(x, p_i) - d(q, p_i)| > r$ is true, then the distance $d(q, x)$ can be deduced to be greater than r and therefore need not be calculated.

VPTs are search structures based on ball partitioning which partition data according to (relative) distances to a pivot stored in a node of the tree. The recursive partitioning and insertion of data leads to a binary tree. The search algorithm for a range query traverses the tree and determines the inclusion or exclusion of the children based on the distance from the pivot to the query object. The determination of whether or not subtrees should be accessed is based on the lower bound of the distance from q to objects in the left and right subtrees.

1.2. Contribution

Here we focus on convex transforms that speed up search, in order to shed light on the nature and consequences of their use in different classes of metric space.

Our contributions are as follows:

- to give a deeper understanding of why the efficiency of search mechanism are improved;
- to show why, in high-dimensional spaces, such transforms are less likely to lead to false negative outcomes than in lower-dimensional spaces;
- to show examples of spaces where transforms can be safely applied, thus improving search efficiency without introducing inaccuracy,
- and to show experimentally that this approach is effective over high-dimensional spaces.

We end with experiments over three data sets drawn from the world of image retrieval which, with normal metrics applied, gain no benefit from metric indexing techniques. We show that applying convex transforms can achieve scalable indexing of these spaces while still returning a majority of correct results.

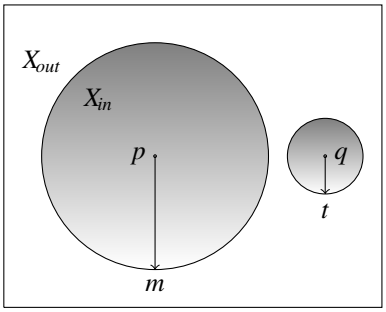


Fig. 1: Pivot-based exclusion. The subsets X_{in} and X_{out} are calculated during pre-processing, according to the distances of objects from p, for some fixed distance m. If $d(q, p) > m + t$, X_{in} cannot contain a solution to the query.

2. Background

2.1. Pivot Exclusion in Metric Spaces

Figure 1 illustrates the principle of *pivot exclusion* in a two-dimensional Euclidean space. The finite dataset X is partitioned during pre-processing into X_{in} and X_{out} , according to whether objects drawn from X are within distance m of a distinguished reference point p, or otherwise.

When query q is presented for evaluation, $d(p, q)$ is calculated. The figure shows a case where $d(q, p) > m + t$. In this case, no triangle can exist with side lengths a, b, c where $a \leq m$; $b \leq t$ and $c = d(q, p)$. It is therefore impossible for any solution to the query to be within X_{in}, and so the entire subset X_{in} can be excluded from the search. Similarly, although not illustrated here, if $d(q, p) < m - t$, X_{out} cannot contain a solution.

In general neither the query q nor the threshold t are available at the time of pre-processing and the choice of pivot may imply that $m - t < d(q, p) < m + t$, meaning that it is impossible to safely exclude either partition. In general this is governed by a probability which is affected by various aspects of the context; however as dimensionality increases, the probability of exclusion generally decreases. In this work, we address this issue.

2.2. Finite 2D Projections

In general, metric spaces cannot be drawn on a 2D plane in a way that faithfully captures all inter-object distances. However in any metric space, the triangle inequality property also

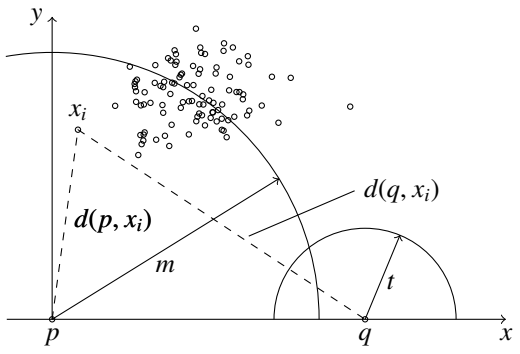


Fig. 2: A 2D(p, q) projection of a uniformly distributed 20-dimensional Euclidean space. Values p and q have been selected randomly from a generated space and plotted at $(0, 0)$ and $(d(p, q), 0)$ respectively. Then for 100 further generated values x_i, a point is plotted in the upper 2D coordinate space, such that $d(p, x_i)$ and $d(q, x_i)$ are preserved.

