Efficient Algorithms for Similarity Search in Axis-Aligned Subspaces

Michael E. Houle[1], Xiguo Ma[2], Vincent Oria[2], and Jichao Sun[2]

[1] National Institute of Informatics, Tokyo 101-8430, Japan
meh@nii.ac.jp
[2] New Jersey Institute of Technology, Newark NJ 07102, USA
{xm23,oria,js87}@njit.edu

Abstract. Many applications — such as content-based image retrieval, subspace clustering, and feature selection — may benefit from efficient subspace similarity search. Given a query object, the goal of subspace similarity search is to retrieve the most similar objects from the database, where the similarity distance is defined over an arbitrary subset of dimensions (or features) — that is, an arbitrary axis-aligned projective subspace. Though much effort has been spent on similarity search in fixed subspaces, relatively little attention has been given to the problem of similarity search when the dimensions are specified at query time. In this paper, we propose several new methods for the subspace similarity search problem. Extensive experiments are provided showing very competitive performance relative to state-of-the-art solutions.

1 Introduction

Similarity search is of great importance to applications in many different areas, such as data mining, multimedia databases, information retrieval, statistics and pattern recognition. Specifically, a similarity query retrieves from the database those objects that most closely resemble a supplied query object, based on some measure of pairwise similarity (typically in the form of a distance function). Due to its importance, much effort has been spent on the efficient support of similarity search. However, most existing approaches consider search only with respect to a fixed feature space. In this paper, we focus on the subspace similarity search problem, in which the calculation of similarity values is restricted to a subset of dimensions specified along with the query object.

As with similarity search on fixed spaces, subspace similarity search may also have an impact in application areas where the feature set under consideration changes from operation to operation. Such changes could be due to a modification of query preferences (as in content-based image retrieval), or to the determination of the local structure at different locations within data (as in subspace clustering), or to a systematic exploration of feature subspaces (as in feature selection). In content-based image retrieval, images are often represented by feature vectors extracted based on color, shape, and texture descriptors. In an exploration of the data set, a query involving one combination of features (such as color) may be followed by a query on a different combination (such as shape). In subspace clustering [1], the formation of an individual cluster is generally assessed with respect to a subset of features that most closely describe the concept

A.J. Machado Traina et al. (Eds.): SISAP 2014, LNCS 8821, pp. 1–12, 2014.
DOI: 10.1007/978-3-319-11988-5_1 © Springer International Publishing Switzerland 2014

associated with the cluster. Since verification of a cluster requires the identification of a feature subset together with an object subset, the effectiveness of the overall clustering process may depend on the efficient processing of subspace similarity queries. Wrapper methods for feature selection [2] require an evaluation process, such as k-nearest neighbor (k-NN) classification, for the identification of effective combinations of features. Exploration of feature subspaces can be extremely time-consuming when the neighborhoods are determined using exhaustive search, due to the exponential number of potential combinations involved. To accelerate the process, the efficient support of subspace similarity search is needed.

Almost all existing similarity search indices require that the similarity measure and associated vector space both be specified before any preprocessing occurs. Traditional methods for fixed spaces (as surveyed in [3]) cannot be effectively applied for the subspace search problem: the subspaces to be searched are typically not known until query time, but even if they were known in advance, constructing an index for every possible query subspace would be prohibitively expensive. Of all the methods for similarity search appearing in the research literature, only very few have been specifically formulated for the subspace search problem; a survey of these methods will be presented in Sect. 2.1. In general, existing solutions for subspace similarity search suffer greatly in terms of the computational cost.

Of the two main types of similarity queries (k-NN queries and range queries), k-NN queries are often more important, due to the difficulty faced by the user in deciding range thresholds. This is especially the case for the search in subspaces, since the range values of interest will typically depend on the number of features associated with the subspace. In this paper, we focus only on k-NN queries.

We now formally define the subspace search problem for k-NN queries. Given an object domain $\mathcal{U}$, let $S \subseteq \mathcal{U}$ denote a set of database objects represented as feature vectors in $\mathbb{R}^D$. The set of features will be denoted simply as $F = \{1, 2, \cdots, D\}$, with feature $i \in F$ corresponding to the i-th coordinate in the vector representation. Let $d : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be a distance function defined for the vector space. Given an object vector $u = (u_1, \ldots, u_{|F|}) \in S$, its projection with respect to a feature subset $F' \subseteq F$ is the vector $u' = (u'_1, \ldots, u'_{|F|})$ such that for all $i \in F$, $u'_i = u_i$ whenever $i \in F'$, and $u'_i = 0$ otherwise. The feature set F' thus indicates a unique axis-aligned projective subspace to which distance calculations can be restricted.

