Efficient Similarity Search on Vector Sets

Stefan Brecheisen Hans-Peter Kriegel Martin Pfeifle

Institute for Computer Science, University of Munich
{brecheis,kriegel,pfeifle}@dbs.informatik.uni-muenchen.de

Abstract: Similarity search in database systems is becoming an increasingly important task in modern application domains such as multimedia, molecular biology, medical imaging, computer aided design and many others. Whereas most of the existing similarity models are based on feature vectors, there exist some models which use very complex object representations such as trees and graphs. A promising way between too simple and too complex object representations in similarity search are sets of feature vectors. In this paper, we first motivate the use of this modeling approach for complete object similarity search as well as for partial object similarity search. After introducing a distance measure between vector sets, suitable for many different application ranges, we present and discuss different filters which are indispensable for efficient query processing. In a broad experimental evaluation based on artificial and real-world test datasets, we show that our approach considerably outperforms both the sequential scan and metric index structures.

1 Introduction

In the last ten years, an increasing number of database applications has emerged for which efficient and effective support for similarity search is substantial. The importance of similarity search grows in application areas such as multimedia, medical imaging, molecular biology, computer aided engineering, marketing, purchasing assistance, and others.

As distance functions form the foundation of similarity search, we need an object representation which allows efficient and meaningful distance computations. A common approach is to represent an object by a numerical feature vector. In this case, a feature transformation extracts distinguishable characteristics which are represented by numerical values and grouped together in a feature vector. On the basis of such a feature transformation and under the assumption that similarity corresponds to feature distance, it is possible to define a distance function between the corresponding feature vectors as a similarity measure for two data objects. Thus, searching for data objects similar to a given query object is transformed into proximity search in the feature space. Most applications use the Euclidean metric (L_2) to evaluate the feature distance, but there are several other metrics commonly used, e.g. the Manhattan metric (L_1) and the maximum metric (L_∞).

Furthermore, there exist quite a few much more complex similarity models based on graphs [KS03] and trees [KKSS04]. Generally, the more complex and precise these mod-

els are, the more exact are the results of a similarity search, but at the same time, its computation cost rises as well.

In this paper, we present a distance measure for an approach somewhere in between single feature vectors and complex trees and graphs. We model an object by a *set of feature vectors* which is a very suitable object representation for many different application ranges. In order to achieve efficient query processing we present three different lower-bounding filters and discuss their properties.

The remainder of this paper is organized as follows. In Section 2, we motivate the use of vector set represented objects by presenting various application ranges which benefit from this modeling approach. In Section 3, we introduce the minimal matching distance between vector sets which is a suitable distance measure for partial and complete similarity search. In Section 4, we sketch the paradigm of multi-step query processing and present appropriate filter techniques for the minimal matching distance on vector sets. In Section 5, we present the results of our experimental evaluation. We conclude this work in Section 6 with a short summary and a few remarks on future work.

2 Application Ranges for Vector Sets

Using sets of feature vectors is a generalization of the use of just one large feature vector. It is always possible to restrict the model to a feature space, in which a data object will be completely represented by just one feature vector. But in some applications the properties of vector set representations allow us to model the dependencies between the extracted features more precisely. As the development of conventional database systems in the recent two decades has shown, the use of more sophisticated ways to model data can enhance both the effectiveness and efficiency for applications using large amounts of data. Another advantage of using sets of feature vectors is the better storage utilization. It is not necessary to force objects into a common size, if they are represented by sets of different cardinality. In the following, we will shortly sketch different application ranges which benefit from the use of vector set data.

CAD databases. In [KBK+03] voxelized spatial objects were modeled by sets of feature vectors, where each feature vector represents a 3D rectangular cover which approximates the object as good as possible. The vector set representation is able to avoid the problems that occur by storing a set of covers according to a strict order, i.e. in one high-dimensional feature vector. Thereby, it is possible to compare two objects more intuitively compared to the distance calculation in the one-vector model. In a broad experimental evaluation it was shown that the use of sets of feature vectors greatly enhances the quality of the similarity model compared to the use of a single feature vector.

Soccer teams. As another example, let us assume that we want to measure the similarity between two soccer teams. It is beneficial to represent each player by a feature vector and the complete team as a set of feature vectors. A feature vector for one player may

consist of attributes like his age, his salary, the number of goals in the last season, etc. We can compare two players by computing the Euclidean distance between the corresponding feature vectors. This measures the similarity between two players rather well. But, what is a suitable distance for comparing two teams? Assuming we have a team A consisting of 10 very young players having a low salary and having scored only a few goals in the last season. Furthermore, team A has one highly paid, rather experienced and successful player. On the other hand, we have a team B where we have 10 rather old, highly paid successful players and one young low-budget player. If we compare each player of team A to the most similar player in team B and vice versa, this yields that the two teams are very similar. This straightforward approach does not reflect the intuitive notion of similarity. On the other hand, if we compare each player from team A to a different player in team B trying to minimize the average distance between two "matched" players, this results in a very accurate similarity measure.

For partial similarity, it is advisable not to compare all players from team A to a different player in team B, but only the s most similar players. For low values of s, e.g. $s = 2$, the two teams A and B are very similar, as each team has an old player with a high salary and a young low-budget player. In this case, the distance between the teams A and B would be very small. For higher values of s, the two teams become more and more dissimilar. Let us note that for $s = 11$ the two notions of partial and complete similarity coincide. This behavior reflects the intuitive perception of similarity. To sum up, the use of vector sets allows us to adjust the degree of the partial similarity in k discrete steps, if we represent the objects by vector sets of cardinality k.

Further application areas. There exist a lot of further possible application fields for sets of feature vectors, e.g.:

- *stock portfolios*, where each stock is represented by the value of one share, the overall number of shares, how many days ago the shares were bought, the risk category, etc.
- *shopping carts*, where each consumer product corresponds to a feature vector containing the category, the price, the quantity, etc.
- *multimedia CDs*, where each media file is represented by the publisher, the artist, the title, the filesize, the kind of content, etc.

To sum up, sets of feature vectors are a natural way to model a lot of complex real-world objects.

3 Distance Measures on Vector Sets

Effective distance functions which allow both complete and partial similarity search as well as suitable filter techniques for efficient query processing are indispensable for the general use of the powerful concept of "sets of feature vectors".

There are already several distance measures proposed on sets of vectors. In [EM97] the authors survey the following four measures, which are computable in polynomial time: the Hausdorff distance, the sum of minimum distances, the (fair-)surjection distance and the link distance. The Hausdorff distance does not seem to be suitable as a similarity measure, because it relies too much on the extreme positions of the elements of both sets. The last three distance measures are suitable for modeling similarity, but are not metric. This circumstance makes them unattractive, since there are only limited possibilities for processing similarity queries efficiently when using a non-metric distance function. In [EM97], the authors also introduce a method for expanding the distance measures into metrics, but as a side effect the complexity of distance calculation becomes exponential. Furthermore, the possibility to match several elements in one set to just one element in the compared set is questionable in the application areas presented in Section 2.

