Fast Image Retrieval Based on Two-dimensional Embedding

Kui Zhao[1,a], Shihan Wang[1,b], Wei Wang[1,c]

[1]Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, China

[a]zhaokui@zju.edu.cn, [b]wangshihan9@gmail.com, [c]wangwei_eagle@zju.edu.cn

Keywords: Image Retrieve; Two-dimensional Embedding; Nearest Neighbor Search

Abstract. Recent years see an explosive growth of digital images and we need efficient technologies to index and retrieve them. The content-based retrieval method is a hot topic in recent research, in which we index the features of image and then retrieve the images through nearest neighbor searching. The features of an image dwell in high dimensional space, which poses particular challenge for nearest neighbor search. The curse of dimensionality raises severe difficulty in traditional methods. In this paper, we proposed a novel method for nearest neighbor search in high dimensional space based on two-dimensional embedding. First, we filter out the non-nearest neighbors using low dimensional information via data embedding, and then efficient nearest neighbor search can be performed in a much smaller candidate set to achieve fast image retrieval. The experimental results on the dataset CIFAR validate the effectiveness and efficiency of our method.

Introduction

The rapid development of the Internet triggers an explosion of the available information, especially complex information such as images and videos. Efficiently indexing and retrieving huge volumes of images become a hot topic both in academia and industry.

Traditional methods for indexing and retrieving of images are dependent on manual text marking. Once we have textual information, such as titles or keywords, the problem is transformed into textual retrieval problem. There are a lot of well-developed methods to do textual retrieval. However, because the textual information marked by hand is usually not sufficient enough to represent the images, the performance of this kind of methods is not very satisfying. What's more, the manual text marking on large data set is very expensive. Another direction is indexing and retrieving the images based on the content of the images [1]. In these methods, when retrieving an image we return the k nearest neighbors, which are k nearest images in the content space of these images. The nearest neighbor search (NNS) algorithm is the most important part for this kind of methods, which almost decides the overall speed and precision of the content-based retrieval.

In this paper, we proposed a fast method for image retrieval based on two-dimensional embedding NNS. We filter out the non-nearest neighbors, which are in high dimensional space, only using low dimensional information through data embedding. Then we can search the nearest neighbor in a much smaller candidate set in which the data points are in high dimensional space. This strategy makes it very fast to find the nearest neighbor and our experimental results on the dataset CIFAR validate the effectiveness and efficiency of our method compared with other methods.

Related Work

The KD-Tree proposed by Smeulders, A. W. etc. is a very famous algorithm based on space division for NNS [2]. They divide the data space equally according to variance recursively and query a point using the relation between query point and division plane. Y Beis, J. S. and Lowe, D. G [3] presented an approximate nearest searching by introducing the best bin first strategy. The vantage point tree [4] and multi vantage point tree [5] are both based on triangle inequality for space division. The performance of those methods is very good in low dimensional space, but the performance declines significantly in high dimensional space.

In order to achieve better performance in high dimensional space, many methods have been proposed. The locality-sensitive hashing [6] reduces dimension according to probability information. The hierarchical K-means tree method [7] uses clustering to construct a tree and choose searching path according to the distances between the query point and center of clusters. The Cover-Tree [8] also constructs a tree data structure for NNS. However, the performance of these methods is also not good enough for high dimensional data.

Two-dimensional Embedding for Nearest Neighbor Searching

The data embedding is used widely in processing high dimensional data. In this paper, we embed the high dimensional data into a low dimensional space inspired by the lower and upper bound of Euclidean distance [9]. For point **x** and **y** in Euclidean space,

$$\mathrm{dist}(\mathbf{x},\mathbf{y})^2 \geq d \times [(\mu_{\mathbf{x}} - \mu_{\mathbf{y}})^2 + (\sigma_{\mathbf{x}} - \sigma_{\mathbf{y}})^2], \tag{1}$$

$$\mathrm{dist}(\mathbf{x},\mathbf{y})^2 \leq d \times [(\mu_{\mathbf{x}} - \mu_{\mathbf{y}})^2 + (\sigma_{\mathbf{x}} + \sigma_{\mathbf{y}})^2]. \tag{2}$$

Where d is the dimension of $\mathbf{x}$, $\mu_{\mathbf{x}}$ and $\sigma_{\mathbf{x}}$ are the mean and variance value of $\mathbf{x}$ over its entire dimension respectively. We project the high dimensional data point **x** to two two-dimensional points $(\mu_{\mathbf{x}}, \sigma_{\mathbf{x}})$ and $(\mu_{\mathbf{x}}, -\sigma_{\mathbf{x}})$, represented by x^+ and x^-. So the lower bound of squared Euclidean distance between **x** and **y** is $d \times \mathrm{dist}(x^+, y^+)^2$ and the upper bound is $d \times \mathrm{dist}(x^+, y^-)^2$.

Then we can design a fast method for nearest neighbor search using vantage point tree (VP-Tree) and two-dimensional embedding mentioned above. The algorithm is shown in Algorithm 1.