implies a finite 2D Euclidean embedding, and so it is valid to discuss the *angles* of a triangle constructed according to the distances among any three objects selected from the space. A useful diagrammatic form is given in Figure 2 by a projection that we denote the $2D(p, q)$ projection. The $2D(p, q)$ projection maps a metric space into a 2D Cartesian coordinate system in a way that preserves the pairwise distances between two fixed points p, q, and arbitrary points x_i from the metric space. We introduce this diagrammatic form to faithfully illustrate the various inter-object distances that we are considering in this paper. The two fixed points p and q are mapped to coordinates $(0, 0)$ and $(d(p, q), 0)$ and any further object is represented by the unique point (x, y), where $y \geq 0$, that preserves distances from x_i to p and q (as shown by the dashed lines in the figure). This projection accurately represents the distances from the two fixed points p and q whereas the apparent distances among the other points x_i are not preserved. In fact, a single point on the $2D(p, q)$ projection represents an unbounded set of loci from the original space, with arbitrary distances among them. Figure 2 depicts the $2D(p, q)$ projection of a generated uniformly distributed 20-dimensional Euclidean space.

The apparent cluster of points seen in the $2D(p, q)$ projection is a manifestation of the so-called *curse of dimensionality*, where the variance observed over sampled distances becomes small. The arc drawn around the point q represents a hypersphere in the original space whose radius t represents a small search radius for the query q. The arc centred at p with diameter m represents a hypersphere which contains approximately half the volume of the space. In general, as is common in high dimensional spaces, if such hyperspheres intersect, exclusion based on triangle inequality is not possible.

As will be seen, we will rely upon certain aspects of these distributions in order to understand the class of transform which can usefully be applied to increase the efficacy of metric indexing.

2.3. Introduction to Convex Transforms

Metric spaces do not define an ordering over objects. However, given a specific object $q \in D$ all objects drawn from D can be ordered according to their distance from q. In this paper we are interested in the application of numeric transforms over the distances within a metric space; that is, a class of numeric functions defined over the range of the original distance function. Any strictly monotonic function can be applied to the metric without affecting the ordering.

Formally, in this paper, we consider continuously increasing functions that are derivable across their whole domain, with strictly positive second derivative. That is increasing strictly *convex transforms* (Blumenthal (1953)). For example, any function of the form $y = x^n$, where $n > 1$, is a convex transform over positive values.

3. Convex Transforms

In this section we show how the application of a convex transform to a metric space increases the efficiency of pivot-based exclusion mechanisms by increasing the probability of exclusion occurring for any pivot/query pair. We also show how any such transform can potentially violate the triangle inequality property of the space, which would therefore invalidate the safety of the exclusion condition. The purpose is to show that in some spaces it may be possible to establish that certain transforms will not violate triangle inequality, and therefore safely increase tractability. In other spaces, it may be possible to calculate the probability of a violation, and thus quantify the value of using a convex transform as an approximate search mechanism by understanding the efficiency/recall tradeoffs.

3.1. Increase of largest angle

Much of our analysis here is based upon *angles*. Many metric domains are non-Euclidean with no inherent concept of angle. However all metric spaces possess the triangle inequality property, which implies a finite embedding in 2D Euclidean space for any three objects. Thus, for any three objects x_i, x_j, x_k from the metric domain, we can consider the angles α, β, γ at their corresponding vertices, independent of the type of domain.

Without loss of generality from here on we consider convex transforms C whose domain and range is $[0, 1]$. In this context we define a convex transform as one where $C(0) = 0$; $C(1) = 1$; and whose second derivative is positive within this range.

Lemma 1. *Let $a, b \in [0, 1]$ with $a < b$. Then*

$$\frac{C(b)}{C(a)} > \frac{b}{a} \quad , \quad a \neq 0$$

Proof. Figure 3 depicts the plot where the x and y axes correspond to original and transformed distances, respectively. Line L given by points $[0, 0]$ and $[a, C(a)]$ is a secant of C. Since C is strictly convex, L intersects C just in these two points, and C is strictly above line L for all $x : a < x \leq 1$. Therefore the line between $[0, 0]$ and $[b, C(b)]$ has a larger gradient, giving the required result. □

In the following, we consider three objects $x_1, x_2, x_3 \in D$ and denote a, b, c their pairwise distances such that $0 < a \leq b \leq c \leq 1$. We denote T the triangle of these distances.

