Non-Hierarchical Clustering with Rival Penalized Competitive Learning for Information Retrieval

Irwin King and Tak-Kan Lau

Department of Computer Science & Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
king@cse.cuhk.edu.hk, http://www.cse.cuhk.edu.hk/~king

Abstract. In large content-based image database applications, efficient information retrieval depends heavily on good indexing structures of the extracted features. While indexing techniques for text retrieval are well understood, efficient and robust indexing methodology for image retrieval is still in its infancy. In this paper, we present a non-hierarchical clustering scheme for index generation using the Rival Penalized Competitive Learning (RPCL) algorithm. RPCL is a stochastic heuristic clustering method which provides good cluster center approximation and is computationally efficient. Using synthetic data as well as real data, we demonstrate the recall and precision performance measurement of nearest-neighbor feature retrieval based on the indexing structure generated by RPCL.

1 Introduction

One of the key issues in information retrieval of data in large and voluminous database is the design and implementation of an efficient and effective indexing structure for the data objects in the database. Without a properly designed indexing structure, the retrieval of information may be reduced to a linear exhaustive search. On the other hand, a good indexing structure will make the retrieval accurate and computationally efficient.

The following paragraphs outline the basic feature vector model for nearest-neighbor search. In our framework, we let $DB = \{I_i\}_{i=1}^N$ be a set of image objects. Without loss of generality, a feature extraction function $f: I \times \theta \rightarrow \mathbb{R}^d$ extracts from an image $I$, with a set of parameters $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$, a real-valued $d$-dimensional vector. Hence, we may view the extracted feature vector as a point in a $d$-dimensional vector space. Furthermore, we may use a random variable $X$ to denote the feature vector extracted from the image set $DB$ and $x_i, i = 1, 2, \ldots, N$ to denote the instance of the feature vector extracted from $DB$.

Once the feature vectors have been obtained, similar feature (content-based) search can be performed as a nearest-neighbor search by using a distance function. A typical distance function $D$ is defined as $D: F \times F \rightarrow \mathbb{R}$ satisfying: (1) $D(x, y) \geq 0$, (2) $D(x, y) = D(y, x)$, (3) $D(x, y) = 0$ iff $x = y$, and (4) $D(x, y) + D(y, z) \geq D(x, z)$ where $x, y$, and $z \in F$ and $F$ is a feature vector set.

Here, $L_2$-norm (Euclidean distance) is one of the common distance functions and it is defined as:

$$D(x, y) = \|x - y\| = \left(\sum_{i=1}^d (x_i - y_i)^2\right)^{1/2}.$$

There are two typical types of query involved in image databases: (1) Range Search and (2) $k$ Nearest-Neighbor Search. Given a set of $N$ features $X = \{x_i\}_{i=1}^N$, a Range Query, $\hat{x}$, returns the set, $P$, of features as $P = \{x \mid x \in X$ and $0 \leq D(x, \hat{x}) \leq \epsilon\}$, where $\epsilon$ is a pre-defined positive real number and $D$ is a distance function. As in the $k$ Nearest-Neighbor Search case, given a set of $N$ features $X = \{x_i\}_{i=1}^N$, a $k$ Nearest-Neighbor Query, $\hat{x}$, returns the set $P \subseteq X$ satisfying: (1) $|P| = k$ for $1 \leq k \leq N$ and (2) $D(\hat{x}, x) \leq D(x, y)$ for $y \in X - P$ where $D$ is a distance function. In this paper, we will focus on the latter type of query.

* This work is supported in part by the RGC Grant #CUHK4176/97E. Portions of this manuscript have been presented in [8].

1 In this paper, data objects and feature vectors are interchangeable unless stated otherwise.
Once the features have been extracted and the query model has been defined. The crucial link between the features and user query is the indexing structure (organization). A well-organized indexing structure of the underlying feature vectors support an efficient and effective retrieval of user queries.

Recently, researchers have developed many new indexing methods for content-based retrieval in multimedia databases. For example, rectangle-based indexing as in R-Tree [6], R+-Tree [11], R*-Tree [1], SR-Tree [7]. Partition-based Indexing as in Quad-tree [5], k-d Tree [2], VP-Tree [4, 13], and MVP-tree [3].

However, one major problem of these indexing techniques has been that these methods fail to utilize the underlying data distribution to their advantage in their indexing structure. This results in what is known as the boundary query problem where the retrieval Precision will degrade when a query is near the boundary of a partition in the indexing structure due to the systematic yet unfavorable partitioning of some indexing techniques. To overcome this, we will present a non-hierarchical clustering algorithm based on Rival Penalized Competitive Learning (RPCL) heuristic and demonstrate its effectiveness in generating indexing structure for large image databases.

In Section 2 we will present more details on RPCL and the associated non-hierarchical indexing. Experimental results of the proposed method are presented in Section 3. We will discuss some issues associated with the proposed method and conclude in Section 4.

2 Non-Hierarchical Clustering with RPCL

There are two main goals in our proposed solution: (1) find a quick way to partition the input feature set into partitions and (2) impose an indexing structure over these partitions so that the nearest-neighbor information retrieval can be made effectively.

Rival Penalized Competitive Learning (RPCL) clustering [12] can be regarded as an unsupervised extension of Kohonen's supervised learning vector quantization algorithm LVQ2 [9]. It can also be regarded as a variation to the more typical Competitive Learning (CL) algorithms [10]. RPCL is a stochastic clustering algorithm that is able to perform adaptive clustering efficiently and quickly leading to an approximation of clusters that are statistically adequate.

The proposed solution is to use RPCL to find hierarchal clusters such that the indexing structure can be created based on a natural partition of the feature vector distribution. Although this may not result in a balanced tree structure, this indexing structure will be able to answer nearest-neighbor queries more effectively.

The main advantages of RPCL are: (1) the heuristic is computationally efficient, (3) it is no worse than other methods when high dimensional features, and (3) RPCL can be implemented in a distributed environment achieving even greater speed-up in generating indexing structure of feature vectors.

2.1 The RPCL Algorithm

Assuming there are \( k \) cluster centers, the basic idea behind RPCL is that in each iteration, the cluster center for the winner's unit is accentuated where as the weight for the second winner, or the rival, is attenuated. The remaining \( k - 2 \) centers are unaffected. The winner is defined as the cluster center that is closest to the randomly selected feature vector. Instead of \( k \) can be of any value, in our application we use the special version of the RPCL clustering algorithm when \( k = 2^i, i = 1, 2, 3, \ldots \) so that a systematic index tree can be formed.

Let \( k, c_w \) and \( c_r \) to denote the number of clusters, cluster center points for the winner and rival clusters respectively. The algorithm is illustrated in the following steps.