A distance measure on vector sets that demonstrates to be suitable for defining similarity is based on the *minimum weight perfect matching* of sets. This well known graph problem can be applied here by building a complete bipartite graph $G = (X \cup Y, E)$ between the vector sets X and Y. The weight of each edge $(x, y) \in E$, where $x \in X$ and $y \in Y$, in this graph G is defined by the distance $d(x, y)$. A perfect matching is a subset $M \subseteq E$ that connects each $x \in X$ to exactly one $y \in Y$ and vice versa. A minimum weight perfect matching is a matching with a minimum sum of weights of its edges. Contrary to the second example of Section 2, where we considered vector sets of equal cardinality, i.e. soccer teams consisting of 11 players, there are a lot of application ranges, where objects are naturally represented by a varying number of vectors. Since a perfect matching can only be found for sets of equal cardinality, we need to introduce suitable weights as a penalty for the unmatched vectors when defining a distance measure between objects of varying cardinality.

Definition 1 (permutation of a set)
Let A be any finite set of arbitrary elements. Then π is a mapping that assigns $a \in A$ a unique number $i \in \{1, .., |A|\}$. This is written as $\pi(A) = (a_1, .., a_{|A|})$. The set of all possible permutations of A is denoted by $\Pi(A)$.

Definition 2 (minimal matching distance)
Let $V \subset \mathbb{R}^d$ and let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a distance function between two d-dimensional feature vectors. Furthermore, let $W : V \to \mathbb{R}$ be a weight function for unmatched elements. Then the minimal matching distance $D_{\text{mm}}^{D,W} : 2^V \times 2^V \to \mathbb{R}$ *is defined as follows:*

$$D_{\text{mm}}^{D,W}(X, Y) = \min_{\pi \in \Pi(Y)} \left(\sum_{i=1}^{|X|} D(\vec{x}_i, \vec{y}_{\pi(i)}) + \sum_{i=|X|+1}^{|Y|} W(\vec{y}_{\pi(i)}) \right)$$

The weight function W provides the penalty given to every unassigned element of the set having larger cardinality. Let us note that the minimal matching distance is a specialization of the *netflow distance* which is proven to be a metric in [RB01]. The minimal matching

distance $D_{\text{mm}}^{D,W}$ is a metric, if the distance function D is a metric and the weight function W meets the following conditions:

(1) $W(\vec{x}) > 0$ for $\vec{x} \in V$
(2) $W(\vec{x}) + W(\vec{y}) \geq D(\vec{x}, \vec{y})$ for $\vec{x}, \vec{y} \in V$

The Kuhn-Munkres algorithm [Kuh55, Mun57] can be used to calculate the minimal matching distance in polynomial time. In a primary initialization step, a distance matrix between the two vector sets containing k d-dimensional vectors is computed. If D is an L_p-distance, this initialization takes $O(k^2 d)$ time. The method itself is based on the successive augmentation of an alternating path between both sets. Since it is guaranteed that this path can be expanded by one further match within each step taking $O(k^2)$ time and there is a maximum of k steps, the overall complexity of a distance calculation is $O(k^3 + k^2 d)$ in the worst case.

The minimal matching distance can be adapted for partial similarity search in vector set represented data. The distance measure defined in the following is based on a partial minimal matching. Given two vector sets X and Y, $|X| \leq |Y|$, we only match $s \leq |X|$ vectors to calculate the distance between X and Y.

Definition 3 (partial minimal matching distance)
Let $V \subset \mathbb{R}^d$ and let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a distance function between two d-dimensional feature vectors. Let $s \leq |X|$. Then the partial minimal matching distance $D_{\text{pmm}}^{D,s} : 2^V \times 2^V \to \mathbb{R}$ *is defined as follows:*

$$D_{\text{pmm}}^{D,s}(X, Y) = \min_{\pi_1 \in \Pi(X), \pi_2 \in \Pi(Y)} \left(\sum_{i=1}^{s} D(\vec{x}_{\pi_1(i)}, \vec{y}_{\pi_2(i)}) \right)$$

Unlike the minimal matching distance the partial variant is not a metric. As the Kuhn-Munkres algorithm produces a partial minimal matching in each step as an intermediate result, we can use it to calculate the partial minimal matching distance $D_{\text{pmm}}^{D,s}(X, Y)$. But we have to take into account all $\binom{|X|}{s}$ combinations of vectors in X to match with vectors in Y. Therefore, the time complexity for a single distance calculation is $O(\binom{k}{s} sk^2 + k^2 d)$. Thus, a filtering technique to speed up query processing is essential.

4 Filters for Vector Sets

Complete similarity search on vector set data can be accelerated by using metric index structures, e.g. the M-tree [CPZ97]. For a detailed survey on metric index structures we refer the reader to [CNBYM01]. Another approach is to use the multi-step query processing paradigm which, in contrast to metric index structures, is also suitable for partial

similarity search. The main goal of multi-step query processing is to reduce the number of complex and therefore time consuming distance calculations in the query process. In order to guarantee that there occur no false drops the used filter distances have to fulfill a lower-bounding distance criterion. For any two objects o_1 and o_2, a lower-bounding distance function D_f in the filter step has to return a value that is not greater than the exact object distance D_o of o_1 and o_2, i.e. $D_f(o_1, o_2) \leq D_o(o_1, o_2)$. With a lower-bounding distance function, it is possible to safely filter out all database objects which have a filter distance greater than the current query range because the exact similarity distance of those objects cannot be less than the query range.

The computation of the minimal matching distance on vector sets is a rather expensive operation. Thus, the employment of selective and efficiently computable filter distance functions for similarity search is very important. In the following, we present three different filter types for query processing on data objects represented by vector sets, namely the *closest pair filter*, the *centroid filter* and the *norm vector filter*.

4.1 Closest Pair Approach

The *closest pair distance* between two vector sets X and Y can be used as a filter distance for the minimal matching distance $D_{\text{mm}}^{D,W}$ and is defined as follows.