Definition 1 (Subspace k-NN Query). *Given a query object $q \in \mathcal{U}$, a query subspace $F' \subseteq F$, and a query neighborhood size k, a subspace k-NN query $\langle q, F', k\rangle$ returns the k objects of S most similar to q, for the distance function $d_{F'}(q, u) \triangleq d(q', u')$, where q' and u' are the projections of q and u with respect to F'.*

As an example of a subspace distance function, for any given $p \in [1, \infty)$, the L_p distance between two objects $q, u \in \mathcal{U}$ restricted to the axis-aligned projective subspace F' is defined as

$$d_{F'}(q, u) = \left(\sum\nolimits_{i \in F'} |q_i - u_i|^p\right)^{\frac{1}{p}} .$$

In this paper, we present algorithms for subspace similarity search following the multi-step search strategy [4,5], utilizing 1-dimensional distances as lower bounds to efficiently prune the search space. The main contributions of this paper are:

- algorithms specifically tailored for subspace similarity search, both exact and approximate;
- a guide to the practical choice of an important algorithm parameter, based on a theoretical analysis of sample properties;
- an experimental evaluation across data sets of a variety of types and sizes, showing the efficiency and competitiveness of our algorithms.

The remainder of this paper is organized as follows. Sect. 2 discusses related work on subspace search and multi-step search algorithms. Our proposed algorithms are presented in Sect. 3. In Sect. 4, through experiments on several real-world datasets, we contrast the performance of our methods with those of existing methods. The discussion is concluded in Sect. 5.

2 Related Work

2.1 Subspace Similarity Search

Relatively few similarity search methods exist that are specifically designed for subspace search. In [6], the Partial VA-file (PVA) was proposed, which adapts the vector approximation file (VA-file) [7] to support subspace queries. The VA-file, designed for fixed-space similarity search, stores a compressed approximation of the data as a single file; at query time, the compressed approximation is scanned in its entirety, and uses the information for pruning the search within the original dataset. PVA, on the other hand, stores an approximation of data on each dimension separately, and processes the search using only those 1-dimensional VA-files that correspond to dimensions involved in the query. In [8], the Dimension-Merge Index (DMI) was developed, which combines multiple 1-dimensional index structures to answer subspace queries. DMI builds an index for each dimension separately (of any desired type), and utilizes those indexes with respect to the query dimensions to perform the search. The final query result is obtained by aggregation across neighborhoods associated with each of the query dimensions. In [9], the Projected R-Tree (PT) was proposed as a redefinition of the classical search structure R-tree [10] for subspace similarity search. Instead of integrating results of queries on 1-dimensional indices, PT utilizes a single index built on the full feature space (an R-tree) to answer queries with respect to subspaces. A best-first search heuristic is employed, subject to the restriction that only the query dimensions are considered for distance computations. PVA, DMI and PT all produce exact query results; however, as we shall see in Sect. 4, all tend to suffer greatly in terms of their computational cost.

Another approach to the subspace search problem was proposed in [11], for range queries. Here, the search space is reduced through the application of the triangle inequality on several pivot points. Since k-NN queries are not directly supported by this algorithm, for the experimental comparison in Sect. 4, we restrict our attention to PVA, DMI and PT.

2.2 Multi-step Search Algorithms

Our proposed solutions for the subspace search problem make use of multi-step search algorithms. Multi-step search was originally proposed for the adaptive similarity search problem, which aims to find the most similar objects to a query object from the database with respect to an adaptive similarity measure — one that can be determined by the user at query time. Multi-step search computes a query result using a fixed 'lower-bounding' distance function that is adapted to answer the same query with respect to a user-supplied 'target' distance function. The function d_l is a lower-bounding distance for the target distance d_t if $d_l(u, v) \leq d_t(u, v)$ for any two objects u, v drawn from a domain for which both d_l and d_t are defined.

The first multi-step k-NN search algorithm was proposed by Korn et al. [12]. Later, Seidl and Kriegel [4] proposed a more efficient multi-step algorithm. The algorithm scans the neighborhood list of the query object with respect to d_l to retrieve candidates for the query result, and stops when the candidate k-NN distance (target distance) is no larger than the lower-bounding distance currently maintained by the scan. The algorithm is optimal in that it produces the minimum number of candidates needed in order to guarantee a correct query result, given only a list of candidates ordered according to d_l. However, despite this performance guarantee, the algorithm may still be expensive in practice. Using the Seidl-Kriegel algorithm as a starting point, Houle et al. [5] designed an approximate multi-step algorithm, MAET+, with an early termination condition. MAET+ utilizes tests of a measure of the intrinsic dimensionality of the data, the *generalized expansion dimension* (GED) [13,5], to guide early termination decisions. In the remainder of this paper, we will refer to the Seidl-Kriegel algorithm as SK.

