Signal Processing: Image Communication 63 (2018) 1–8

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Large-scale vocabularies with local graph diffusion and mode seeking

Shanmin Pang [a,*], Jianru Xue [b], Zhanning Gao [b], Lihong Zheng [c], Li Zhu [a]

[a] School of Software Engineering, Xi'an Jiaotong University, No.28, Xianning West Road, Xi'an, Shaanxi, 710049, PR China
[b] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, No.28, Xianning West Road, Xi'an, Shaanxi, 710049, PR China
[c] Engineering University of PAP, Xi'an, Shaanxi, 710086, PR China

ARTICLE INFO

Keywords:
Image retrieval
Local graph diffusion
Mode-seeking
Large-scale clustering

ABSTRACT

In this work, we propose a large-scale clustering method that captures the intrinsic manifold structure of local features by graph diffusion for image retrieval. The proposed method is a mode seeking like algorithm, and it finds the mode of each data point with the defined stochastic matrix resulted by a same local graph diffusion process. While mode seeking algorithms are normally costly, our method is efficient to generate large-scale vocabularies as it is not iterative, and the major computational steps are done in parallel. Furthermore, unlike other clustering methods, such as k-means and spectral clustering, the proposed clustering algorithm does not need to empirically appoint the number of clusters beforehand, and its time complexity is independent on the number of clusters. Experimental results on standard image retrieval datasets demonstrate that the proposed method compares favorably to previous large-scale clustering methods.

1. Introduction

The widespread availability of digital cameras and the advance in storage capacity have enabled the creation of large image datasets. In order to deal with these data, it is necessary to develop appropriate information systems to efficiently manage these collections. Content Based Image Retrieval (CBIR) is one of the most important technologies for this purpose.

The task of CBIR is to find similar images of a given query image from a large collection of image dataset. The last decade has witnessed the great development of CBIR. Most state-of-the-art CBIR technologies [1–5] are based on the Bag-of-Word (BoW) model, introduced in [6]. It quantizes descriptors of local features (such as SIFT [7]) into visual words obtained by performing descriptor clustering on a training set, and then represents an image as the histogram of visual words. When giving a query image, fast access to the images with common words is achieved via an inverted file. Finally, a post-processing step, usually with spatial verification [2,8], is used to improve the retrieval performance.

For image retrieval, especially at large scale, the number of descriptors to be clustered and the number of clusters are typically in the order of 10^7 and 10^6, respectively. Thus, many clustering algorithms, such as mean-shift [9], spectral clustering [10] and random walks based methods [11,12], are ruled out due to their high time complexity. In the literature, the standard k-means algorithm [13] or its variants, such as hierarchical k-means (HKM) [1] and approximate k-means (AKM) [2], is one of the most popular methods for the construction of large-scale visual vocabularies, or codebooks. Although k-means and its variants are simple and popular for codebook construction, they suffer from the following problems: (1) the number of clusters has to be pre-specified; (2) randomly initial assignments of the centroids do not guarantee to obtain the same result with each run; (3) these methods often choose suboptimal codebooks in which most of the cluster centers are near high density regions [14].

In this paper, a novel clustering algorithm with a local graph diffusion process capturing manifold structure, is presented to alleviate the aforementioned problems of k-means based clustering algorithms. Graph diffusion, propagating the similarity information on a weighted graph representing a set of data points, is well known by its ability in revealing the intrinsic relation between the data points [15]. Benefiting from its good properties, graph diffusion has been successfully applied to many vision tasks, for instance, nonlinear dimensionality reduction [15], shape retrieval [16] and affinity learning [17]. Although graph diffusion has been studied extensively, to our best knowledge, it has never been studied for clustering high dimensional data points in the order of 10^7. The proposed Local Graph Diffusion based Clustering algorithm, dubbed as LGDC, is essentially a mode-seeking like algorithm. As will be demonstrated, LGDC is non-iterative and can be done in parallel,

* Corresponding author.
E-mail addresses: pangsm@xjtu.edu.cn (S. Pang), jrxue@mail.xjtu.edu.cn (J. Xue), gaozn1990@stu.xjtu.edu.cn (Z. Gao), zhenglh@stu.xjtu.edu.cn (L. Zheng), zhuli@mail.xjtu.edu.cn (L. Zhu).