Algorithm 1 NNS Based on Two-dimensional Embedding

Input: Searching data point set {**x**}, dimenson d of **x** and query point **q**;
Output: Nearest neighbor point for query point **q**;
Initialization: candidate = NULL, min_dist = ∞
Caculate the projection set $\{x^+\}$ and $\{x^-\}$ for {**x**};
Construct VP-Tree for $\{x^+\}$ and $\{x^-\}$, denoted by VPT^+, VPT^-;
Caculate the projection q^+ for **q**;
Find the nearest point to q^+ in VPT^- and denote the distance as U_{min};
Find the points in VPT^+ whose distance to q^+ are less than U_{min},
and put these points in the set $\{\mathbf{x}\}^*$;
for each $\mathbf{x}_i$ **in** $\{\mathbf{x}\}^*$
  if $d \times \mathrm{dist}(x_i^+, q^+)^2 \geq \mathrm{min\_dist}^2$
    continue;
  else
. **if** $\mathrm{dist}(\mathbf{x}_i, \mathbf{q})^2 < \mathrm{min\_dist}^2$
. candidate = $\mathbf{x}^i$, $\mathrm{min\_dist}^2 = \mathrm{dist}(\mathbf{x}_i, \mathbf{q})^2$;
. **end**
. **end**
. **Return** candidate

As we can see, after filtering the non-nearest neighbor points, we can search the nearest neighbor point in a much smaller set $\{\mathbf{x}\}^*$. For a special data set, we only need to run step 1-2 of Algorithm 1 one time. When we query point **q**, we run Algorithme from step 3 to the end.

Experimental Results

In the experimental evaluation, the data set we have used is CIFAR [10], which contains 50,000 color images. The image in this data set has 32×32 pixel points, each pixel point is represented by three number which ranges in [0, 255], so the dimension of each image is 32×32×3 = 3072. Most of methods for nearest neighbor search cannot work efficiently in such a high dimensional space. Besides, we construct two new data sets from the original one, the size of images in them are 24×24

and 16×16 respectively. For each data set, we choose 100 images randomly as query set and the remaining 49,900 images as data set for searching.

First, we focus on the filtering rate of our method. The experiments on different data sets will show us the dimension of data affect the filtering rate. The filtering rates of our method on different data sets are shown in Table1.

Table1 The filtering rates on different data sets

Size of image	Filtering rate max	Filtering rate max	Filtering rate mean	Filtering rate variance
16×16	96.8%	24.6%	59.4%	0.042
24×24	96.8%	23.3%	58.2%	0.043
32×32	95.4%	12.8%	46.8%	0.060

As we can see, the filtering rates of query points are quite different. The maximum filtering rates are all above 95%, but the minimum filtering rate and average filtering rate decrease when the dimension increases. What's more, the filtering rate is more unstable in high dimensional space.

In order to indicate the effectiveness of our method, we compare its searching time with Cover-Tree [10], and the baseline is linear scanning. Linear scanning will compare the query point with all points in the searching data set, which is the worst case. The average query time for different method is shown in Table2.

Table2 The average time for different methods

Size of image	Our Method	Cover-Tree	Linear Scanning
16×16	0.0996s	0.142s	0.268s
24×24	0.0867s	0.282s	0.606s
32×32	0.248s	0.416s	1.03s

We also normalize the average query time by setting the time of linear scanning to be 1.0. The results are presented in Table3.

Table3 The normalized average time of different methods

Size of image	Our Method	Cover-Tree	Linear Scanning
16×16	0.188	0.530	1.0
24×24	0.186	0.466	1.0
32×32	0.241	0.404	1.0

As we can see, the efficiency of our method is improved at least 4 times compared with linear scanning and 1.6 times compared with Cover-Tree. What's more, the efficiency improvement in low dimensional space is even more significant.

Summary

We have proposed a novel method for nearest neighbor search based on two-dimensional embedding. We filter out the non-nearest neighbors using low dimensional information, and then search the nearest neighbor much faster in a smaller candidate set. The experimental results on the dataset CIFAR indicate the effectiveness and efficiency of our method. The method we proposed can be used for fast content-based image retrieval or other high dimensional data retrieval, such as video retrieval.

For future work, we consider making our method more stable for different query points through better data embedding algorithm. We also consider designing a more efficient strategy to choose the threshold for filtering out the non-nearest neighbors. This way, we can apply our method in many practical situations and optimize its performance accordingly, as well as making it more robust. All of these works will improve the practicability of our method for searching the nearest neighbor in high dimensional space.

Acknowledgement

This work is supported by National Key Technology R&D Program (Grant No. 2012BAI34B01).

References

[1] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(12), 1349-1380.

[2] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509-517.

[3] Y Beis, J. S., & Lowe, D. G. (1997, June). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on (pp. 1000-1006). IEEE.

[4] Yianilos, P. N. (1993, January). Data structures and algorithms for nearest neighbor search in general metric spaces. In Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms (pp. 311-321). Society for Industrial and Applied Mathematics.

[5] Bozkaya, T., & Ozsoyoglu, M. (1997, June). Distance-based indexing for high-dimensional metric spaces. In ACM SIGMOD Record (Vol. 26, No. 2, pp. 357-368). ACM.

[6] Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. (2004, June). Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the twentieth annual symposium on Computational geometry (pp. 253-262). ACM.

[7] Fukunaga, K., & Narendra, P. M. (1975). A branch and bound algorithm for computing k-nearest neighbors. Computers, IEEE Transactions on, 100(7), 750-753.

[8] Beygelzimer, A., Kakade, S., & Langford, J. (2006, June). Cover trees for nearest neighbor. In Proceedings of the 23rd international conference on Machine learning (pp. 97-104). ACM.

[9] Hwang, Y., & Ahn, H. K. (2011). Convergent bounds on the euclidean distance. In Advances in Neural Information Processing Systems (pp. 388-396).

[10] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. Computer Science Department, University of Toronto, Tech. Rep.