3 Algorithm

We now present our solutions to the subspace similarity search problem. Let us first introduce some additional notation. For any object $q \in \mathcal{U}$ and any subspace $F' \subseteq F$, let $N_{F'}(q, k)$ denote the set of k-nearest neighbors of q within database S with respect to subspace distance $d_{F'}$. Ties are broken arbitrarily but consistently. Let $\delta_{F'}(q, k)$ denote the k-th smallest subspace distance (with respect to F') from q to the objects in S.

The strategy underlying our methods involves the application of multi-step search, using a lower-bounding distance function to filter a candidate set from the database, and using the target distance function to refine the candidate set to obtain the final query result. The main concern here is the determination at query time of a lower-bounding distance function suitable for the indicated subspace. Due to the exponential number of possible subspaces, it is impossible to explicitly preprocess the data for every subspace. Instead, as potential lower-bounding distance functions, we consider only the 1-dimensional distance $d_{\{i\}}$ associated with each feature $i \in F$. Assuming that the lower-bounding property holds between $d_{\{i\}}$ and subspace distance $d_{F'}$ for all $i \in F'$ (which is the case for many practical distance measures, including the Euclidean distance), there are $|F'|$ lower-bounding distance functions that can be used in the search. However, practical performance may vary considerably according to the choice of $d_{\{i\}}$. In order to minimize the risk of choosing a poorly-performing lower-bounding distance, we select the distance function corresponding to the most discriminative query dimension. This is done by ranking the dimensions based on data variance, a simple yet effective ranking technique. Two ranking strategies are proposed in this paper: Single Ranking (SR) and Multiple Ranking (MR).

```
Algorithm SK_SR (query q, subspace $F'$, target neighborhood size $k$)
   // Preprocessing step: obtain a single ranking of all dimensions.
1: for each dimension $i \in F$ do
2:     $\mu_i \leftarrow \frac{1}{|S|} \sum_{u \in S} u_i$.
3:     $\mathrm{Var}_i \leftarrow \frac{1}{|S|} \sum_{u \in S} (u_i - \mu_i)^2$.
4: end for
5: Rank all dimensions $i \in F$ in decreasing order of $\mathrm{Var}_i$. Let $\Re(F)$ denote this ranking.
   // Query processing step: perform a multi-step search.
6: Among all the dimensions in subspace $F'$, select the dimension $i^*$ with the highest ranking
   according to $\Re(F)$.
7: Call SK$(q, k)$ to produce the query result, with $d_{\{i^*\}}$ as the lower-bounding distance function,
   and $d_{F'}$ as the target distance function.
```

Fig. 1. The description of algorithm SK_SR

3.1 Single Ranking Strategy

The first of our proposed algorithms — SK_SR, described in Fig. 1 — employs a single overall ranking of dimensions based on variance. There are two main phases: a preprocessing phase and a query processing phase. In the preprocessing phase, the algorithm generates a single ranking of the dimensions, in terms of the variances of the data values computed separately for each of the dimensional coordinates — the larger the data variance for a given dimension, the higher the ranking of that dimension. In the query processing phase, as the lower-bounding distance function used in multi-step search, the algorithm chooses the dimension of highest rank from among the query dimensions. When Algorithm SK is used for performing the multi-step search (in Line 7), the query result is guaranteed to be correct. As an alternative, we may also utilize the approximate multi-step algorithm MAET+; this variant of subspace similarity search will be referred to as MAET+_SR. Specifically, we make a call to MAET+(q, k, t), where $t > 0$ is a parameter governing an early termination criterion. Larger choices of t can be expected to yield query results with higher accuracies at the possible expense of computational cost. In [5], a sampling method was designed for choosing t so that a desired proportion of potential queries can be correctly answered with high probability. For more details, we refer the reader to [5].

Note that like DMI, our search strategy requires the construction of a separate index for each of the dimensions. However, unlike DMI, our algorithms access only a single index per query, namely the most discriminative query dimension in terms of variance.

