PROFI

Project number:	FP6-511572
Project acronym:	PROFI
Title:	Perceptually-relevant Retrieval Of Figurative Images

Deliverable No: D6.3: Report on partial matching with the vantage indexing methods

In general, a distance measure does not satisfy the triangle inequality when it allows partial matching. As a consequence, avoidance of false negatives is by the vantage indexing scheme not guaranteed anymore. This conflicts with the zero-tolerance requirement of the central application of this project. This deliverable describes several different modifications and additions to the vantage indexing scheme to overcome this problem. The suitability for the project is investigated for each method.

Due month:	M33
Delivery month:	M33
Lead partner:	Utrecht University
Partners contributed:	Utrecht University
Classification:	PU

Project funded by the European Community under the "Information Society Technologies" Programme

1 Introduction

PROFI Work Package 6 embodies Shape Indexing. Its three objectives are in order of delivery date: the development of simulation tool software (6.1), the development of strategies for selecting vantage objects (6.2) and investigating how partial matching can be best supported by the vantage indexing method (6.3). This document reports the progress made on 6.3: vantage indexing and partial matching.

One of the indexing techniques which are investigated within the PROFI project is called Vantage Indexing. This indexing technique works as follows: given a multimedia database A and a distance measure $d : A \times A \rightarrow \mathbb{R}$, select from the database a set of k objects $A^* = \{A_1^*, ...A_k^*\}$, the so called vantage objects. Compute the distance from each database object A_i to each vantage object, thus creating a point $p_i = (x_1, ...x_k)$, such that $x_j = d(A_i, A_j^*)$. Each database object corresponds to a point in the m-dimensional vantage space.

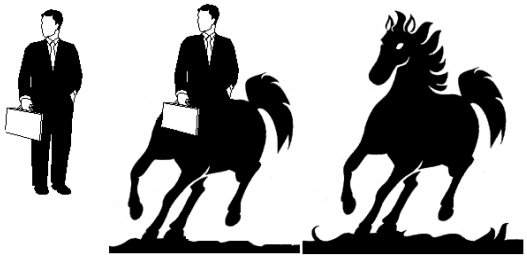
A query on the database now translates to a range-search or a nearest-neighbor search in this k-dimensional vantage space: compute the distance from the query object q to each vantage object (i.e. position q in the vantage space) and retrieve all objects within a certain range around q (in the case of a range query), or retrieve the nearest neighbors to q (in case of a nearest neighbor query). The distance measure δ used on the points in vantage space is L_∞.

As long as the triangle inequality holds for the distance measure d defined on the database objects, recall (ratio of number of relevant retrieved objects to the total number of relevant objects in the whole data base) is 100%, meaning that there are no false negatives. However, when partial matching is supported by the underlying distance function d, this constraint is violated, and the avoidance of false negatives is not guaranteed anymore. This report investigates possible solutions to this problem, which is stated in more detail in the following section.

1.1 Partial matching and the triangle inequality

If the distance measure d allows partial matching, i.e. a part of the query matches to (a part of) a database object or vice versa, the triangle inequality constraint may be violated. See Figure 1 for an example of this situation. Figure 2 illustrates (with more abstract shapes) how this triplet of objects fails during the querying procedure in the vantage space.

Figure 1: Due to partial matching, a violation of the triangle inequality may occur



This also means that taking the intersection of all regions of interest introduces false negatives as well, since one partially matching vantage object alone may pull the relevant database object out of the region of interest. This is illustrated in Figure 3.

Figure 2: Database object a is a false negative for query q because it doesn't resemble vantage object v (so it's not in the region of interest), but matches to a part of q.