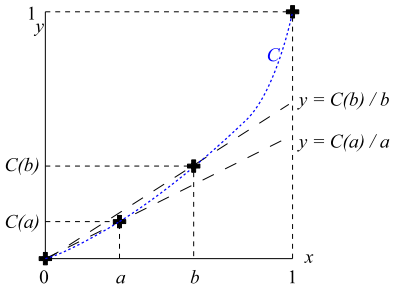


Fig. 3: Convex transforms increase larger distances relative to smaller ones. x-axis: original distances, y-axis: transformed distances.

Theorem 1. *The application of C to triangle T increases the maximum angle in T.*

Proof. Consider the application of C to the sides of triangle T to form triangle T'. If c is the longest side of T we can superimpose T' onto T along c scaled by the factor $\frac{c}{C(c)}$. Then by Lemma 1 the smaller sides $C(a)$ and $C(b)$ are relatively shorter, thus increasing the angle between a and b:

$$\begin{aligned} C(c) \cdot c/C(c) &= c \\ C(a) \cdot c/C(c) &< C(a) \cdot a/C(a) = a \\ C(b) \cdot c/C(c) &< C(b) \cdot b/C(b) = b \end{aligned}$$

□

3.2. Increased Probability of Pivot Exclusion

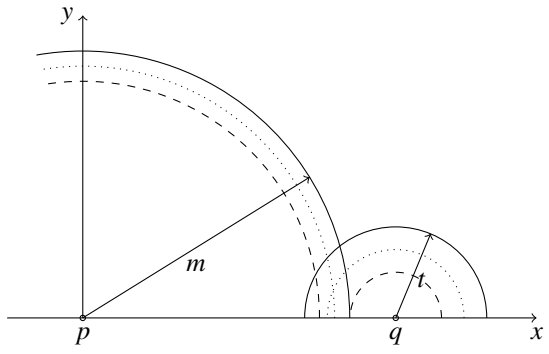


Fig. 4: The effect of convex transforms on the hypersphere boundaries. In this diagram two transforms are applied, and the resulting 2D space is overlaid after magnification to preserve the coordinates of q in the $2D(p, q)$ projection. After the second projection, the hyperspheres no longer intersect and an exclusion operation would be allowed.

The solid lines in Figure 4 show a $2D(p, q)$ projection where exclusion is not permitted according to the triangle inequality property, as the hypersphere centred around p with radius m intersects with the hypersphere centred around q with radius t.

The dotted and dashed lines also show representations of the same space after two convex transforms have been applied. The application of the transform will reduce all of the measurements in the original space as discussed; for each transform a magnification has been applied to place q on the same point: that is, the Cartesian plane for each transform C has been magnified by $d(p, q)/C(d(p, q))$.

In both cases it can be seen that all distances less that $d(p, q)$, e.g. m and t, are relatively reduced as according to Lemma 1. The dashed lines show the effect of applying the transform has reduced the intersection between the hyperspheres to zero. In this case, any exclusion mechanism will allow the exclusion of the hypersphere centred at p to occur.

In any metric indexing mechanism, there will be a large number of such cases where a possible exclusion is sought; for each of these, therefore, an increased probability of exclusion is given by any convex transform.

This argument is valid only for the case where $d(p, q) > m$. A similar argument exists where $d(p, q) < m$.

3.3. Consequences of Transform Application

Convex transforms can violate triangle inequality. For example, if three points p, x, q are considered such that $d(p, x) + d(x, q) = d(p, q)$, then under any strictly convex transform the triangle inequality property is lost, since $C(d(p, x)) + C(d(x, q)) < C(d(p, q))$ due to Lemma 1.

From the properties of the space however, it may be possible to deduce that some transforms may be safely applied. For example, there may be no co-linear points, or it may be possible to place an upper bound on the largest angle. Under such circumstances there exists a safe class of transform. If this is a probabilistic determination, then a convex transform might give a useful engineering compromise between efficiency and accuracy. In such cases precision is never compromised but recall may be diminished.

To give an example, Figure 5 shows a $2D(p, q)$ projection of a metric space which is in the class of so-called *square metrics* (see section 4.1). One important property of such metrics is that, for any three objects from the space, a triangle constructed from the three inter-object distances contains only acute angles.

In this case, we can observe that the points x_1 and x_2 in the $2D(p, q)$ projection form obtuse triangles with the points p and q and therefore no objects in the metric space can map to these points. By a generalisation of this observation, those areas shaded in red cannot contain any objects since, when combined with the points p and q will form triangles containing obtuse angles. Therefore the intersection of the depicted hyperspheres represented by the two arcs does not contain any objects.