3.2 Multiple Ranking Strategy

The single ranking strategy has the advantage of being straightforward to apply. However, its effectiveness may be limited whenever the variance of a particular dimension differs greatly when restricted to the vicinity of differing query objects. For this reason, we have also designed a multiple ranking strategy that takes the query object into account when generating a ranking of dimensions.

Our multiple ranking strategy for subspace similarity search, SK_MR, is described in Fig. 2. In the preprocessing step, the algorithm first samples m reference points from

Algorithm **SK_MR** (*query q, subspace F', target neighborhood size k, sample size m, variance neighborhood size K*)

```
// Preprocessing step: create multiple rankings of dimensions.
Create a reference set $R \subseteq S$ by sampling $m$ points from the database, uniformly at random and without replacement.
for each reference point $v \in R$ do
    for each dimension $i \in F$ do
        $\mu_{v,i} \leftarrow \frac{1}{|K|} \sum_{u \in N_{\{i\}}(v,K)} u_i$.
        $\mathrm{Var}_{v,i} \leftarrow \frac{1}{|K|} \sum_{u \in N_{\{i\}}(v,K)} (u_i - \mu_{v,i})^2$.
    end for
    Rank all dimensions $i \in F$ in decreasing order of $\mathrm{Var}_{v,i}$. Let $\Re_v(F)$ denote this ranking.
end for
// Query processing step: perform a multi-step search.
Linearly scan $R$ to find $v^*$, the nearest reference point to $q$ with respect to $d_{F'}$.
Select the query dimension $i^* \in F'$ with the highest ranking according to $\Re_{v^*}(F)$.
Call SK$(q, k)$ to produce the query result, with $d_{\{i^*\}}$ being the lower-bounding distance function and $d_{F'}$ being the target distance function.
```

Fig. 2. The description of algorithm SK_MR

the database. Then, with respect to each reference point v, the algorithm determines a ranking (from highest to lowest) of all dimensions based on the variance of the coordinate values for the dimension in question, this time computed over a neighbor set of v (instead of over the entire dataset S). In the query processing step, the algorithm first finds the nearest reference point v^* of q in the query subspace (using sequential search within the reference set), and then uses the ranking of dimensions precomputed for v^* in the processing of query q. Again, we may replace SK with MAET+ to derive an approximation variant, MAET+_MR.

Two parameter choices must be considered when applying the multiple ranking strategy: the number of reference points m, and the size K of the neighborhoods within which data variance is computed. As will be shown in Sect. 4, the choice of K does not greatly affect the performance, provided that it is small relative to the dataset size $|S|$. On the other hand, the number of reference points m must be chosen with more care. If m is too large, the identification of the most discriminative query dimension may become unaffordable. If m is too small, the dimension i^* selected for multi-step search may not be very discriminative for the query. We next discuss how to choose a reasonable value for m.

Determining the Reference Set Size. For the multiple ranking strategy to be effective, for any given query point q, its nearest reference point v^* should be among the nearest neighbors of q within S (all with respect to the query subspace). Otherwise, the ranking of dimensions based at v^* may fail to approximate the ranking based at q. Fortunately, the following technical lemma shows that with even a relatively small number of reference points, v^* can lie in the local neighborhood of q with high probability.

Lemma 1 (Houle et al. [5]). *Let A be a set of positive integers, and let $A' \subseteq A$ be a subset sampled uniformly at random without replacement. Given a threshold τ, let a* *and a' refer to the number of elements in A and A', respectively, that are no greater than τ. Take η and η' to refer to the proportion of those elements within A and A', respectively. For any real number $\phi \geq 0$, we have* $\mathsf{Pr}[|\eta - \eta'| \geq \phi] \leq 2e^{-2\phi^2|A'|}$.

Proof. Since A' is generated by uniform selection from A, random variable a' follows the hypergeometric distribution with expectation $\mathsf{E}[a'] = a|A'|/|A|$. In [14], Chvátal showed that random variable a' satisfies both $\mathsf{Pr}[\mathsf{E}[a'] \geq a' + \phi|A'|] \leq e^{-2\phi^2|A'|}$ and $\mathsf{Pr}[\mathsf{E}[a'] \leq a' - \phi|A'|] \leq e^{-2\phi^2|A'|}$. Both inequalities can be combined to yield the following error bound:

$$\begin{aligned}\mathsf{Pr}[|\eta - \eta'| \geq \phi] &= \mathsf{Pr}\left[\left|\frac{a}{|A|} - \frac{a'}{|A'|}\right| \geq \phi\right] = \mathsf{Pr}\left[\left|\frac{\mathsf{E}[a']}{|A'|} - \frac{a'}{|A'|}\right| \geq \phi\right] \\ &= \mathsf{Pr}[|\mathsf{E}[a'] - a'| \geq \phi|A'|] \leq 2e^{-2\phi^2|A'|}. \qquad \square\end{aligned}$$