https://doi.org/10.1016/j.image.2018.01.004
Received 27 May 2017; Received in revised form 3 November 2017; Accepted 16 January 2018
0923-5965/

therefore it is efficient to generate large vocabularies. Compared with AKM and HKM, LGDC takes into account information with a higher order, thus it comes at no surprise that it is capable of obtaining better retrieval performance on benchmark image search datasets. Another three appealing advantages of LGDC are that, (1) it is a deterministic procedure, (2) the number of clusters does not need to be empirically appointed in advance, and (3) the time complexity is independent on the number of clusters. To summarize, the contributions of this paper are as follows:

- Due to high time complexity of graph diffusion, it has never been used for clustering points in the order of 10^7. To our knowledge, this paper is the first to introduce graph diffusion to very large-scale clustering.
- Unlike most existing mode seeking algorithms, LGDC is efficient as it is not iterative, and the major computational steps (Line 1–6 in Algorithm 1) are done in parallel.
- LGDC has several advantages over a common used large-scale clustering algorithm AKM in the literature, and it obtains much better retrieval performance on benchmark image search datasets.

The rest of this paper is organized as follows. Section 2 reviews the related works. Then, we present the proposed large-scale codebook construction method in Section 3. Section 4 supports the proposed clustering method by the experimental results on several image search datasets. Finally, we draw conclusions in Section 5.

2. Related works

In the literature, there are mainly three kinds of retrieval scenarios: text-based [18,19], sketch-based [20,21], and content-based [6,22] image retrieval (CBIR). In CBIR, query is a visual image, and each image is usually represented either by a real-valued vector [23–25] or by a compressed binary vector [26–29]. Our work belongs to CBIR, and we represent images with real-valued vectors.

Many popular real-valued vector representation based CBIR systems [1–5,30,31] represent images with the BoW model, in which vocabulary construction is of great importance to search accuracy. The pioneer work [6] builds a relative small vocabulary with the order of 10^4 visual words using the standard k-means algorithm. As shown in [1,2], searching with such a small vocabulary is effective but not efficient for large scale image search. To scale up the approach [6] to large databases, HKM [1] is the first attempt to construct large vocabularies with a hierarchical k-means tree scheme. Although HKM reduces the time complexity of k-means significantly, image search with vocabularies built by HKM suffers from low retrieval quality on benchmark datasets [2]. Most, if not all, recent image search systems construct large vocabularies with AKM [2], another alteration to k-means. Note that the vast majority of computation time in k-means is spent on calculating nearest neighbors between the points and cluster centers, AKM replaces this exact computation by an approximate nearest neighbor method, which uses multiple randomized k-d trees [32] built over the cluster centers at the beginning of each iteration to increase efficiency. Similarly to AKM, approximate Gaussian mixture (AGM) [31], a more recent method to large scale visual vocabulary learning for image retrieval, also assigns a point to the nearest cluster via approximate nearest neighbor search presented in [2]. As claimed in [31], AGM is a variant of expectation maximization that can be as fast as AKM while allowing dynamic estimation of the number of clusters.

In common, the aforementioned methods all directly use pairwise Euclidean distance between features as the similarity measure in vocabulary construction. In the literature, there also exists several other works that build codebooks in different ways. For example, Philbin et al. [33] learn a projection from the original descriptor space to a new space in which standard clustering techniques with Euclidean metric are more likely to assign matching descriptors to the same cluster, and non-matching descriptors to different clusters. Abandoning the assumption that the descriptor distance provides a good similarity measure, Mikulik et al. [23] first use a fine vocabulary to partition the descriptor space, and then learn a probabilistic relationship between visual words. Although promising results with a 16M vocabulary are reported in [23], the method has quite high time complexity. Similar to the method in [23,33], we also find that the Euclidean distance in descriptor space is not an optimal metric for similarity. But in contrast with them, our method is more simple, and we propose a mode-seeking like clustering algorithm in which each point finds its mode with a local graph diffusion process to solve the problem.