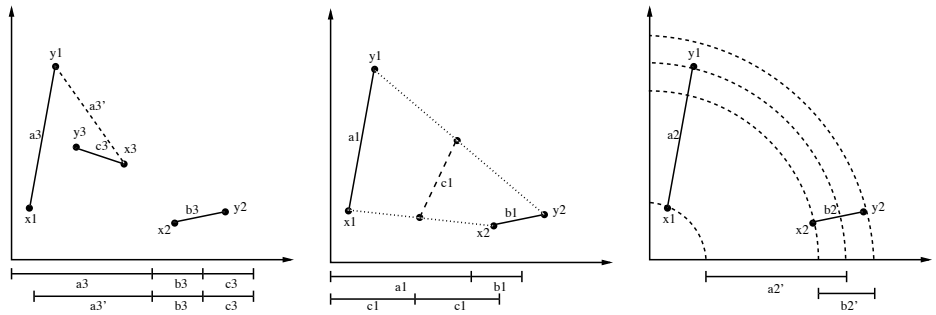
Definition 4 (closest pair distance)
Let $V \subset \mathbb{R}^d$ and $\vec{\omega} \in \mathbb{R}^d \setminus V$. Let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a distance function. Let $X' = \{\vec{x}_1, \ldots \vec{x}_{|Y|}\}$ be a multiset where $\vec{x}_i = \vec{\omega}$ for $i \in \{|X|+1, \ldots |Y|\}$. Then the closest pair distance $D_{\text{cp}}^{D,\vec{\omega}}(X, Y) : 2^V \times 2^V \to \mathbb{R}$ *is defined as follows.*

$$D_{\text{cp}}^{D,\vec{\omega}}(X, Y) = \max \left(\sum_{i=1}^{|Y|} \min_{j=1,\ldots|Y|} D(\vec{x}_i, \vec{y}_j), \sum_{i=1}^{|Y|} \min_{j=1,\ldots|Y|} D(\vec{x}_j, \vec{y}_i) \right)$$

Let us note that the closest pair filter works directly on the set of vectors, i.e. on the original data, and not on approximated data. The filter distance can be computed by scanning the matrix of distance values between each pair of vectors in X and Y for the closest pairs. We will now show that the closest pair distance between two vector sets is a lower bound for the minimal matching distance.

Theorem 1 *Let $V \subset \mathbb{R}^d$ and $\vec{\omega} \in \mathbb{R}^d \setminus V$. Let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a distance function. Furthermore, let $W_{\vec{\omega}} : V \to \mathbb{R}$, $W_{\vec{\omega}}(\vec{v}) = D(\vec{v}, \vec{\omega})$, be a weight function for unmatched elements. Then the following inequality holds:*

$$D_{\text{cp}}^{D,\vec{\omega}}(X, Y) \leq D_{\text{mm}}^{D,W_{\vec{\omega}}}(X, Y)$$



(a) Closest pair. (b) Centroid. (c) Norm vector.

Figure 1: Filters for the minimal matching distance.

Proof: See Appendix A.1.

A 2-dimensional example for the closest pair filter is depicted in Fig. 1(a), where $|X| = |Y| = 3$ and $a'_3 + b_3 + c_3 = D_{\text{cp}}^{L_2,\vec{0}}(X,Y) \leq D_{\text{mm}}^{L_2,W_{\vec{0}}}(X,Y) = a_3 + b_3 + c_3$. As $a'_3 < a_3$, $\vec{x}_3$ is matched to both $\vec{y}_1$ and $\vec{y}_3$ during the filter distance calculation, whereas the minimal matching distance is based on one-to-one matchings.

We adapt the closest pair filter to partial similarity search by adding up just the distances of the s closest pairs of vectors. Thus, the *partial closest pair distance* is defined as follows.

Definition 5 (partial closest pair distance)
Let $V \subset \mathbb{R}^d$ and let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a distance function. Let $s \leq |X|$. Then the partial closest pair distance $D_{\text{pcp}}^{D,s}(X,Y) : 2^V \times 2^V \to \mathbb{R}$ *is defined as follows.*

$$D_{\text{pcp}}^{D,s}(X,Y) = \max\left(\min_{\pi \in \Pi(X)} \sum_{i=1}^{s} \min_{j=1,\ldots|Y|} D(\vec{x}_{\pi(i)}, \vec{y}_j), \right.$$
$$\left. \min_{\pi \in \Pi(Y)} \sum_{i=1}^{s} \min_{j=1,\ldots|X|} D(\vec{x}_j, \vec{y}_{\pi(i)}) \right)$$

The partial closest pair distance is a lower bound for the partial minimal matching distance.

Theorem 2 *Let $V \subset \mathbb{R}^d$ and let $X, Y \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a distance function. Let $s \leq |X|$. Then the following inequality holds:*

$$D_{\text{pcp}}^{D,s}(X,Y) \leq D_{\text{pmm}}^{D,s}(X,Y)$$

Proof: Analogous to the proof of Theorem 1.

As the partial closest pair distance can be computed rather efficiently by scanning the matrix of distance values between each pair of vectors in X and Y for the closest pairs and organizing the s closest distances in a heap structure, it is a very beneficial filter for the partial minimal matching distance. The overall runtime complexity is $O(k^2 d)$ for the complete version and $O(k^2 d \log s)$ for the partial version of the clostest pair distance, when an L_p-distance is used between vectors. Although this is more complex than the closest pair approach on norm vectors (cf. Section 4.3), it is a more selective filter that saves more of the very expensive calculations of the exact partial minimal matching distance.

4.2 Centroid Approach

This filter step is based on the relation between a set of feature vectors and its *extended centroid* [KBK+03].

Definition 6 (extended centroid)
Let $V \subset \mathbb{R}^d$ and $\vec{\omega} \in \mathbb{R}^d \setminus V$. Let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\} \in 2^V$ be a vector set where $|X| \leq k$. Then the extended centroid $C_{k,\vec{\omega}}(X)$ *is defined as follows:*

$$C_{k,\vec{\omega}}(X) = \frac{\sum_{i=1}^{|X|} \vec{x}_i + (k - |X|)\,\vec{\omega}}{k}$$

Note how the vector $\vec{\omega}$ is used as a "dummy" vector to fill up vector sets with a cardinality of less than k.

Theorem 3 *Let $V \subset \mathbb{R}^d$ and $\vec{\omega} \in \mathbb{R}^d \setminus V$. Let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}, Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets where $|X|, |Y| \leq k$ and let $C_{k,\vec{\omega}}(X)$, $C_{k,\vec{\omega}}(Y)$ be their extended centroids. Furthermore, let $W_{\vec{\omega}} : V \to \mathbb{R}$, $W_{\vec{\omega}}(\vec{v}) = \|\vec{v} - \vec{\omega}\|_p$, be a weight function for unmatched elements. Then the following inequality holds:*

$$k \,\|C_{k,\vec{\omega}}(X) - C_{k,\vec{\omega}}(Y)\|_p \leq D_{\text{mm}}^{L_p, W_{\vec{\omega}}}(X, Y)$$

See [KBK+03] for the proof of this theorem. We have shown that the L_p-distance between the extended centroids multiplied by k is a lower bound for the minimal matching distance under the named preconditions. Therefore, when computing e.g. ε-range queries, we do not need to examine objects whose extended centroids have a distance to the query object q that is larger than $\frac{\varepsilon}{k}$. Often a good choice of $\vec{\omega}$ is $\vec{0}$, since $\vec{0} \notin V$ holds for a lot of applications. Thus, Conditions (1) and (2) for the metric character of the minimal matching distance $D_{\text{mm}}^{L_2, W_{\vec{0}}}$ are satisfied. A 2-dimensional example for the extended centroid filter is depicted in Fig. 1(b), where $|X| = |Y| = 2$ and $2c_1 = 2\,\|C_{k,\vec{0}}(X) - C_{k,\vec{0}}(Y)\|_2 \leq D_{\text{mm}}^{L_2, W_{\vec{0}}}(X, Y) = a_1 + b_1$.