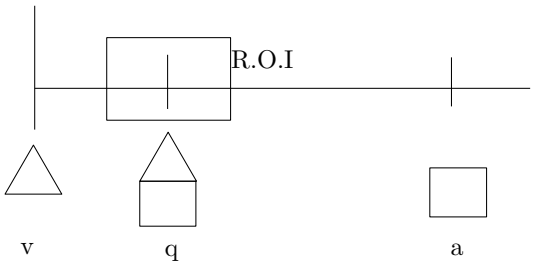
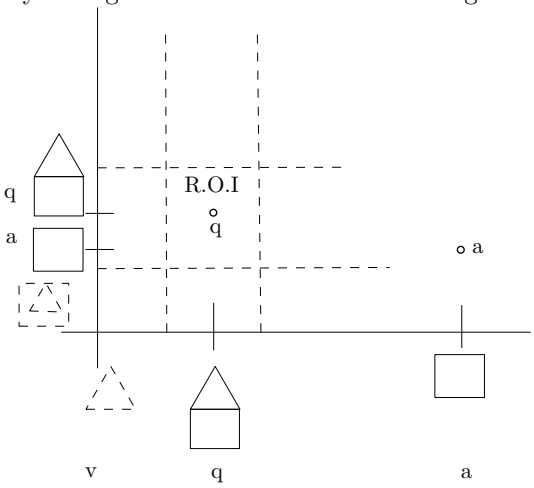


Figure 3: Database object a is a false negative for query q because it isn't in the region of interest around q in this vantage space, given the two vantage objects (shapes represented by dashed lines). By taking the intersection of the two regions of interest, a is excluded from the returned result.



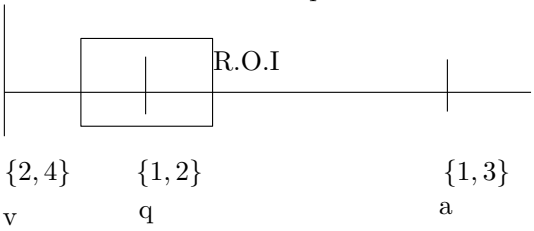
2 Related work

In [8], Jacobs et al. state that in a non-metric space, the distance between two objects is not a good estimator for redundancy, should these two objects be used as class representatives or vantage objects. Moreover, they propose to use a redundancy measure that is similar to the correlation criterion that we use in [13]. The main point they make is that exemplar-based methods can still be applied when working with non-metric similarity measures, as long as the methods to find the representatives are adjusted to deal with the non-metric case. For this purpose, they use the correlation-like criterion. However, as we have seen and argued in the previous Section, it is not so obvious that exemplar-based methods can be used in conjunction with non-metric distance functions, in particular when the triangle inequality is violated as a result of allowing partial matching.

3 Proposed solutions to the problem

In [13] we propose a strategy for selecting good vantage objects. The possible solutions to this new problem explored below should be taken in this light and as an extension to the

Figure 4: Vantage object v is not an appropriate vantage object for retrieving a given query q, because the match between a and q is on a different feature.



referred work.

3.1 Weak triangle inequality

One may argue that in practice, under certain conditions, the false negative as depicted in Figure 2 for instance, will not lie very far away from q. In that case, by simply extending the search radius object a can be included in the region of interest or search result. The question that arises immediately is how large the search radius then should be to be sure (or sure with a certain known uncertainty) that a is in the returned set. Furthermore, extending the search radius will probably lead to an increase in false positives.

This solution is in principle a form of softening the triangle inequality constraint to a *weak* triangle inequality. In fact, this instance of the triangle inequality amounts to

$$d(a, q) \leq d(a, v) + d(q, v) + \epsilon$$

where an additive constant ϵ is necessary to compensate for violation of the triangle inequality. If such a value ϵ can be inferred from the distance measure (either analytically, or experimentally by inspecting a sufficiently large sample of distance triplets), extending the search range accordingly may form a solution to the problem.

The same procedure can be carried out for multiplicative factors, or other forms of the weak triangle inequality.