To apply this lemma to the analysis of the choice of reference set size, let $A = \{1, 2, 3, \ldots, |S|\}$ represent the ranks of all the objects in S with respect to a query object q, and let $A' \subseteq A$ store the ranks of all the reference points ($|A'| = m$). Also, let τ be the rank of the reference point v^*, which implies that $\eta' = 1/|A'|$. A small value of η would therefore indicate that v^* is in the local neighborhood of q, as desired. From Lemma 1, we know that the probability of η deviating from $\eta' = 1/|A'|$ by more than $\phi \geq 0$ is at most $2e^{-2\phi^2|A'|}$. That is, the probability of η being significantly larger than $1/|A'|$ vanishes quickly as the sample size $|A'|$ grows. In practice, even small sample sizes allow us to obtain reasonably small values of η with high probability. For example, if $|A'| = 5,000$ and $\phi = 0.02$, the lemma indicates that the probability of $\eta \geq 0.0202$ is at most 0.037, or equivalently, the probability of $\eta < 0.0202$ is at least 0.963.

4 Experimental Results

In this section, we present the results of our experimentation. We compared our algorithms with the state-of-the-art approaches PVA, PT and DMI.

4.1 Experimental Framework

Data Sets. Five publicly-available data sets were considered for the experimentation, so as to compare across a variety of set sizes, dimensions and data types.

- The Amsterdam Library of Object Images (ALOI) [15] consists of $110,250$ images of 1000 small objects taken from different viewpoints and illumination directions. The images are represented by 641-dimensional feature vectors based on color and texture histograms (for a detailed description of the image features, see [16]).
- The MNIST data set [17] consists of $70,000$ images of handwritten digits from 500 different writers, with each image represented by 784 gray-scale texture values.
- The Cortina data set [18] consists of $1,088,864$ images gathered from the World Wide Web. Each image is represented by a 74-dimensional feature vector based on homogeneous texture, dominant color and edge histograms.

- The Forest Cover Type set (FCT) [19] consists of $581,012$ data points, with each representing a 30×30 square meter area of forest. Each point is represented by 54 attributes, associated with elevation, aspect, slope and other geographical characteristics.
- The ANN_SIFT data set [20] consists of 10^7 SIFT descriptors [21] of 128 dimensions. The SIFT descriptors were extracted from approximately 10^6 general images.

Methodology. For each test, 1000 queries were generated at random, each consisting of an object q selected from the database, and a query subspace F'. Unless stated otherwise, the number of query dimensions was $|F'| = 8$, and the target neighborhood size was $k = 10$. Two quantities were measured for the evaluation: query result accuracy and execution time. The results were reported as averages over the 1000 queries performed. The execution time is shown as a proportion of the time needed for a sequential search of the entire dataset. For each query, the accuracy of its k-NN result is defined as the proportion of the result falling within the true k-NN (subspace) distance to q:

$$\frac{\mid \{v \in Y \mid \ d_{F'}(q,v) \leq \delta_{F'}(q,k)\} \mid}{k},$$

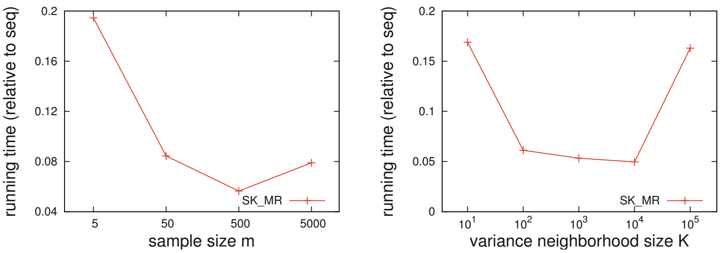
where Y denotes the k-NN query result of q in subspace F' ($|Y| = k$). The Euclidean distance was used for all experiments.

4.2 Effects of Varying m and K on the Multiple Ranking Strategy

For the first set of experiments, for all of the datasets under consideration, we tested the effects on the multiple ranking strategy due to variation of the sample size m and variance neighborhood size K. When varying the sample size m, the variance neighborhood size K was chosen to be approximately 1% of the dataset size: specifically, the choices were $K = 10^3$ for ALOI and MNIST, $K = 10^4$ for Cortina and FCT, and $K = 10^5$ for ANN_SIFT. When varying K, the sample size m was fixed at 500 for all datasets tested. Since we observed similar trends in the results for all datasets, due to space limitations, in this version of the paper, we show the results of varying m and K only for the ALOI dataset.