The centroid approach is not suitable as a filter for the partial minimal matching distance, as the centroid invariably aggregates information of all vectors contained in a vector set.

4.3 Norm Vector Approach

Another possible filter for vector set represented data is based on the L_p-norms of all vector elements of a vector set. The idea is as follows: For all vectors $\vec{x}$ in a vector set X, $|X| \leq k$, we compute the L_p-norms $\|\vec{x}\|_p$ and organize these norm values in descending order in a k-dimensional vector. We call this filter the *norm vector filter*.

Definition 7 (norm vector)
Let $V \subset \mathbb{R}^d$. Let $X \in 2^V$ be a vector set where $|X| \leq k$. Let $(\|\vec{x}_1\|_p, \ldots \|\vec{x}_{|X|}\|_p)$ be the sequence of the L_p-norm values of the vectors in X in descending order, i.e. for all $i < j \in \{1, \ldots |X|\}$ holds $\|\vec{x}_i\|_p \geq \|\vec{x}_j\|_p$. Then the norm vector *$V_k(X) = (v_1, \ldots v_k)^t \in \mathbb{R}^k$ is defined as follows:*

$$v_i = \begin{cases} \|\vec{x}_i\|_p & \text{for } i = 1, \ldots |X| \\ 0 & \text{for } i = |X|+1, \ldots k \end{cases}$$

Note that if X has a cardinality smaller than k, dimensions $|X|+1$ to k of the norm vector will get filled with 0. We employ the Manhattan distance as a distance function between two norm vectors $V_k(X)$ and $V_k(Y)$. This distance measure fulfills the lower-bounding property with respect to the minimal matching distance, if the L_p-norm is used as the weight function W.

Theorem 4 *Let $V \subset \mathbb{R}^d$ and let $X, Y \in 2^V$ be two vector sets. Their norm vectors are denoted by $V_k(X)$ and $V_k(Y)$. Furthermore, let $W_{\vec{0}} : V \to \mathbb{R}$, $W_{\vec{0}}(\vec{v}) = \|\vec{v}\|_p$, be the L_p-norm used as a weight function for the minimal matching distance. Then the following inequality holds:*

$$\|V_k(X) - V_k(Y)\|_1 \leq D_{\text{mm}}^{L_p, W_{\vec{0}}}(X, Y)$$

Proof: See Appendix A.2.

A 2-dimensional example for the norm vector filter is depicted in Fig. 1(c), where $|X| = |Y| = 2$ and $a'_2 + b'_2 = \|V_k(X) - V_k(Y)\|_1 \leq D_{\text{mm}}^{L_2, W_{\vec{0}}}(X, Y) = a_2 + b_2$.

An approach for partial similarity search is to apply a parallel scan through the norm vectors $V_k(X)$ and $V_k(Y)$ and to build a heap structure containing the distances between the closest pairs of norm values found during the parallel scan. Finally, the sum of the top s elements of the heap is reported as the distance measure. This can be done very efficiently in $O(k \log s)$ time using the algorithm in Fig. 2. The algorithm corresponds to a closest pair approach on the norm values of the feature vectors, which lower bounds the partial minimal matching distance.

```
algorithm partialNormVectorFilter(VectorSet X, VectorSet Y, Integer k, Integer s)
begin
    return max(comp(X, Y, k, s), comp(Y, X, k, s));
end;

algorithm comp(VectorSet X, VectorSet Y, Integer k, Integer s)
begin
    (x_1, ... x_k) := V_k(X);                          // initialize
    (y_1, ... y_k) := V_k(Y);
    j := 1;
    for i in 1..k do                                   // parallel scan
        while j < k ∧ |x_i − y_j| ≥ |x_i − y_{j+1}| do
            j := j + 1;
        end while;
        heap.insert(|x_i − y_j|);
    end for;
    dist := 0;                                         // add up the distance
    for i in 1..s do
        dist := dist + heap.top();
    end for;
    return dist;
end;
```

Figure 2: Partial norm vector filter algorithm.

Theorem 5 *Let $V \subset \mathbb{R}^d$ and let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $s \leq |X|$. Let $\hat{X} = \{\|\vec{x}_1\|_p, \ldots \|\vec{x}_{|X|}\|_p\}$, $\hat{Y} = \{\|\vec{y}_1\|_p, \ldots \|\vec{y}_{|Y|}\|_p\}$ be multisets containing the L_p-norm values of the vectors in X and Y. Then the following inequality holds:*

$$D_{\mathrm{pcp}}^{L_p,s}(\hat{X}, \hat{Y}) \leq D_{\mathrm{pmm}}^{L_p,s}(X, Y)$$

Proof: See Appendix A.3.

4.4 Summary

As the computation of the minimal matching distance is rather time-consuming, we introduced three different filters. The centroid and the norm vector filtering techniques can be profitably combined. The exact distance computation is only performed if the results of both filter distance computations on the centroids and the norm vectors are small enough. This way, a good deal of the information in the vector sets is incorporated in the filter distance computation. Given d-dimensional data, the centroid filter maps each dimension to a single value, resulting in a d-dimensional vector. On the other hand, the norm vector filter maps each vector to a single value resulting in a k-dimensional vector. Thus, the combined filter contains aggregated information over both the dimensions and the vectors and is therefore suitable for a lot of different data distributions. The time complexity for a

Table 1: Runtime complexity of the proposed filters.

	exact distance	closest pair	centroid	norm vector
complete similarity	$O(k^3 + k^2 d)$	$O(k^2 d)$	$O(d)$	$O(k)$
partial similarity	$O(\binom{k}{s} s k^2 + k^2 d)$	$O(k^2 d \log s)$	n/a	$O(k \log s)$

combined filter distance evaluation is $O(d+k)$. As the centroid approach is not applicable for partial similarity search, we cannot use the combined filter for this purpose.

In contrast to the other two approaches, which derive a single feature vector for approximating a vector set, the closest pair filter works directly on the vector sets. The resulting distance measure lower bounds the minimal matching distance and can be computed more efficiently than the exact minimal matching distance. The runtime complexities for partial and complete similarity distance calculations based on the three different filters are summed up in Table 1, where we assume vector sets containing k d-dimensional vectors, a partial similarity parameter $s \in \{1, \ldots k\}$, and an L_p-distance between vectors.