3.1.1 Suitability for PROFI

Although this report deals mainly with triangle inequality violation as a result of partial matching, a probabilistic algorithm that approximates exact matching can cause violation as well. Complete-complete matching by the probabilistic algorithm that was developed within work package 4 are a good example of this scenario. Extending the search radius (obeying a weaker version of the triangle inequality) forms a good solution to this problem.

3.2 Use of image primitives

Another solution is to reduce each image (database, vantage and query) to image primitives and apply vantage indexing on this level. An image primitive is defined as a part of the image that should completely match to a complete other image primitive. In other words, the idea is to segment the image such that complete-complete matching can be performed on the parts, thus avoiding false negatives. One query image would therefore lead to several queries performed with its parts, so the result consists of multiple sets of candidate matches. These

sets have to be combined in some way to present one overall set of candidate matches for the original query image. Rank aggregation techniques (combining several ranked lists into one general ranking) can be used for this purpose [5]. More details on rank aggregation are given in Section 3.5.

3.2.1 Suitability for PROFI

As a result of work package 5, one matching method takes already place at image primitive level, i.e. the transformation based matching method is applied to image primitives, and all primitive similarities are combined into a final matching result. This solution therefore follows naturally. It however requires a redefinition of the matching; the final matching will no longer be done *prior* to indexing, but will be a direct *result* of the indexing / rank aggregation. Matching the primitives will provide the building blocks for the final matching step, the way these are combined should reflect the way image primitive matching is combined into one matching result now.

3.3 Specific triplets (query, db object, vantage object)

It is sufficient if the triangle inequality holds for the specific triplet of objects (query, database object, vantage object). This could be taken into account as an extra constraint while selecting the vantage objects. It requires prior knowledge of the queries that will be used.

3.3.1 Suitability for PROFI

In the context of this project it is undesirable to allow only a fixed set of queries. This solution is therefore inappropriate.

3.4 Inferring extra information from the distance measure

In some cases it is possible to exploit additional information that is provided by the matching algorithm. In section 3.1 we already gave an example of how to use the severeness of violation of the triangle inequality to adjust the search radius. This solution however requires a preprocessing step in which distance triplets are sampled to estimate the violation.

Another example, which does not require an estimating preprocessing step, is to infer from the distance evaluation the fraction of the object that has been matched to the vantage object, in case of a partial match. More specifically, to infer *which* part of the query has been matched to the vantage object. In this section we will give details on how this can possibly solve the problem.

In the scenario depicted in Figure 3, the triangular vantage object is not a good vantage object to use if we want to retrieve object a given query q. However, if a would have been a triangular object instead of a rectangular object, it would have been positioned within the Region Of Interest (ROI) around q and it would have been retrieved. To put this problem in a somewhat broader perspective, let an image (or database object in general) be composed of several primitive shapes (or building blocks, parts) $1, 2, 3..$ etc. See Figure 4 for an illustration. A vantage object is only a good vantage object for retrieving a certain database object, if the match between the query and the database object is based on image feature that is present within the vantage object. So in the example depicted in Figure 4, v is not a good vantage object to retrieve a given query q, since the match between a and q is on totally different

grounds; The match between a and q is based on component 1, whereas the match between q and v is based on component 2, and there is no match at all between a and v.

Now let's assume that all database objects that contain component 2, or a highly similar component, are in the ROI around q with respect to vantage object v (see Figure 4). We now may still have false negatives residing in the database that match to the query because they contain component 1, or a similar component. Therefore, we should also match the query against a vantage object that contains (or only consists of) component 1. The results of these two querying procedures can not be intersected, this time the union has to be reported.

To generalize, the following procedure is proposed here. A large collection of vantage objects is maintained, ideally one for each component that is present in the database objects, but it can be less. Let's assume that there are k vantage objects. The query is matched against all these vantage objects, and the vantage objects are sorted in decreasing order based on their similarity to the query. Furthermore, for each match, it is known which part of the query was used to match with the vantage object. This information is used to select only the first k' vantage objects, such that the complete query has been matched against the k' vantage objects, albeit that the different parts of the query match with different vantage objects. For each vantage object with the selected k', all objects lying within the search range around q are added to the final querying result.