It may be further noted that if the transform $f(x) = x^2$ is applied to metric underlying this $2D(p, q)$ projection, then any object x_i which has mapped to the unshaded area will maintain the triangle inequality property with p and q.

Figure 6 shows a randomly sampled selection of data from the SIFT data set, which will be properly introduced in Section 5. Here, p and q represent arbitrary objects from the data. The plot shows the $2D(p, q)$ projection of 200 other points from the set: 100 randomly selected, and the 100 nearest neighbours to q. The arc centred on p represents the mean distance of the randomly selected points to p, and the arc centred on q represents the query threshold which returns these 100 nearest neighbours.

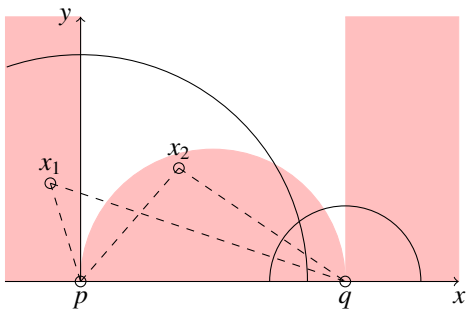


Fig. 5: For a so-called "square" metric, once p and q are projected into the 2D space, it is impossible for any third object to be projected into the shaded region. Any triangle formed with p, q and a point in the shaded region would contain an obtuse angle.

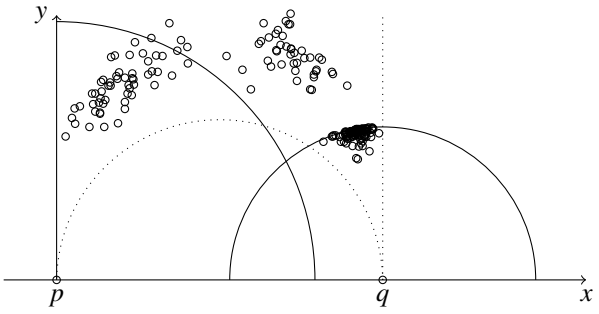


Fig. 6: Here, p and q represent arbitrarily chosen objects from the SIFT data set (see Section 5.) The plot shows the $2D(p, q)$ projection of 200 other points: 100 randomly selected, and the 100 nearest neighbours to q. The intersection of the two arcs is highly likely to be empty, and therefore any convex transform is probably safe to apply.

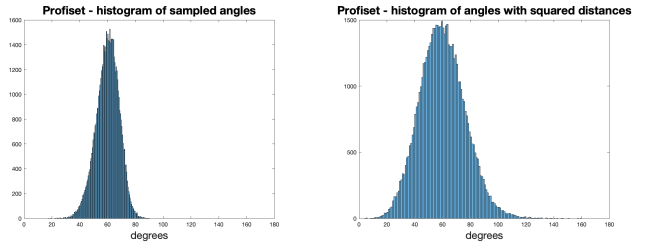
While triangle inequality does not allow the exclusion of the subset closest to p, it can be seen from the patterns that there is a low probability of the two intersecting hyperspheres containing any data. If Figures 5 and 6 are compared, it can be seen that the distribution of distances in the latter all lie within the unshaded areas in the former, showing that at least for this sample, the space has the property of a square metric. If all subsets of the data conform to this same pattern, then it would be safe to apply the metric $f(x) = x^2$ to this space.

4. Convex Transforms between Metrics

In this section, we examine convex functions which define the transformation of one metric space (D, d) into another metric space $(D, C \circ d)$. We have observed that pairs of spaces exist in which one space is a transform of the other, and in such cases the latter may always be searched more efficiently than the former without loss of accuracy.

4.1. Square Metrics

The class of metrics of *negative type*, also referred to as ℓ^2 or "square" metrics, attracts considerable mathematical interest. These are metric spaces of the form (D, d) such that there exists an isometric embedding within a Euclidean space $(\mathbb{R}^n, \sqrt{d})$.



(a) Original space, no obtuse angles (b) Squared dists, no triangle violations

Fig. 7: Angles in the 10,000 triangles T sampled from the DeCAF dataset before and after the convex transform $C(x) = x^2$.