It should be noted that, besides searching with large-scale vocabularies described above, recently there emerges image search systems without codebook training [26] or with a very small codebook [25,34]. Zhou et al. [26] transform SIFT descriptors to 256-bit binary vectors by a scalar quantization scheme. Without training a codebook, this method selects 32 bits from the 256-bit vector as a codeword for indexing and search. Although this method does not require resources to train off-line, it consumes large amount of memory online as it stores more than two hundred bits per feature in the inverted file. VLAD and Temb, which aggregate local descriptors into a compact image vector with a small vocabulary of size typically 64, are introduced by [34] and [25], respectively. Both VLAD and Temb can be considered as extensions of BoW. Although they are more memory efficient than BoW, VLAD and Temb are often inferior to BoW based methods in terms of retrieval accuracy.

It is also possible to improve retrieval quality with a post-processing step. Well known post-processing methods include [2,8,24,35–37]. In a nutshell, these methods employ additional information (e.g., spatial configures of features) that is ignored at the searching step to provide a more precise of the retrieved images. It is worth noting that, the earlier post-processing methods [2,8,35] improve retrieval quality at the cost of considerable additional memory consumption as they store geometrical information of local features. Fortunately, this drawback is overcame by the latter methods [24,36,37]. In this paper, we apply an image level post-processing Query Fusion (QF) [37] in our retrieval system to illustrate our clustering method is compatible with post-processing.

3. Vocabulary construction with local graph diffusion

3.1. Local graph diffusion

Let us denote $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_i, \dots, \boldsymbol{x}_N\}$ as a set of data points, and $\boldsymbol{x}_{i,j} \in \mathcal{X}$ $(j = 1, \dots, k)$ as the k nearest neighbors of $\boldsymbol{x}_i$. As points that are far away from $\boldsymbol{x}_i$ are usually meaningless to $\boldsymbol{x}_i$, so for each point $\boldsymbol{x}_i$ we perform the diffusion process only within its neighbors. We describe the relationship among $\boldsymbol{x}_i, \boldsymbol{x}_{i,1}, \dots, \boldsymbol{x}_{i,k}$ with an edge-weighted graph $G_A = (\mathcal{V}, \boldsymbol{A})$, where $\mathcal{V} = \{\boldsymbol{x}_i, \boldsymbol{x}_{i,1}, \dots, \boldsymbol{x}_{i,k}\}$ is the set of vertices, and $\boldsymbol{A}$ is the graph adjacency matrix with each entry representing the edge weight from one vertex to another. We consider $\mathcal{V}$ as the neighbor set of $\boldsymbol{x}_i$ and rewrite $\mathcal{V}$ as $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_{k+1}\}$ for convenience, where $\boldsymbol{y}_1 = \boldsymbol{x}_i$ and $\boldsymbol{y}_{j+1} = \boldsymbol{x}_{i,j}$, $j = 1, \dots, k$. Consequently, the (m,n)-th $(m, n = 1, \dots, k+1)$ entry of $\boldsymbol{A}$ can be written as a_{mn}, and a_{mn} is defined as:

$$a_{mn} = f(\boldsymbol{y}_m, \boldsymbol{y}_n), \tag{1}$$

where f is a symmetric (i.e., $f(\boldsymbol{y}_m, \boldsymbol{y}_n) = f(\boldsymbol{y}_n, \boldsymbol{y}_m)$)and nonnegative function that reflects the similarity between data points $\boldsymbol{y}_m$ and $\boldsymbol{y}_n$. Thus, the matrix $\boldsymbol{A}$ is obviously symmetric and nonnegative.

After getting the affinity matrix $\boldsymbol{A}$, we transform it to a column stochastic matrix $\boldsymbol{P}$:

$$p_{mn} = \frac{a_{mn}}{\sum_{m=1}^{k+1} a_{mn}}, \quad m, n = 1, 2, \dots, k+1. \tag{2}$$

Thus, a Markov random walk on the graph G_A is defined, and p_{mn} can be interpreted as the probability of transition from $\boldsymbol{y}_n$ to $\boldsymbol{y}_m$. As known,