In the approach proposed in this section, the k-dimensional range search or nearest neighbor search has been replaced by k individual searches, each based on just one vantage object, each producing one ranked list. These k ranked lists need to be combined, or aggregated into one list on which most lists can agree. This process, called rank aggregation, is an interesting problem in itself, for which several strategies have been proposed [5]. In this following section, rank aggregation as a solution to the problem will be described in more detail.

3.4.1 Suitability for PROFI

As the probabilistic transformation-based matching method developed in work package 4 supports partial matching and can output the required information (which part matched), this approach is well suited to solve the problem within the PROFI project.

3.5 Rank aggregation

It takes only one non-triangular triplet of query, vantage object and database object to produce a false negative: when the database object is within the search range according to all but one vantage objects, this object is not in the returned result. This is a consequence of using L_∞ in vantage space: all returned items need to be positioned within a hypercube around the query. In practice, probably not all triplets of query, database object and vantage object will violate the triangle inequality. It is therefore interesting to investigate alternatives to range or nearest neighbor querying using these hypercubes. One of these alternatives is rank aggregation.

Rank aggregation is the study of combining several ranked lists into one ranking. The goal is to unify all rankings such that the combined ranking reflects the order of the individual "voters" as adequately as possible. To quantify this property of rank resemblance, a distance measure between two ranked lists is necessary. The most well-known distance measures for ranked lists are the *Kendall-tau distance* and the *Spearman footrule distance*.

3.5.1 Ranked list distances and aggregation properties

The Spearman footrule distance measures the distance between two ranked lists by summing the differences in rankings of each item. That is, given two complete rankings r_1 and r_2 of a universe U of items , $F(r_1, r_2) = \sum_{i \in U} |r_1(i) - r_2(i)|$ [3].

The Kendall-tau distance is defined as the number of pairwise disagreements in the relative rankings of items in the two lists. That is, $K(r_1, r_2) = |i, j \text{ s.t. } r_1(i) < r_1(j) \text{ and } r_2(i) > r_2(j)|$ [9].

When aggregating a collection of partial lists $\tau_1, \tau_2, \ldots, \tau_k$ into an aggregation σ, it is said to be *Kemeny-optimal* if $SK(\tau_1, \tau_2, \ldots, \tau_k) = \sum_{i=1}^{k}(K(r_1, r_2))$ is minimized.

An attractive property of a Kemeny-optimal aggregation is that it satisfies the textitConcordet-criterion, that states that if a certain item defeats every other item in simple majority voting, it should be ranked first [2]. In fact, Kemeny-optimal aggregations are known to satisfy Truchon's extended Condorcet criterion, that states that if there is a partition (T, U) of the items $\{1, \ldots, n\}$ such that for any $x \in T$ and $y \in U$ the majority prefers x to y, then x must be ranked above y [12].

3.5.2 Aggregation methods

Markov chains. Unfortunately, finding a Kemeny-optimal aggregation is NP-hard, as was shown even for the special case of 4 complete lists by Dwork et al. [5, 4]. Therefore, Dwork et al. propose *local* Kemeny-optimality, which is basically a relaxed version of Kemeny-optimality. There methods, based on Markov Chains, are guaranteed to satisfy both the extended Condorcet criterion and local Kemeny-optimality, whilst running in polynomial time.

Borda's method. Formally, this method first proposed in 1770 is defined on a set of complete rankings R as follows. For each item i and list $r_k \in R$, let $B_{r_k(i)}$ equal the number of items j in r_k such that $r_k(j) > r_k(i)$. The total Borda score for the item i is given by $B_t(i) = \sum_{r \in R} B_r(i)$. Ranks are assigned by sorting scores B_t from highest to lowest, with the highest score getting the lowest rank.