Such spaces underlie important results for example in the domains of compression and cuts (Brinkman and Charikar (2005); Chawla et al. (2005)). By definition, the underlying Euclidean space can be transformed by $C(x) = x^2$ without loss of metric properties. One of the properties of such spaces is described by the following theorem:

Theorem 2. *For any metric space (D, d) where, for any $x, y, z \in D$, all of the angles within a triangle formed with sides of length $d(x, y)$, $d(y, z)$ and $d(x, z)$ are acute, then the space (D, d^2) is also a proper metric space.*

Proof. Let $a = d(x, y)$, $b = d(y, z)$ and $c = d(x, z)$, and γ be the angle at y in the 2D triangle formed from these distances. The triangle inequality ensures $a + b \geq c$, and from the cosine rule:

$$a^2 + b^2 = c^2 + 2ab\cos\gamma$$

If all angles are acute, i.e. $\gamma \leq 90°$, then $\cos\gamma$ is positive, and therefore

$$a^2 + b^2 \geq c^2$$

which shows that triangle inequality is preserved in the squared metric. The other metric properties (positivity, symmetry, identity) are trivially preserved, therefore (D, d^2) is a proper metric space. □

One property of high dimensional spaces is the decreasing variance of sampled distances, which leads to a predominance of acute-only triangles. We illustrate this phenomenon using the *DeCAF* descriptors as described in Section 5.

Figure 7a shows the histogram of 30,000 angles sampled from the DeCAF dataset, and Figure 7b shows the angles after squaring each distance. There are no obtuse angles in the original sample and, consistent with the analysis above, no triangle inequality violations occur when the distances are squared. This outcome gives some confidence that squaring distances can be applied to improve tractability of this dataset with little loss of recall.

This analysis is of course intuitive rather than rigorous, as is that shown for the *SIFT* data in Figure 6. As will be seen, neither space perfectly preserves triangle inequality under the square transform, but both can usefully have the square transform applied to improve indexing with little loss of accuracy.

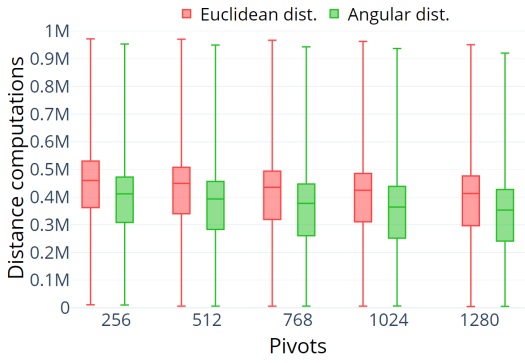


Fig. 8: Distribution of Euclidean and angular distances for the LAESA data set, 1M SIFT descriptors, 100NN queries, 1000 random query objects.

4.2. Cosine Distances

Cosine "distance" is used in many contexts (including *SIFT*, see Section 5) as a dissimilarity measure. The usual form of the measurement, that is $1 - \cos\theta$ where θ is the angle between $\vec{x_1}$ and $\vec{x_2}$, is not a proper metric. There are two easy ways to make it so: using the angle θ itself, or using the difference of ℓ_2-normalised vectors.

The following *angular* and *Euclidean* distance functions[1] define the same ordering of objects with respect to an arbitrary selected pivot:

$$d_{ang}(\vec{x_1}, \vec{x_2}) = \theta/\pi$$

where θ is the angle between the vectors, and

$$d_{eu}(\vec{X_1}, \vec{X_2}) = \left(\sqrt{\sum_i (X_1[i] - X_2[i])^2} \right) /2$$

where $\vec{X_1}, \vec{X_2}$ are ℓ_2-normalised forms of $\vec{x_1}, \vec{x_2}$.

The latter distance is in common use. The following convex transform maps from the Euclidean distance to the angular distance, which is more efficient for indexing:

$$C_{eu\_ang}(x) = \frac{cos^{-1}(1 - 2x^2)}{\pi}$$

Figure 8 shows the relative query efficiency before and after the application of this transform to the SIFT descriptors. The x-axis shows the number of pivots[2] and the y-axis the distribution of distance measurements for 100 nearest neighbour queries with 1000 randomly selected query objects. The ℓ_2 normalised vectors compared using angular distance thus form a more tractable space than using the Euclidean distance, while the underlying mathematical analysis shows there is no loss in query accuracy.

5. Experimental Evaluation

This section gives empirical evidence to support our mathematical reasoning only as a proof of concept. We show results with a high-dimensional generated Euclidean space, followed by analysis of three spaces derived from the image retrieval domain. The spaces used are summarised in Table 1.

Table 1: Data Sets

Name	Dimensions	Metric
Random28	28	Euclidean
SIFT	128	ℓ_2-normed Euclidean
MPEG7	280	MPEG-7 standard distance
DeCAF	4096	post-RELU Euclidean

In all cases, the space as presented is intractable for metric search techniques, where *scalability*[3] is generally considered to be the most important factor.