5 Experimental Evaluation

In this section, we present our experimental results. We generated and used two artificial datasets, each containing 100,000 random vector sets. The first dataset consists of vector sets containing 10 2-dimensional vectors each. The other dataset consists of vector sets containing 2 10-dimensional vectors each. The vectors are generated so that all of their components are uniformly distributed in the interval between 0 and 1. All distance measures between vector sets were implemented in Java 1.4 and the experiments were run on a workstation with a Xeon 2.4 GHz processor and 2 GB main memory under Linux.

Furthermore, we used the similarity model presented in [KBK$^+$03], where CAD objects were represented by a vector set consisting of either 3, 5 or 7 vectors in 6D. All experiments were carried out on a dataset containing 5,000 CAD objects from an American aircraft producer. We conducted our experiments on top of the Oracle9i Server using PL/SQL for the computational main memory based programming. We compared our different filters for vector set represented data to a PL/SQL implementation of the M-tree [CPZ97]. For the M-tree based k-nearest neighbor queries the ranking algorithm of [HS95] was used. The experiments were performed on a Pentium III/700 machine with IDE hard drives. The database block cache was set to 500 disk blocks with a block size of 8 KB and was used exclusively by one active session.

The minimal matching distances between sets of feature vectors were computed using an implementation of the Kuhn-Munkres algorithm. Throughout our experiments we used the Euclidean distance as the distance measure between two single vectors. The range queries were based on a sequential scan. The k-nn queries with exact distance calculations were

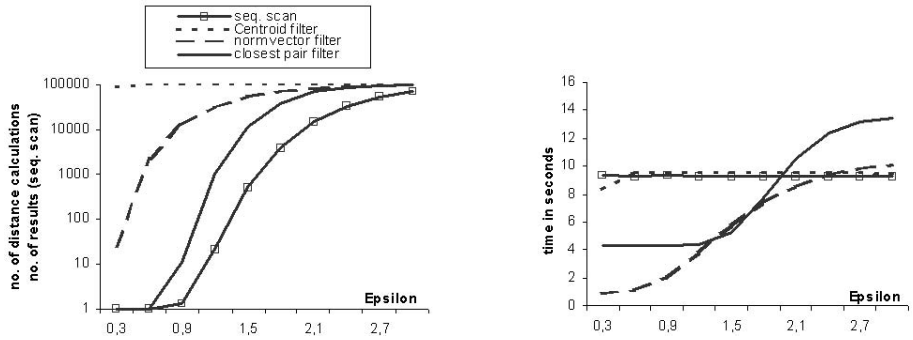


Figure 3: Complete range queries, artificial dataset, cardinality 10, dimensionality 2.

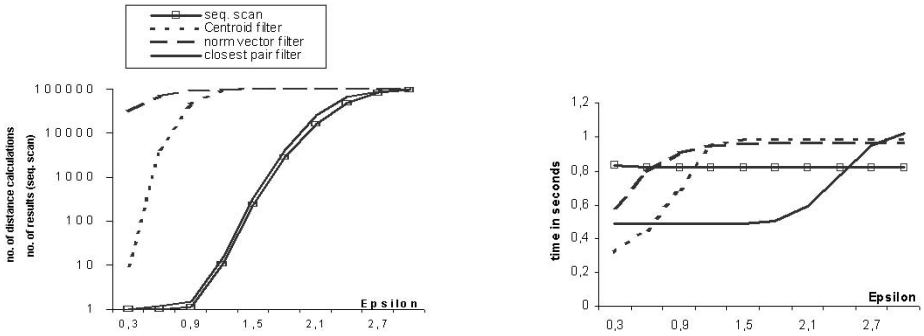


Figure 4: Complete range queries, artificial dataset, cardinality 2, dimensionality 10.

also based on a sequential scan. For the filtered k-nn queries the filter distances between the query object and all vector sets in the database were calculated and sorted in ascending order. Then the optimal multi-step k-nn search algorithm [SK98] was used. In all tests, we processed 10 different similarity range queries as well as k-nn queries. The presented figures depict the average results from these tests.

5.1 Complete Similarity Search

In a first experiment, we carried out range queries on the two artificial datasets. Figure 3 shows rather good results for the norm vector filter, while the centroid filter performs rather badly. The superiority of the norm vector filter is due to the fact that more information is preserved by approximating a vector set by a 10-dimensional vector in contrast to the 2-dimensional centroid computed by the centroid approach. As expected, the situation is reversed in Fig. 4 where each vector set contains 2 10-dimensional vectors. In both tests, the closest pair filter has good to optimal selectivity, but due to its computational

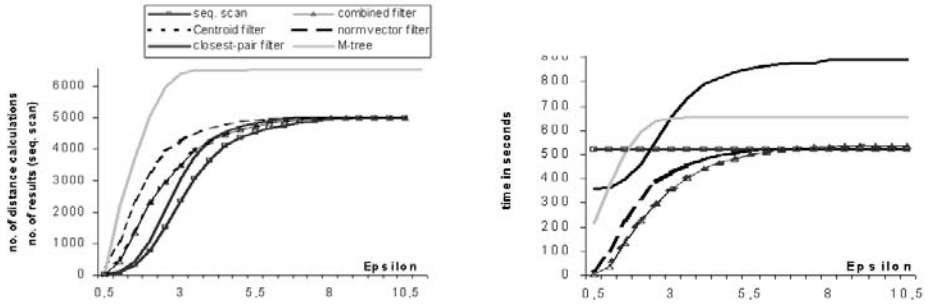


Figure 5: Complete range queries, CAD dataset, cardinality 5, dimensionality 6.

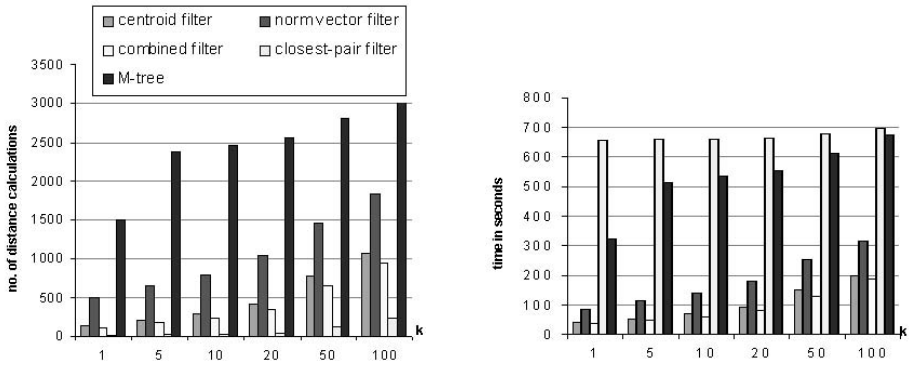


Figure 6: Complete k-nn queries, CAD dataset, cardinality 7, dimensionality 6 (sequential scan took about 1014 sec. for each k).

complexity the overall runtime is rather high especially for high ε-values.

