Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

A comparison of pivot selection techniques for permutation-based indexing[☆]

Giuseppe Amato, Andrea Esuli, Fabrizio Falchi[*]

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", via G. Moruzzi 1, Pisa 56124, Italy

ARTICLE INFO

Available online 3 February 2015

Keywords:
Permutation-based
Pivot
Metric space
Similarity search
Inverted files
Content based image retrieval

ABSTRACT

Recently, permutation based indexes have attracted interest in the area of similarity search. The basic idea of permutation based indexes is that data objects are represented as appropriately generated permutations of a set of pivots (or reference objects). Similarity queries are executed by searching for data objects whose permutation representation is similar to that of the query, following the assumption that similar objects are represented by similar permutations of the pivots. In the context of permutation-based indexing, most authors propose to select pivots randomly from the data set, given that traditional pivot selection techniques do not reveal better performance. However, to the best of our knowledge, no rigorous comparison has been performed yet. In this paper we compare five pivot selection techniques on three permutation-based similarity access methods. Among those, we propose a novel technique specifically designed for permutations. Two significant observations emerge from our tests. First, random selection is always outperformed by at least one of the tested techniques. Second, there is no technique that is universally the best for all permutation-based access methods; rather different techniques are optimal for different methods. This indicates that the pivot selection technique should be considered as an integrating and relevant part of any permutation-based access method.

1. Introduction

Given a set of objects C from a domain $\mathcal{D}$, a *distance function* $d: \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$, and a query object $q \in \mathcal{D}$, a similarity search problem can be generally defined as the problem of finding a subset $S \subset C$ of the objects that are closer to q with respect to d. Specific formulations of the problem can, for example, require to find the k closest objects (k-nearest neighbors search, k-NN), i.e., $|S| = k$ and $\forall x \in S, y \in \ (C \setminus S)$ $(d(x,q) \le d(y,q))$, or all the objects that are closer than a given threshold distance t. i.e., $S = \{x | x \in \ C \wedge d(x,q) \le t\}$. The k-NN formulation is the most common one.

Similarity search is a difficult problem and various indexing schema have been defined to process similarity queries efficiently. Good surveys of the various approaches proposed in the literature can be found in [39,34]. However, in most applications, as for instance multimedia retrieval, an exact solution to the similarity search problem is not strictly required. In these cases, performing an approximate similarity search [40,30] is sufficient. Accepting even a small degree of approximation in results allows to obtain them much more efficiently.

Permutation-based indexes have been proposed as a new approach to efficient and effective approximate similarity search [2,12,16,28]. In permutation-based indexes, data objects and queries are represented as appropriate permutations of a set of n pivots $P = \{p_1 \ldots p_n\} \subset \mathcal{D}$. Formally, every object $o \in \mathcal{D}$ is associated with a permutation Π_o that lists the identifiers of the pivots by their closeness to o, i.e., $\forall j \in \{1, 2, \ldots, n-1\}, d(o, p_{\Pi_o(j)}) \le d(o, p_{\Pi_o(j+1)})$, where $p_{\Pi_o(j)}$ indicates the

[☆] This is a revised and extended version of a paper appeared as [1].

[*] Corresponding author. Tel.: +39 050 315 2911.
E-mail addresses: giuseppe.amato@isti.cnr.it (G. Amato), andrea.esuli@isti.cnr.it (A. Esuli), fabrizio.falchi@isti.cnr.it (F. Falchi).

http://dx.doi.org/10.1016/j.is.2015.01.010
0306-4379/© 2015 Elsevier Ltd. All rights reserved.

pivot at position j in the permutation associated with object o. For convenience, we denote the position of a pivot p_i, in the permutation of an object $o \in \mathcal{D}$, as $\Pi_o^{-1}(i)$ so that $\Pi_o(\Pi_o^{-1}(i)) = i$.

The similarity between objects is approximated by comparing their representation in terms of permutations. The basic intuition is that if the permutations relative to two objects are similar, i.e. the two objects *see* the pivots in a similar order of distance, then the two objects are likely to be similar also with respect to the original distance function d.

Once the set of pivots P is defined it must be kept fixed for all the indexed objects and queries, because the permutations deriving from different sets of pivots are not comparable. A selection of a "good" set of pivots is thus an important step in the indexing process, where the "goodness" of the set is measured by the effectiveness and efficacy of the resulting index structure at search time.