Median ranking. This method assigns ranks based on the median of all ranks an i has in a given set of complete rankings R. This method can produce footrule optimal aggregations, which are within a constant bound of Kemeny optimal. The rankins it produces satisfy the extended Condorcet criterion, and it may be computed efficiently, especially in the case where only the top k aggregate rankings are required. [6].

Extensions with item similarity. A limitation of these distance measures (Kendall-tau, Spearman's footrule) and above mentioned methods that are based on them, is that, while comparing two ranked lists r_1 and r_2, they only take into account items that are ranked in both ranked lists. In practical applications and real-world situations however, such as web searching, meta searching or advertisement placing, ranked lists are often noisy, incomplete or even disjoint. Therefore, Sculley proposed to use similarity between *items* in combination with rank aggregation methods [10]. The intuition behind this approach is that similar items should be ranked similarly. He extended the Kendall-tau and Spearman's footrule distances with item similarity, as well as the Markov Chain based methods by Dwork et al. and two other rank aggregation methods.

3.5.3 Rank aggregation and similarity search

An interesting application of rank aggregation is found by Fagin et al., who apply it to nearest neighbor search in vector spaces [6]. Whereas this problem is usually solved by multidimensional access methods (MAMs), they propose to produce k ranked lists for a given query q, when the objects are represented in a k-dimensional vector space. Every dimension of the vector space is seen as a voter: the j-th ranked list contains the database objects in decreasing order of similarity with respect to the query, according to the j-th coordinate. They use median rank aggregation: sort based on the median of the ranks an item has in all the ranked lists. The k items with the highest aggregate values are returned as result of the k-nearest neighbor search. They show that their method is a reduction of the ϵ-approximate Euclidean nearest neighbor problem to the problem of finding the candidate with the best median rank in an election where there are n candidates and $O(\epsilon^{-2\log n})$ voters. Furthermore, they show their algorithm is *instance optimal* [7], which means that it is accessing the ranked lists only sequentially, but uses only a constant factor more access operations than *any* algorithm that uses both sequential and random access operation. This makes the approach efficient and database friendly.

3.5.4 Extensions

Besides normal rank aggregation, where each ranked list is equally important, a priority can be given to the different vantage objects or ranked lists. This can be done using simple heuristics such as the match size and/or quality (Figure 5: a receives a higher weight than b).

Alternatively, a relevance feedback loop may be incorporated to prioritize the rankings. More specifically, the user is asked to indicate a number of relevant items in the result. The system then evaluates with which vantage objects these relevant items matched closely, apparently these are important vantage objects to the user. This information can be used to assign more influence to these vantage objects in the rank aggregation.

3.5.5 Suitability for PROFI

As was pointed out in Sections 3.2 and 3.4, rank aggregation can be a tool to combine search results of partial queries. Furthermore, it can provide a solution to the problem in its own respect. The influence of a non-triangular triplet can be reduced using rank aggregation methods described. Moreover it can support vantage object priority through relevance feedback more naturally than geometrical range search. It is therefore very suited to solve the problem within the context of the PROFI project.

3.6 Triangle-generating modifiers

An interesting approach to bridge the gap between non-metric distance measures and pivot-based indexing algorithms is proposed by Skopal [11]. By sampling many triplets of objects, the amount of triangle inequality violation of the distance measure is estimated. A monotonically increasing function is trained (a *Triangle Generating Modifier*) that transforms distances such that the triangle inequality constraint is satisfied again. See Figure 6 for an example; the distances in Figure 6(a) violate the triangle inequality, but after applying the Triangle Generating Modifier $f(x) = \sqrt{x}$, the triangle inequality is satisfied (Figure 6(b). Due to their required monotonicity, Triangle Generating Modifiers are order-preserving. Therefore, the

Figure 5: (a) The query q matches well against the rectangular vantage object and the database object a is in the R.O.I, (b) q matches well to the triangular vantage object as well and now database object b is in the R.O.I., (c) q does not match against the circular object, and no object lies inside the R.O.I. now. (d) The rectangular and triangular vantage objects are the two best matching vantage objects, and besides that, they *cover* the complete query. Therefore, these two vantage objects are used for this query; note that not the intersection of the slabs, but the union is taken. Otherwise, the result set would have been empty.