Using the CAD datasets, we carried out different range queries on a vector set consisting of 5 6-dimensional vectors. Figure 5 shows that the selectivity of the closest pair filter is almost optimal, i.e. few unnecessary candidates are produced. Nevertheless, the overall runtime of this filter-step is very high as the runtime complexity of the filter-step is almost as high as the computation of the minimal matching distance itself (cf. Fig. 5). Good results were obtained by using the centroid approach. The good performance of the centroid approach can slightly be increased by using the combined filter, i.e. the combination of the norm vector filter and the centroid filter, which can also be efficiently computed and has a slightly higher selectivity. Note that both the selectivity as well as the runtime behavior of the M-tree are outperformed by this combined filter for all ε-values.

Figure 6 shows the average results we obtained for carrying out different k-nn queries on CAD objects represented by vector sets containing 7 vectors. Basically, we made the same observations as for range queries. Although the closest pair filter has a rather good

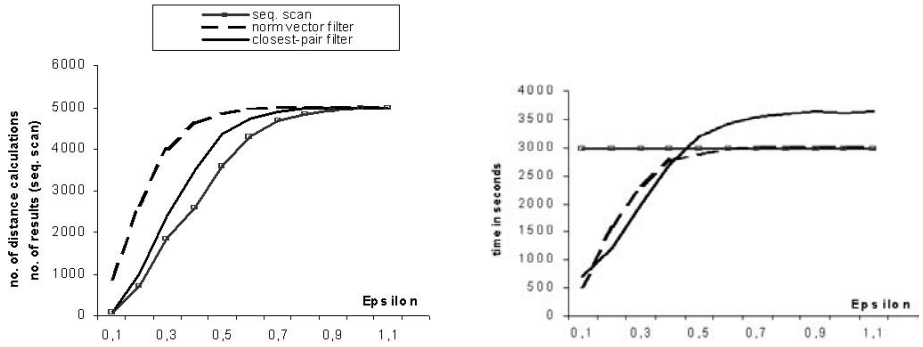


Figure 7: Partial range queries for $s = 2$, CAD dataset, cardinality 7, dimensionality 6.

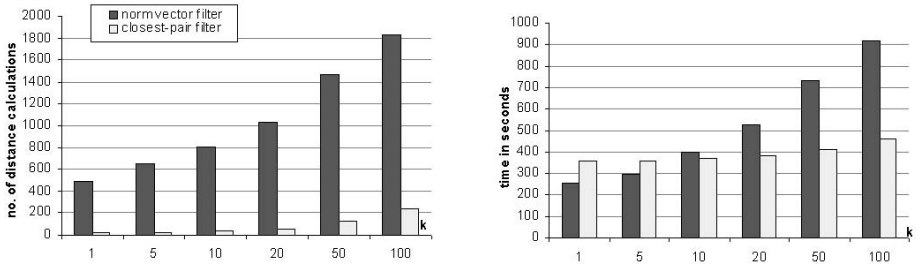


Figure 8: Partial k-nn queries for $s = 3$, CAD dataset, cardinality 5, dimensionality 6 (sequential scan took about 2123 sec. for each k).

selectivity, it is rather expensive. The best trade off is achieved by using the combination of the norm vector filter and the centroid filter. All filters have a rather good selectivity and accelerate the query process enormously. For instance, for k-nn queries where k is smaller than 20, the combined filter accelerates the query process on the 6-dimensional vector sets by more than one order of magnitude compared to the sequential scan. Again, the selectivity as well as the runtime behavior of the M-tree is clearly outperformed by this combined filter for all values of k, e.g. for k=5 the combined filter outperforms the M-tree by an order of magnitude. We made the same observations for the CAD datasets with 3 and 5 vectors per vector set, except that the absolute runtime is higher for the larger vector sets. The average runtime for 7 vectors is about four times the average runtime for 3 vectors.

5.2 Partial Similarity Search

In this section, we tested the closest pair algorithm on L_2-norm vectors, called norm vector filter, and directly on the d-dimensional vectors, called closest pair filter. Let us note that

detecting partial similarity is a very expensive operation. Furthermore, we cannot apply the M-tree as the distance function is not a metric (cf. Definition 3).

Figure 7 shows the average of 10 range queries for varying ε-values on a vector set of 7 vectors. The partial similarity parameter s was set to 2. Again, the closest pair filter is very selective. As the exact distance function is very expensive, the closest pair filter can be beneficially used for small ε-values. For higher ε-values, the rather high evaluation cost of the closest pair filter carry into weight. On the other hand, the norm vector can safely be used for all values of ε, as there is no noteworthy overhead. For rather small ε-values, it even outperforms the closest pair filter, although the norm vector has a lower selectivity than the closest pair filter. This is because the lower computational cost of the norm vector filter still pays off, compared to the slightly more exact distance computations which have to be carried out.

Figure 8 shows the average of 10 k-nn queries for vector sets of 5 vectors each having a dimensionality of 6 and a partial similarity parameter $s = 3$. For small values of k, the norm vector filter outperforms the exact distance computation by almost one order of magnitude. For higher values of k, the selectivity of the norm vector filter decreases and thus the overall response time increases. For values of k equal to 100, the norm vector filter still accelerates the query process by 100%. As already mentioned, the closest pair filter is rather expensive. Although it has an excellent selectivity, the norm vector filter is better for rather small values of k. For increasing values of k, the closest pair filter outperforms the norm vector filter because of the much better selectivity and the very expensive exact distance calculations.

6 Conclusions

In this paper, we motivated the use of vector set data by pointing out the different application areas of this promising representation technique. We introduced a suitable distance function on vector sets, which reflects the intuitive notion of similarity for the presented application ranges. Furthermore, we presented different filtering techniques with different runtime complexities. Our experimental evaluation and our analytical reasoning showed that the closest pair filter is the most selective filter. As this filter is rather expensive, it only pays off for partial similarity queries which are extremely expensive themselves. For complete similarity queries, the combination of the norm vector filter and the centroid filter is the method of choice for a lot of different data distributions, as it can be computed efficiently and the information of each vector and each dimension is taken into consideration. The experimental evaluation on real world datasets demonstrates that the presented filtering techniques accelerate similarity range queries and k-nn queries by up to one order of magnitude compared to metric index structures and the sequential scan.

In our future work, we want to show how the paradigm of "sets of feature vectors" can be applied to effective and efficient data mining tasks, e.g. clustering and classification.

References

[CNBYM01] E. Chávez, G. Navarro, R. Beaza-Yates, and J. Marroquín. "Search in Metric Spaces". *ACM Computing Surveyes*, 33(3):273–321, 2001.

[CPZ97] P. Ciaccia, M. Patella, and P. Zezula. "M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces". In *Proc. 23rd Int. Conf. of Very Large Data Bases, Athens, Greece*, pages 426–435, 1997.

[EM97] T. Eiter and H. Mannila. "Distance Measures for Point Sets and Their Computation". *Acta Informatica*, 34(2):103–133, 1997.