All of the data used comes from public open sources, and our code is available for download[4].

5.1. Test Data

All of the sets chosen have 10^6 elements, which is considered small in this context.

Random28 comprises 28-dimensional vectors of floating point numbers, generated using the Java *Random* class with an uniform distribution and queried with Euclidean (ℓ_2) distance.

SIFT descriptors (Lowe (1999)) derive from the *ANN_SIFT1M* dataset[5] and comprise 128 floating point values. Although queried with the ℓ_2 distance, these vectors are ℓ_2 normalised and thus this metric acts as a proxy for Cosine distance, as discussed in Section 4.2.

MPEG7 comprises a combination of five MPEG-7 visual descriptors (MPEG7 (2002)) from the CoPhIR data collection (Bolettieri et al. (2009)). Each descriptor is represented as a 280-dimensional vector, and the metric used is the metric defined by the MPEG-7 standards body.

DeCAF descriptors (Donahue et al. (2014)) are extracted from the *Profiset* image collection[6] using the AlexNet convolutional neural network (Krizhevsky et al. (2012)), from which the second-last fully connected layer is extracted as a *post-Relu* 4,096-dimensional array of floating-point values.

5.2. Experimental Setup

In all cases we search for k nearest neighbours[7] of a 1,000 randomly selected query objects q_i. In all the experiments we use simple transforms of the form $f(x) = x^n$ with a range of exponent n starting at 1.0, which represents the unchanged space.

For each of 1,000 randomly selected queries, we use a precalculated ground truth. We set each query threshold to return 100 nearest neighbours when $n = 1.0$. We use this threshold with different exponents and report the number of results returned. As n increases, this number decreases in return for a more efficient search. We report the number of distance calculations executed as a measure of efficiency. It is worth noting

[1] whose range is normalised into $[0, 1]$

[2] Since the SIFT dataset consists of vectors of 128 floating point numbers, the LAESA algorithm with many pivots can be more expensive than sequential evaluation of all distances. The purpose of this experiment is to illustrate the filtering power of the triangle inequalities in two different spaces.

[3] usually defined as the cost of search being $O(\log n)$

[4] `https://bitbucket.org/richardconnor/convex_transforms`

[5] `http://corpus-texmex.irisa.fr/`

[6] `http://disa.fi.muni.cz/profiset/`

[7] with fixed-radius search in each case, using pre-computed radii

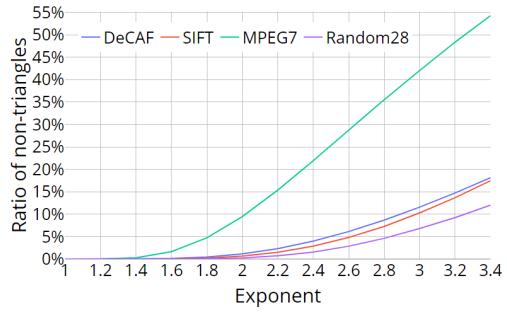
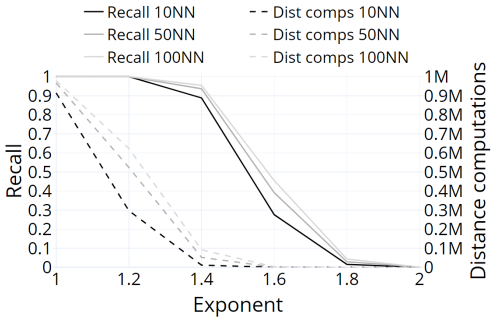
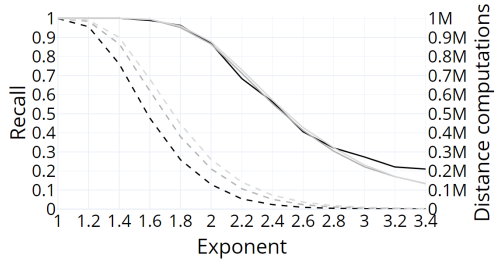


Fig. 9: Percentage of triangle violations for various powers



(a) LAESA



(b) VPT

Fig. 10: 10,50,100 NN queries over Random28

that in all cases precision is perfect; our approach cannot return false positive results. For *Random28*, we report mean recall for 10, 50 and 100NN searches respectively; for the other sets we report only recall at 100NN as this allows the presentation of per-query distribution of results.