Permutation based methods share some ideas with the Shared Nearest Neighbors methods (SNN) [22,32,14]. These methods introduce the concept of secondary similarity measures, which evaluates the similarity among two objects by considering the amount of overlap of their neighborhoods. The neighborhood of an object is determined using the original distance and all objects of the dataset. Secondary similarity measures have been shown to be able to reduce the impact of the curse of dimensionality in cases in which the discriminative power of the primary similarity measure is reduced by the high dimensionality of the similarity space. The difference between the permutation-based methods and the SNN methods is that permutation-based methods encode original objects with neighbor objects taken from a very small subset of the entire dataset, rather than the entire dataset. Moreover, the primary purpose of this encoding is to build efficient and scalable approximate similarity search index structures, rather than computing a better distance than the original distance, as SNN method do. In addition, no pivot selection technique is needed by SNN methods, given that they use the entire dataset to determine the neighborhood of objects.

In the field of permutation-based access methods the most commonly adopted technique for the definition of P is to randomly select the n objects from C [2,12,16]. Even though there is a relatively rich literature on pivot selection techniques for the general class of *pivot-based* access methods [39] (see Sections 2 and 3), to the best of our knowledge, no rigorous comparison of the effectiveness of the various selection techniques, when used in combination with permutation-based access methods, has been performed yet. In this paper we compare five techniques for the definition of sets of pivots to be used by permutation-based access methods. One of the techniques that we compare is a novel proposal that we have designed to be used with permutation based index.

In summary, the contribution of this paper is twofold.

1. We test various pivot selection techniques, including random selection, on a number of permutation based indexes. An interesting result is that different selection methods were optimal for different index schema. In fact, the way in which permutations are used, by different indexing schema, is basically different, and this is reflected in the pivot selection techniques.
2. We propose and compare a new pivot selection criterium, expressly designed to be used with permutation-based indexes. This method is clearly superior when used with the MI-File index [2].

The paper is structured as follows. In Section 2 we discuss related work. Section 3 presents the techniques being compared. The tested similarity search access methods are presented in Section 4. Section 5 describes the experiments and comments their results. Conclusion and future work are given in Section 6.

2. Related work

The study of pivot selection techniques for access methods usually classified as pivot-based [39] has been an active research topic, in the field of similarity search in metric spaces, since the nineties. Most access methods make use of pivots to reduce the number of data objects accessed during similarity query execution. The choice of the pivots plays a relevant role in allowing the access methods to achieve their best performance. In an early work by Shapiro [37], it was noticed that good performance was obtained by locating pivots *far away* from data clusters. In [8,26,38], following this intuition, several heuristics were proposed to select pivots between the *outliers* and far away from each other.

In [18] it is shown that it is possible to find an optimal set of pivots selecting them as the vertices of a large regular simplex containing all the objects of the database but this result does not apply to general metric spaces.

Pivot selection techniques that maximize a lower bound estimate of the original distance d by means of a *pivoted distance* were exploited in [11] (see Section 3.3). For these techniques is was also observed that while good pivots are usually outliers, not all outliers can be good pivots [9]. In [10,31], the problem of dynamic pivot selection as the database grows is faced. In [25] Principal Component Analysis (PCA) has been proposed for pivot selection. Principal components (PC) of the dataset are identified by applying PCA on it (actually a subset to make the method computationally feasible) and the objects in the dataset that are best aligned with PC vectors are selected as pivots. In [38] it was proposed to select the corners of the data set as pivots for Vantage Point Tree (VPT). Multi-Vantage Point Tree (MVPT) [7] selects multiple corners.

Works that use permutation-based indexing techniques have mostly performed a random selection of pivots [2,16,12] following the observation that the role of pivots in permutation-based indexes appears to be substantially different from the one they have in traditional pivot-based access methods. In fact, the use of previous selection techniques, with permutation based indexes, did not reveal significant advantages. At the best of our knowledge, the only report on the definition of a specific selection techniques for permutation-based indexing is in [12], where it was mentioned that no significant improvement, with respect to random selection, was obtained by maximizing or minimizing the Spearman Rho distance through a greedy algorithm.