[HS95] G. R. Hjaltason and H. Samet. "Ranking in Spatial Databases". In *Proc. 4th Int. Symposium on Large Spatial Databases (SSD'95)*, volume 951 of *Lecture Notes in Computer Science (LNCS)*, pages 83–95. Springer, 1995.

[KBK+03] H.-P. Kriegel, S. Brecheisen, P. Kröger, M. Pfeifle, and M. Schubert. "Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects". In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'03), San Diego, CA*, 2003.

[KKSS04] K. Kailing, H.-P. Kriegel, S. Schönauer, and T. Seidl. "Efficient Similarity Search for Hierarchical Data in Large Databases". In *Proc. 9th Int. Conf. on Extending Database Technology (EDBT'04), Heraklion, Greece*, 2004.

[KS03] H.-P. Kriegel and S. Schönauer. "Similarity Search in Structured Data". In *Proc. 5th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'03), Prague, Czech Republic*, volume 2737 of *Lecture Notes in Computer Science (LNCS)*, pages 309–319. Springer, 2003.

[Kuh55] H. W. Kuhn. "The Hungarian method for the assignment problem". *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[Mun57] J. Munkres. "Algorithms for the assignment and transportation problems". *Journal of the SIAM*, 6:32–38, 1957.

[RB01] J. Ramon and M. Bruynooghe. "A polynomial time computable metric between point sets". *Acta Informatica*, 37:765–780, 2001.

[SK98] T. Seidl and H.-P. Kriegel. "Optimal Multi-Step k-Nearest Neighbor Search". In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 154–165, 1998.

Appendix

A Formal Proofs

We will use the following three lemmas to proof Theorems 1, 4 and 5.

Lemma 1 *Let $\vec{x}, \vec{y} \in \mathbb{R}^d$ be two d-dimensional feature vectors. Then the difference between the L_p-norms of $\vec{x}$ and $\vec{y}$ underestimates the L_p-distance between $\vec{x}$ and $\vec{y}$:*

$$\big|\|\vec{x}\|_p - \|\vec{y}\|_p\big| \leq \|\vec{x} - \vec{y}\|_p$$

Proof: $\|\vec{x}\|_p = \|\vec{x} - \vec{0}\|_p \overset{\text{tri. ineq.}}{\leq} \|\vec{x} - \vec{y}\|_p + \|\vec{y} - \vec{0}\|_p = \|\vec{x} - \vec{y}\|_p + \|\vec{y}\|_p$ follows $\|\vec{x}\|_p - \|\vec{y}\|_p \leq \|\vec{x} - \vec{y}\|_p$.

$\|\vec{y}\|_p = \|\vec{y} - \vec{0}\|_p \overset{\text{tri. ineq.}}{\leq} \|\vec{x} - \vec{y}\|_p + \|\vec{x} - \vec{0}\|_p = \|\vec{x} - \vec{y}\|_p + \|\vec{x}\|_p$ follows $\|\vec{y}\|_p - \|\vec{x}\|_p \leq \|\vec{x} - \vec{y}\|_p$.

Then $\big|\|\vec{x}\|_p - \|\vec{y}\|_p\big| = \max(\|\vec{x}\|_p - \|\vec{y}\|_p, \|\vec{y}\|_p - \|\vec{x}\|_p) \leq \|\vec{x} - \vec{y}\|_p$. □

Lemma 2 *Let $V \subset \mathbb{R}^d$. Let $X = \{\vec{x}_1, \ldots \vec{x}_{|X|}\}$, $Y = \{\vec{y}_1, \ldots \vec{y}_{|Y|}\} \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Then the following inequality holds:*

$$\sum_{i=1}^{|X|} \big|\|\vec{x}_i\|_p - \|\vec{y}_i\|_p\big| \leq \sum_{i=1}^{|X|} \|\vec{x}_i - \vec{y}_i\|_p$$

Proof: The proposition holds if $\forall i \in \{1, \ldots |X|\} : \big|\|\vec{x}_i\|_p - \|\vec{y}_i\|_p\big| \leq \|\vec{x}_i - \vec{y}_i\|_p$ and this follows directly from Lemma 1. □

Lemma 3 *Let $V \subset \mathbb{R}^d$ and let $X, Y \in 2^V$ be two vector sets. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Their norm vectors are denoted by $V_k(X)$ and $V_k(Y)$. Let the sequences of the L_p-norm values of the vectors in X and Y in descending order be denoted by $(\|\vec{x}_1\|_p, \ldots \|\vec{x}_{|X|}\|_p)$ and $(\|\vec{y}_1\|_p, \ldots \|\vec{y}_{|Y|}\|_p)$. Let $\pi \in \Pi(Y)$. Then the following inequality holds:*

$$\|V_k(X) - V_k(Y)\|_1 \leq \sum_{i=1}^{|X|} \big|\|\vec{x}_i\|_p - \|\vec{y}_{\pi(i)}\|_p\big| + \sum_{i=|X|+1}^{|Y|} \|\vec{y}_{\pi(i)}\|_p$$

Proof: (Sketch) Let $V_k(X) = (x_1, \ldots x_k)^t$, $V_k(Y) = (y_1, \ldots y_k)^t$.
We first show that the following holds:

$$\|V_k(X) - V_k(Y)\|_1 = \sum_{i=1}^{k} |x_i - y_i| \leq \sum_{i=1}^{k} |x_i - y_{\pi(i)}| \qquad (*)$$

Every given permutation π can be constructed from adjacent permutations $\pi_1, \dots \pi_n$, such that $\pi = \pi_1 \circ \dots \circ \pi_n$ and for each π_l there is some $q \in \{1, \dots |X|\}$, such that $\pi_l(q) = q+1$, $\pi_l(q+1) = q$ and $\forall q' \notin \{q, q+1\} : \pi_l(q') = q'$. Given π_l, we show that $|x_q - y_{\pi_l(q)}| + |x_{q+1} - y_{\pi_l(q+1)}| \geq |x_{q+1} - y_{q+1}| + |x_q - y_q|$. There are in total six cases, because of the ordering within the norm vectors:

1. $x_q \leq x_{q+1} \leq y_{\pi_l(q+1)} \leq y_{\pi_l(q)}$
2. $x_q \leq y_{\pi_l(q+1)} \leq x_{q+1} \leq y_{\pi_l(q)}$
3. $x_q \leq y_{\pi_l(q+1)} \leq y_{\pi_l(q)} \leq x_{q+1}$
4. $y_{\pi_l(q+1)} \leq x_q \leq x_{q+1} \leq y_{\pi_l(q)}$
5. $y_{\pi_l(q+1)} \leq x_q \leq y_{\pi_l(q)} \leq x_{q+1}$
6. $y_{\pi_l(q+1)} \leq y_{\pi_l(q)} \leq x_q \leq x_{q+1}$

We exemplarily show the third case. The proofs of the other five cases are very similar.