In one final experiment, we have selected random object triples from each space to measure the percentage of triangle violations, the results of which are shown in Figure 9.

5.3. Results

Figures 10a and 10b depict results of the search with LAESA and VPT over the *Random28* data. The solid lines depict the mean recall, and the dashed lines express the mean number of distance computations needed to evaluate each query. Recall is plotted against the left-hand y axis, and the number of distance computations performed is plotted against the right-hand y axis.

Figures 11 and 12 show the results of querying the other data sets. To show the variance we use the *Tukey box-plots* to illustrate the distribution of measured values.

5.4. Discussion

All of the results justify the analysis of this paper, showing that a single class of convex transform serves to improve the efficiency of two different metric access methods, in return for a loss of recall. One important quantification however is the relative rates of change: in all cases we have examined, the evaluation cost improves significantly before the recall is affected.

In all cases, the space as presented is intractable for metric search techniques, where *scalability*[8] is generally considered to be the most important factor. The VPT is a data structure that allows scalability. However, this will only be achieved in spaces where only a small proportion of the distance calculations are required for small spaces, as this implies that for larger data sets, an increasing proportion of distance calculations may be avoided. A commonly used rule of thumb is that around 90% of calculations should be avoided in small spaces. It can be seen from Figure 12 that, at this point, the VPT still returns a healthy proportion of correct results.

It can be seen that convex transforms give more false negative results for LAESA than VPT at a given exponent. This is due to the fact that LAESA uses 256 pivots to prune the search space, whereas the depth of the VPT is $\log_2$ of the data size, 20 in these experiments. Since an error in any of these will impact the results, there is much more chance of this with the higher number of pivots used by LAESA.

The recall/cost ratio is generally better for LAESA, however LAESA does not have the potential to give a scalable search. Whereas the VPT mechanism has $O(\log n)$ asymptotic complexity in both time and space, LAESA has $O(n)$.

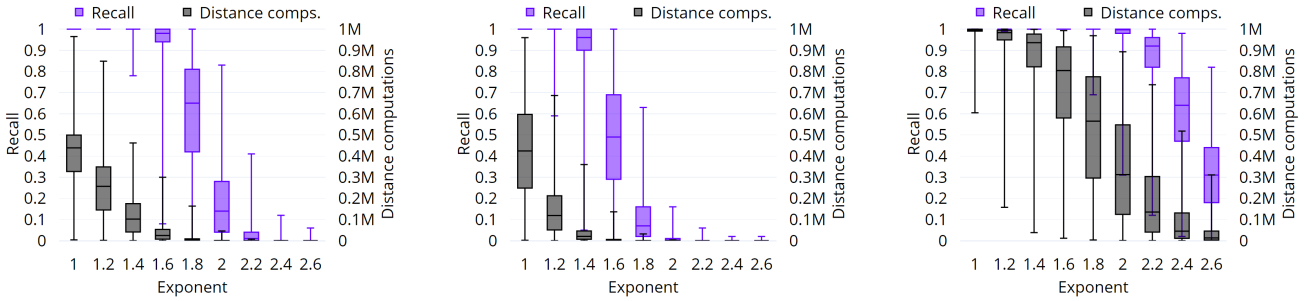
6. Conclusions and Further Work

We have explored how the application of monotonic convex transforms can improve the efficiency of metric search. This improved efficiency stems from the changes to the distribution of distances in the space when the transform is applied. We have shown in detail how these changes increase the efficacy of any pivot-based metric search mechanism. Our experiments demonstrate that the approach of applying convex transforms is effective over high-dimensional spaces, in particular showing that a scalable space, in which the majority of correct results are returned, may be achieved.

The use of convex functions can, in general, violate the triangle inequality constraint, which is one of the fundamental postulates of a metric space. However, our mathematical and empirical analysis shows cases where convex transformations can be guaranteed to remain metric and thus safely applied to improve efficiency without introducing inaccuracies. Even if the guarantee cannot be achieved, application of transforms can give useful efficiency/accuracy tradeoffs. We have shown that this is particularly the case in high-dimensional spaces.