The results for varying m are shown in Fig. 3(a). Here, we see that $m = 500$ is a sufficiently-large sample size for multiple ranking strategy to be effective, which is better than indicated by the theoretical analysis. From Lemma 1, we know that if $m = 500$, then for any dataset with any number of data points, the probability of $\eta < 0.062$ is at least 0.945 ($\phi = 0.06$). Recall that the effectiveness of the multiple ranking strategy is expected to increase as η diminishes. Our experimental findings show that the value of η in practice is typically much smaller than what the analysis indicates. In order to reduce the computational cost of the experimentation, we therefore set $m = 500$ for all remaining experiments.

Fig. 3(b) shows the results of varying K. As expected, the variance neighborhood size K does not greatly affect the performance, provided that it is set to reasonably small values relative to the dataset size. For all remaining experiments, we set $K = 10^3$ for ALOI and MNIST, $K = 10^4$ for Cortina and FCT, and $K = 10^5$ for ANN_SIFT.



(a) $K = 1000$ (b) $m = 500$

Fig. 3. The effects of varying m and K for the multiple ranking strategy, with dataset ALOI

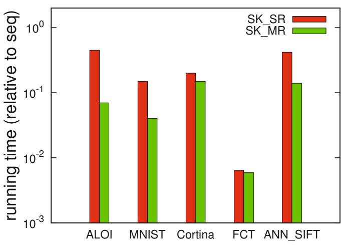


Fig. 4. The comparison of SR and MR on all datasets tested

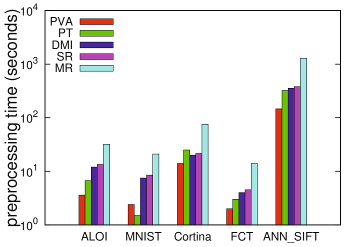


Fig. 5. Preprocessing costs for all datasets tested

4.3 Comparison of Single Ranking and Multiple Ranking

We next compared the performance of the single ranking and multiple ranking strategies; the results are shown in Fig. 4. Unsurprisingly, multiple ranking outperformed single ranking for all datasets tested. Due to space limitations, in all experiments involving competing methods, we show a comparison of results only for multiple ranking.

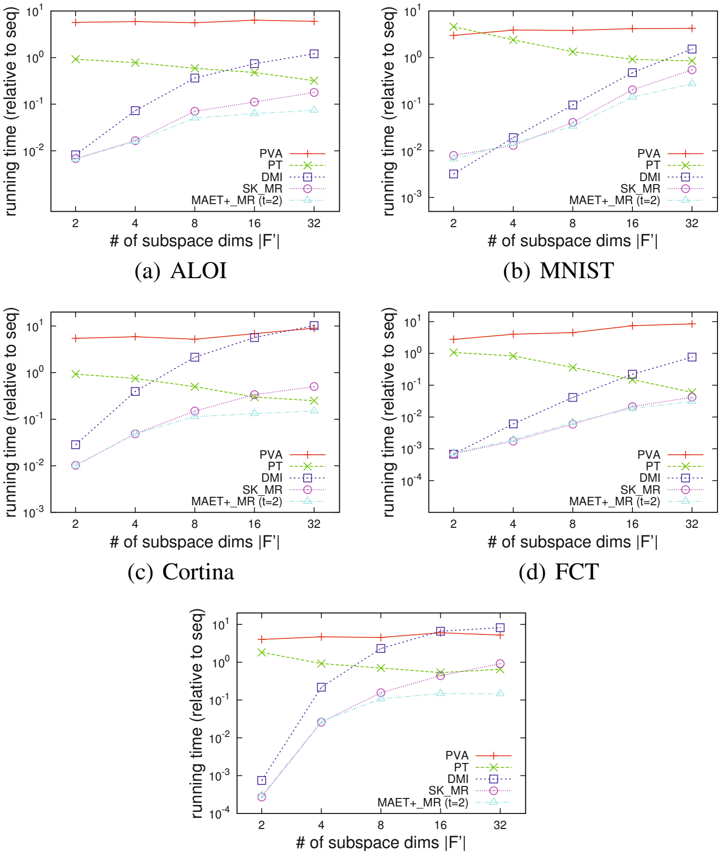
4.4 Comparison with Other Methods

We conducted two sets of experiments for the comparison of our algorithms with competing methods, varying each of two parameters in turn: the number of subspace dimensions $|F'|$, and the target neighborhood size k. Specifically, we varied $|F'|$ from 2 to 32 while fixing $k = 10$, and varied k from 5 to 40 while fixing $|F'| = 8$.