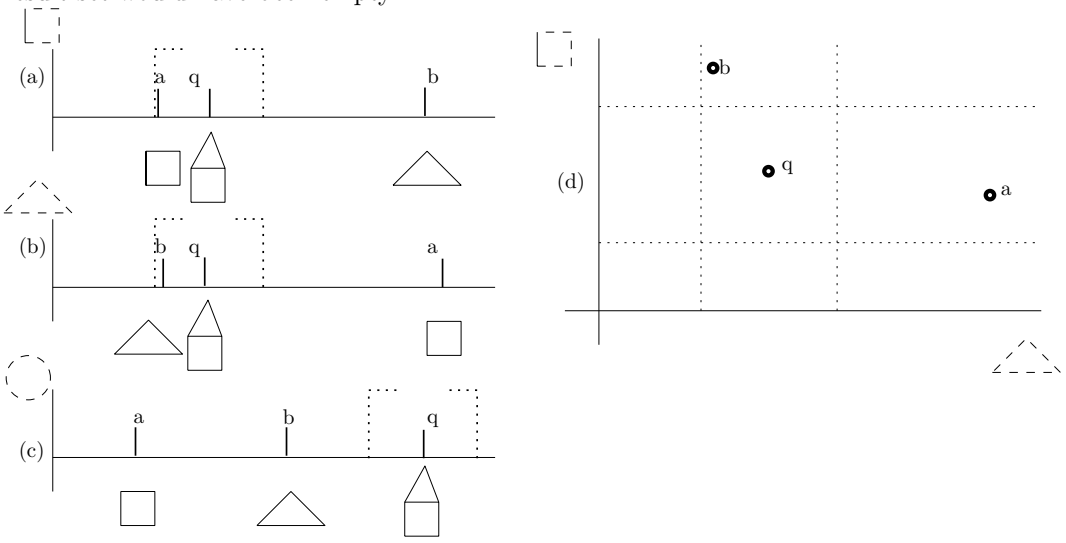
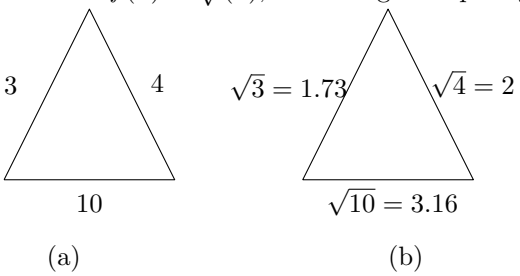


Figure 6: (a) Distances violate the triangle inequality. (b) After applying the Triangle Generating Modifier $f(x) = \sqrt{(}x)$, the triangle inequality is satisfied.

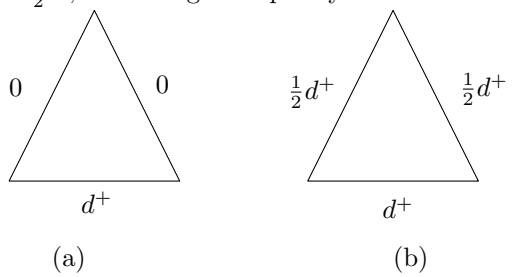


ranking for a given query will not change, only the distances of the database objects to the query will change.

As a result of transforming the distance measure into a metric, the intrinsic dimensionality of the obtained metric space may become high. This increase in dimensionality should be kept as low as possible, since higher dimensionality makes the pivot-based search more difficult. It can be seen easily that, given a bounded non-metric distance function with maximal distance d^+, the Triangle Generating Modifier $f(x) = \frac{x+d^+}{2}$ always solves the problem. See Figure 7 for an illustration. However, this Triangle Generating Modifier increases the intrinsic dimensionality of the obtained metric space significantly, and pivot-based searching algorithms used for indexing will tend to sequential search.