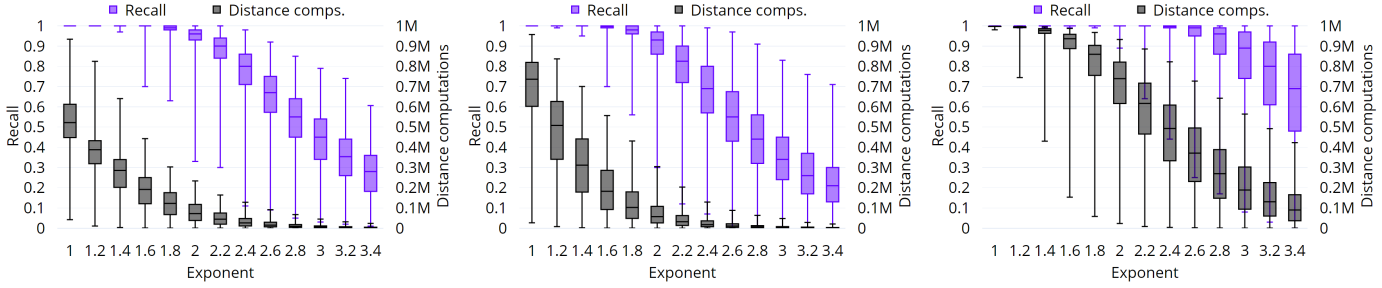
Understanding exactly when violation of the triangle inequality is likely to occur, based on properties of the original space, remains an interesting topic. Intuition in this respect

[8] usually defined as the cost of search being $O(\log n)$



(a) SIFT (b) MPEG7 (c) DeCAF

Fig. 11: LAESA with 256 pivots, 100NN queries, 1000 random query objects



(a) SIFT (b) MPEG7 (c) DeCAF

Fig. 12: VPT, 100NN queries, 1000 random query objects

seems to be helped by Theorem 1 which states that, for any triangle formed by three objects of the space, application of any convex transform increases the biggest angle. In one particular case the observation that the biggest angle $\leq 90°$ allows all distances to be squared without violation. However we have not yet found a generalisation which allows a safe transform to be deduced from geometric properties of the original space, or a general method for predicting the probability of individual violations. In this paper, we have considered only a single class of convex transform, namely $f(x) = x^n$ for various values of n. It is very likely that other classes of transform suit particular spaces better, for example that shown in Section 4.2.

Acknowledgments

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16 019/0000822) and by ESRC "Administrative Data Research Centres 2018" (No. ES/S007407/1).

References

Bernhauer, D., Skopal, T., 2019. Non-metric similarity search using genetic trigen, in: Similarity Search and Applications - 12th International Conference, SISAP 2019, Newark, NJ, USA, Proceedings, pp. 86–93.

Blumenthal, L., 1953. Theory and applications of distance geometry. Clarendon Press, London.

Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., Rabitti, F., 2009. CoPhIR: a test collection for content-based image retrieval. CoRR abs/0905.4627v2. URL: `http://cophir.isti.cnr.it`.

Brinkman, B., Charikar, M., 2005. On the impossibility of dimension reduction in l1. J. ACM 52, 766–788.

Chawla, S., Gupta, A., Räcke, H., 2005. Embeddings of negative-type metrics and an improved approximation to generalized sparsest cut, in: Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA. pp. 102–111.

Ciaccia, P., Patella, M., Zezula, P., 1997. M-tree: An efficient access method for similarity search in metric spaces, in: VLDB'97, August 25-29, 1997, Athens, Greece, Morgan Kaufmann. pp. 426–435.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T., 2014. Decaf: A deep convolutional activation feature for generic visual recognition, in: ICML 2014, Beijing, China, pp. 647–655.

Kelly, J.L., 1955. General Topology. The university series in higher mathematics. D. Van Nostrand Company, Inc.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097–1105.

Lowe, D.G., 1999. Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999, pp. 1150–1157.

Micó, L., Oncina, J., Vidal, E., 1994. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. Pattern Recog. Letters 15, 9–17.

MPEG7, 2002. Multimedia content description interfaces. part 3: Visual.

Skopal, T., 2007. Unified framework for fast exact and approximate search in dissimilarity spaces. ACM Trans. Database Syst. 32, 29.

Skopal, T., Lokoc, J., 2008. Nm-tree: Flexible approximate similarity search in metric and non-metric spaces, in: Database and Expert Systems Applications, 19th International Conference, DEXA 2008, Italy. Proceedings, Springer. pp. 312–325.

Yianilos, P.N., 1993. Data structures and algorithms for nearest neighbor search in general metric spaces, in: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 311–321.

Zezula, P., Amato, G., Dohnal, V., Batko, M., 2006. Similarity Search – The Metric Space Approach. volume 32 of *Advances in Database Systems*. Springer.