$$|x_q - y_{\pi_l(q)}| + |x_{q+1} - y_{\pi_l(q+1)}| = x_{q+1} - y_q + y_{q+1} - x_q =$$
$$(x_{q+1} - y_{q+1}) + (y_{q+1} - y_q) + (y_q - x_q) + (y_{q+1} - y_q) =$$
$$|x_{q+1} - y_{q+1}| + |x_q - y_q| + 2|y_{q+1} - y_q| \geq |x_{q+1} - y_{q+1}| + |x_q - y_q|$$

As for each application of a π_l the sum on the right side of proposition (*) will grow or remain equal, the sum will grow or remain equal when applying π. Thus, proposition (*) holds. Then the following holds:

$$\|V_k(X) - V_k(Y)\|_1 \overset{(*)}{\leq} \sum_{i=1}^{k} |x_i - y_{\pi(i)}| = \sum_{i=1}^{|X|} \big| \|\vec{x}_i\|_p - \|\vec{y}_{\pi(i)}\|_p \big| +$$
$$\sum_{i=|X|+1}^{|Y|} \big| \|0\|_p - \|\vec{y}_{\pi(i)}\|_p \big| + \sum_{i=|Y|+1}^{k} \big| \|0\|_p - \|0\|_p \big| =$$
$$\sum_{i=1}^{|X|} \big| \|\vec{x}_i\|_p - \|\vec{y}_{\pi(i)}\|_p \big| + \sum_{i=|X|+1}^{|Y|} \|\vec{y}_{\pi(i)}\|_p$$

□

A.1 Theorem 1

Proof: Let $\pi \in \Pi(Y)$ be the permutation of Y that results from the minimum weight perfect matching of X and Y, i.e.

$$D_{\text{mm}}^{D,W_{\vec{\omega}}}(X, Y) = \sum_{i=1}^{|X|} D(\vec{x}_i, \vec{y}_{\pi(i)}) + \sum_{i=|X|+1}^{|Y|} D(\vec{\omega}, \vec{y}_{\pi(i)})$$

The proof consists of two cases.
(1) $D_{\text{cp}}^{D,\vec{\omega}}(X, Y) = \sum_{i=1}^{|Y|} \min_{j=1,\dots|Y|} D(\vec{x}_i, \vec{y}_j)$.

$$\sum_{i=1}^{|Y|} \min_{j=1,\dots|Y|} D(\vec{x}_i, \vec{y}_j) =$$
$$\sum_{i=1}^{|X|} \min_{j=1,\dots|Y|} D(\vec{x}_i, \vec{y}_j) + \sum_{i=|X|+1}^{|Y|} \min_{j=1,\dots|Y|} D(\vec{\omega}, \vec{y}_j) \leq$$
$$\sum_{i=1}^{|X|} D(\vec{x}_i, \vec{y}_{\pi(i)}) + \sum_{i=|X|+1}^{|Y|} D(\vec{\omega}, \vec{y}_{\pi(i)})$$

The inequality holds, if it holds for every pair of i-th addends. This is obviously the case, as we always pick the $\vec{y}_j \in Y$ which minimizes $D(\vec{x}_i, \vec{y}_j)$.

(2) $D_{\mathrm{cp}}^{D,\vec{\omega}}(X,Y) = \sum_{i=1}^{|Y|} \min_{j=1,\ldots|Y|} D(\vec{x}_j, \vec{y}_i)$.

$$\sum_{i=1}^{|Y|} \min_{j=1,\ldots|Y|} D(\vec{x}_j, \vec{y}_i) = \sum_{i=1}^{|Y|} \min_{j=1,\ldots|Y|} D(\vec{x}_j, \vec{y}_{\pi(i)}) =$$
$$\sum_{i=1}^{|X|} \min_{j=1,\ldots|Y|} D(\vec{x}_j, \vec{y}_{\pi(i)}) + \sum_{i=|X|+1}^{|Y|} \min_{j=1,\ldots|Y|} D(\vec{x}_j, \vec{y}_{\pi(i)}) \leq$$
$$\sum_{i=1}^{|X|} D(\vec{x}_i, \vec{y}_{\pi(i)}) + \sum_{i=|X|+1}^{|Y|} D(\vec{\omega}, \vec{y}_{\pi(i)})$$

Again, the inequality holds, if it holds for every pair of i-th addends. This is obviously the case, as we always pick the $\vec{x}_j \in X'$ which minimizes $D(\vec{x}_j, \vec{y}_{\pi(i)})$ (note that $\vec{\omega} \in X'$ if $|X| < |Y|$). □

A.2 Theorem 4

Proof: Let the sequences of the L_p-norm values of the vectors in X and Y in descending order be denoted by $(\|\vec{x}_1\|_p, \ldots \|\vec{x}_{|X|}\|_p)$ and $(\|\vec{y}_1\|_p, \ldots \|\vec{y}_{|Y|}\|_p)$. We assume w.l.o.g. $|X| \leq |Y| \leq k$. Let $\pi \in \Pi(Y)$ be the permutation of Y that results from the minimum weight perfect matching of X and Y. We combine the results from Lemmas 2 and 3.

$$\|V_k(X) - V_k(Y)\|_1 \overset{\text{Lemma 3}}{\leq}$$
$$\sum_{i=1}^{|X|} \left| \|\vec{x}_i\|_p - \|\vec{y}_{\pi(i)}\|_p \right| + \sum_{i=|X|+1}^{|Y|} \|\vec{y}_{\pi(i)}\|_p \overset{\text{Lemma 2}}{\leq}$$
$$\sum_{i=1}^{|X|} \|\vec{x}_i - \vec{y}_{\pi(i)}\|_p + \sum_{i=|X|+1}^{|Y|} \|\vec{y}_{\pi(i)}\|_p = D_{\mathrm{mm}}^{L_p, W_{\vec{0}}}(X,Y)$$

□

A.3 Theorem 5

Proof: According to Theorem 2, $D_{\mathrm{pcp}}^{L_p,s}(\hat{X},\hat{Y}) \leq D_{\mathrm{pmm}}^{L_p,s}(\hat{X},\hat{Y})$ holds.
To obtain $D_{\mathrm{pmm}}^{L_p,s}(\hat{X},\hat{Y}) \leq D_{\mathrm{pmm}}^{L_p,s}(X,Y)$ we have to show that

$$\min_{\pi_1 \in \Pi(X), \pi_2 \in \Pi(Y)} \left(\sum_{i=1}^{s} \left| \|\vec{x}_{\pi_1(i)}\|_p - \|\vec{y}_{\pi_2(i)}\|_p \right| \right) \leq$$
$$\min_{\pi_1 \in \Pi(X), \pi_2 \in \Pi(Y)} \left(\sum_{i=1}^{s} \|\vec{x}_{\pi_1(i)} - \vec{y}_{\pi_2(i)}\|_p \right)$$

and this follows from Lemma 2. □