The results of varying $|F'|$ are shown in Fig. 6. For all datasets and all choices of $|F'|$, our proposed methods generally outperform their competitors. Among all the methods tested, PVA is the most expensive, perhaps due to its use of sequential scan.

PT utilizes an R-tree built on the full-dimensional space to answer queries in subspaces; consequently, one would expect it to be less effective for subspaces in which $d_{F'}$ differs greatly from d_F. This can explain the improvement in the performance of PT as the number of subspace dimensions increases. Nevertheless, due to the limits on the performance of R-trees for spaces of even moderate dimensionality, PT will still become prohibitively expensive as the number of subspace dimensions grows.

DMI processes queries by aggregating partial results across neighborhoods with respect to every query dimension. The aggregation may become prohibitively expensive



(a) ALOI

(b) MNIST

(c) Cortina

(d) FCT

(e) ANN_SIFT

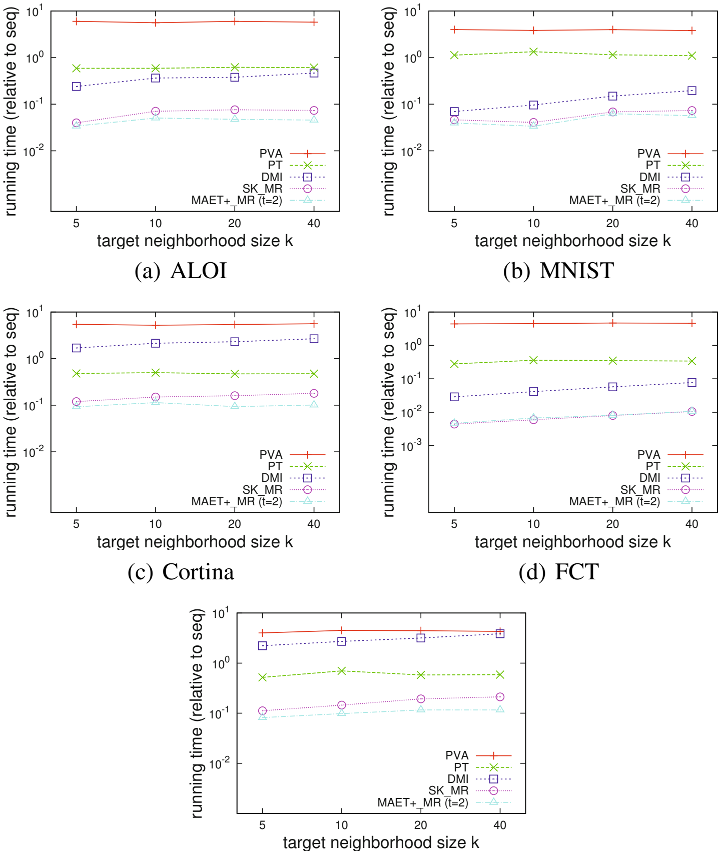
Fig. 6. The results of varying $|F'|$ on all tested datasets, with $k = 10$. The results are exact, except for those of MAET+_MR. The average accuracies of MAET+_MR with $t = 2$ are approximately 92%, 90%, 88%, 97% and 90% for ALOI, MNIST, Cortina, FCT and ANN_SIFT, respectively.

as the number of subspace dimensions increases. In contrast, our algorithms avoid expensive aggregation by restricting the processing to a single query dimension.

Relative to SK_MR, we observe that for high subspace dimensionality, MAET+_MR can achieve a significant improvement in running time while still achieving a high level of accuracy. We note that as the value of $|F'|$ increases, the computational cost of all tested methods must eventually tend to that of sequential search, as one would expect due to the curse of dimensionality.

Fig. 7 shows the results of varying k. Again, our proposed methods generally outperform their competitors, with MAET+_MR achieving a slight improvement in running time over SK_MR, at the cost of a slight loss of accuracy. We also observe that the behaviors of all tested methods are quite stable with respect to k.

Finally, Fig. 5 shows the preprocessing costs of all methods considered in our experimentation. While the preprocessing costs of our methods is substantial, the costs are justifiable in light of their improved performance at query time.