Specifically, only those triplets that do violate the triangle inequality should be dealt with by the modifier. Skopal proposes an algorithm that parametrizes a base function such as $\mathrm{FP}(x, w) = x^{\frac{1}{1+w}}$ such that the triangle inequality is just satisfied for a given triplet.

Figure 7: Given a bounded non-metric distance function with maximal distance d^+, (a) distances violate the triangle inequality maximally. (b) After applying the Triangle Generating Modifier $f(x) = \frac{x+d^+}{2}$, the triangle inequality is satisfied.



(a) (b)

Composing all these functions will provide the final Triangle Generating Modifier. A measure for intrinsic dimensionality for general metric spaces was proposed by Chavez et al. [1]. During the construction/training of the Triangle Generating Modifier, this measure is evaluated for a sample of the database. It is kept as low as possible.

3.6.1 Suitability for PROFI

Although appealing because of its formal mathematical framework, the suitability of this approach for the project is limited. Triangle inequality violation is not introduced directly by a fundamentally non-metric distance measure, but by allowing partial matching. This makes it dubious to estimate a triangle violation error: it is rather a property of the dataset and query than of the distance measure.

4 Concluding remarks

A distance measure does not satisfy the triangle inequality when it allows partial matching. As a consequence, avoidance of false negatives is by the vantage indexing scheme not guaranteed anymore. This deliverable described several solutions to deal with this problem at the indexing side. A powerful concept that can exploit several properties of the developed matching algorithms is rank aggregation. The transformation based matching approach can provide information on *what* part of the query matched, and it has been applied in a more general matching scheme where it is used to match image primitives. Both approaches allow querying in a vantage space with parts of the query, yielding several ranked lists for one query. Through the process of rank aggregation these lists can be combined into one.

Moreover, rank aggregation in itself is a viable solution to the problem as well. While performing a geometric range query, one non-triangular triplet can have fatal consequences for a relevant item. When the geometric range query is replaced by aggregating ranked lists that are produced by the different vantage objects, this influence can be reduced significantly and the relevant item may still be ranked high.

Transforming the non-metric distance function into a metric by processing the distances such that all sampled triplets are triangular is an appealing approach because of its formality and mathematical foundation. It is however less appropriate within the context of this project than the other methods, because the violation of the triangle inequality is not a result of a fundamentally non-metric distance measure, but caused by partial matching. Training

a general function on dataset properties can therefore be considered overfitting, instead of solving the problem generically and robust with respect to future database objects and queries.

References

[1] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, 2001.

[2] M.-J. Condorcet. Essai sur l'application de l'analyse a la probabilite des decisions rendeus a la pluralite des voix. 1785.

[3] P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Roayl Statistical Society*, 32(2):262–268, 1977.

[4] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited.

[5] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, 2001.

[6] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312, New York, NY, USA, 2003. ACM.

[7] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 102–113, New York, NY, USA, 2001. ACM.

[8] David W. Jacobs, Daphna Weinshall, and Yoram Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.

[9] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(12):81–93, 1938.

[10] D. Sculley. Rank aggregation for similar items. In *Siam Conference on Data Mining 2007*, 2007.

[11] Tomás Skopal. On fast non-metric similarity search by metric access methods. In *EDBT*, pages 718–736, 2006.

[12] M. Truchon. An extension of the condorcet criterion and kemeny orders. *cahier 98-15 du Centre de Recherche en Economie et Finance Appliquees*, 1998.

[13] Reinier H. van Leuken, Remco C. Veltkamp, and Rainer Typke. Selecting vantage objects for similarity indexing. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 453–456, Washington, DC, USA, 2006. IEEE Computer Society.