(a) ALOI

(b) MNIST

(c) Cortina

(d) FCT

(e) ANN_SIFT

Fig. 7. The results of varying k on all tested datasets, with $|F'| = 8$. The results are exact, except for those of MAET+_MR. The average accuracies of MAET+_MR with $t = 2$ are approximately 88%, 96%, 89%, 98% and 92% for ALOI, MNIST, Cortina, FCT and ANN_SIFT, respectively.

5 Conclusion

We have presented new solutions for the subspace similarity search problem based on multi-step search, utilizing 1-dimensional lower-bounding distances for the efficient pruning of the search space. Our extensive experimental study showed that our algorithms are able to outperform their state-of-the-art competitors (PVA, PT and DMI) for a relatively wide range of subspace dimensions. We have also shown how practical choices of algorithm parameters can be guided by an analysis of sampling properties.

One possible direction for future research may include the investigation of multi-dimensional lower-bounding distances for pruning in multi-step subspace search. Although multi-dimensional distances could provide a tighter lower bound on the target distance, they cover fewer combinations of query dimensions, and thus may be only of limited practicality.

Acknowledgments. This work was partially supported by NSF under Grant 1241976 and by JSPS under Kakenhi Kiban (C) Research Grant 24500135.

References

1. Kriegel, H.P., Kröger, P., Zimek, A.: Subspace Clustering. WIREs Data Mining and Knowl. Discov. 2(4), 351–364 (2012)
2. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. Artif. Intell. 97(1-2), 273–324 (1997)
3. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann (2006)
4. Seidl, T., Kriegel, H.P.: Optimal Multi-Step k-Nearest Neighbor Search. In: SIGMOD, pp. 154–165 (1998)
5. Houle, M., Ma, X., Nett, M., Oria, V.: Dimensional Testing for Multi-step Similarity Search. In: ICDM, pp. 299–308 (2012)
6. Kriegel, H.P., Kröger, P., Schubert, M., Zhu, Z.: Efficient Query Processing in Arbitrary Subspaces Using Vector Approximations. In: SSDBM, pp. 184–190 (2006)
7. Weber, R., Schek, H.J., Blott, S.: A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In: VLDB, pp. 194–205 (1998)
8. Bernecker, T., Emrich, T., Graf, F., Kriegel, H.P., Kröger, P., Renz, M., Schubert, E., Zimek, A.: Subspace Similarity Search Using the Ideas of Ranking and Top-k Retrieval. In: Proc. ICDE Workshop DBRank, pp. 4–9 (2010)
9. Bernecker, T., Emrich, T., Graf, F., Kriegel, H.-P., Kröger, P., Renz, M., Schubert, E., Zimek, A.: Subspace Similarity Search: Efficient k-NN Queries in Arbitrary Subspaces. In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 555–564. Springer, Heidelberg (2010)
10. Guttman, A.: R-trees: a Dynamic Index Structure for Spatial Searching. In: SIGMOD, pp. 47–57 (1984)
11. Lian, X., Chen, L.: Similarity Search in Arbitrary Subspaces Under L_p-Norm. In: ICDE, pp. 317–326 (2008)
12. Korn, F., Sidiropoulos, N., Faloutsos, C., Siegel, E., Protopapas, Z.: Fast Nearest Neighbor Search in Medical Image Databases. In: VLDB, pp. 215–226 (1996)
13. Houle, M., Kashima, H., Nett, M.: Generalized Expansion Dimension. In: Proc. ICDM Workshop PTDM, pp. 587–594 (2012)
14. Chvátal, V.: The Tail of the Hypergeometric Distribution. Discrete Mathematics 25, 285–287 (1979)
15. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The Amsterdam Library of Object Images. International Journal of Computer Vision 61(1), 103–112 (2005)
16. Boujemaa, N., Fauqueur, J., Ferecatu, M., Fleuret, F., Gouet, V., Saux, B.L., Sahbi, H.: IKONA: Interactive Generic and Specific Image Retrieval. In: Proc. Intern. Workshop on Multimedia Content-Based Indexing and Retrieval (2001)
17. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proc. IEEE 86(11), 2278–2324 (1998)
18. Rose, K., Manjunath, B.S.: The Cortina Data Set, `http://www.scl.ece.ucsb.edu/datasets/index.htm`
19. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
20. Jégou, H., Tavenard, R., Douze, M., Amsaleg, L.: Searching in One Billion Vectors: Re-rank with Source Coding. In: ICASSP, pp. 861–864 (2011)
21. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)