Available at: www.researchpublications.org

Data Mining With Big Data Image De-Duplication In Social Networking Websites

Hashmi S.Taslim
First Year ME
Jalgaon
faruki11@yahoo.co.in

ABSTRACT

Big data is the term for a collection of data sets which are large and complex.Data comes from everywhere,sensor used to gather climate information,posts to social media sites and video.This data is known as Big data.useful data can be extracted from this Big data with the help of data mining we propose an efficient approach based on the search of closed patterns. Moreover,we present a novel way to encode the bag-of-words image representation into data mining transactions. We validate our approach on a new dataset of one million Internet images obtained with random searches on Google image search. Using the proposed method, we find more than 80 thousands groups of duplicates among the one million images in less than three minutes while using only 150 Megabytes of memory. Unlikeother existing approaches, our method can scale gracefully to larger datasets as it has linear time and space (memory) complexities.We propose an efficient way of storing and De-Duplication of images on server of On-line Social Networks. In this approach server will maintain only one copy of image on server and provides access to all users who have uploaded it. This is achieved through a flexible rule-based system that allows users to upload images on server, and in background before storing image on server it will check whether any duplicate image is exist or not, if image is already available then it will not upload this image, instead of it server will tag this user with old image.we focus primarily the method requires only a small amount of data need be stored.we demonstrate our method on the Trec 2006 data set which contain approximately 146k key frames.The propsed method uses a Visual vocabulary of vector quantized local feature descriptor(SURF) and for retrival exploits enhanced min hash techniques.The algorithm select min-hash algorithm.

General Terms

visual vocabulary of vector quantized local feature descriptors (SURF)and for retrieval exploits enhanced min-Hash techniques

Keywords

Big data,Data minig,min-hash,image compression

1. INTRODUCTION

Querying 'Paris' on an image search engine such as Google image search returns more than two billion links to image files spread over the Internet. A quick glance at the first page of results reveals quite a few similar images of Eiffel tower. This simple observation suggests two conclusions (a) the number of images on the Internet is unimaginable and (b) interms of true content, there is potentially a high amount of redundancy. Such redundanciesare natural on the Internet as different entities (e.g. news websites, website designers) mightobtain their original content from the same source (e.g. Reuters, commercial image galleries,respectively), slightly modify and reuse them. Even the same entity (e.g. people) might upload the same images, or slightly modified versions, on different sites (Flickr, Facebook etc.) simultaneously

We also proposes two novel image similarity measures for fast indexing via locality sensitive hashing. The similarity measures are applied and evaluated in the context of near duplicate image detection. The proposed method uses a visual vocabulary of vector quantized local feature descriptors (SURF) and for retrieval exploits enhanced min-Hash techniques. Standard min-Hash uses an approximate set intersection between document descriptors was used as a similarity measure.

2. Image Representation and Similarity Measures

Recently, most of the successful image indexing approaches are based on the bag-of visual-words representation. In this framework, for each image in the data set affine invariant interest regions are detected. Popular choices are MSER, DoG (difference of Gaussians) [14] or multi-scale Hessian interest points . Each detected feature determines an affine covariant measurement region, typically an ellipse defined by thesecond moment matrix of the region. An affine invariant descriptor is then extracted from the measurement regions. Often a 128-dimensional SURF descriptor is used. A 'visual vocabulary' is then constructed by vector quantization of feature descriptors. Often, k-means or some variant is used to build the vocabulary . The image database or a random subset can be used as the training data for clustering. The k-means cluster centers define visual words and the SURF features in every image are then assigned to the nearest cluster center to give a visual word representation

Assume a vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$ where each visual word is encoded with unique identifier from . {1,.., $\mathcal{V}$}A bag-of-visual-words approach represents an image by a vector of length $|\mathcal{V}|$, where each element denotes the number of features in the image that are represented by given visual word.a set $\mathcal{A}_i$ of words $\mathcal{A}_i \subset \mathcal{V}$ is a weaker representation that does not store the number of features but only whether they are present or not.

Set similarity. The distance measure between two images is computed as the similarity of

Sets $\mathscr{A}_i$ and $\mathscr{A}_i$ which is defined as the ratio of the number of elements in the intersection over the union:

$$\mathrm{sim}_s(\mathscr{A}_1, \mathscr{A}_2) = \frac{|\mathscr{A}_1 \cap \mathscr{A}_2|}{|\mathscr{A}_1 \cup \mathscr{A}_2|}.$$

This similarity measure is used by text search engines to detect near-duplicate text documents. In NDID, the method was used .The efficient algorithm for retrieving near duplicate documents, called min-Hash.

.

Weighted set similarity The set similarity measure assumes that all words are equally important.Here we extend the definition of similarity to sets of words with differing importance. Let dw ≥ 0 be an importance of a visual word Xw. The similarity of two sets $\mathscr{V}$ and $\mathscr{V}$ is

$$\mathrm{sim}_w(\mathscr{A}_1, \mathscr{A}_2) = \frac{\sum_{X_w \in \mathscr{A}_1 \cap \mathscr{A}_2} d_w}{\sum_{X_w \in \mathscr{A}_1 \cup \mathscr{A}_2} d_w}.$$

Histogram intersection.

Let ti be a vector of size $|\mathscr{V}|$ where each coordinate t_i^w is the number of visual words Xw present in the i-th document. The histogram intersection measure is defined as

$$\mathrm{sim}_{h_0}(\mathscr{A}_1, \mathscr{A}_2) = \frac{\sum_w \min(t_1^w, t_2^w)}{\sum_w \max(t_1^w, t_2^w)}.$$

This measure can be also extended using word weightings to give:

$$\mathrm{sim}_h(\mathscr{A}_1, \mathscr{A}_2) = \frac{\sum_w d_w \min(t_1^w, t_2^w)}{\sum_w d_w \max(t_1^w, t_2^w)}.$$

3. Min Hash Background

we describe how a method originally developed for text near-duplicate detection is adopted to near-duplicate detection of images. Two documents are near duplicate if the similarity sims is higher than a given threshold ρ. The goal is to retrieve all documents in the database that are similar to a query document. This section reviews an efficient randomized hashing based procedure that retrieves near duplicate documents in time proportional to the number of near duplicate documents. The outline of the algorithm is as follows: First a list of min-Hashes are extracted from each document. A min-Hash is a single number having the property that two sets $\mathscr{V}$ and $\mathscr{V}$ have the same value of min-Hash with probability equal to their similarity $\mathrm{sim}_s(\mathscr{A}_1, \mathscr{A}_2)$. For efficient retrieval the min-Hashes are grouped into n-tuples called sketches. Identical sketches are then efficiently found using a hash table. Documents with at least h identical sketches (sketch hits) are considered as possible near duplicate candidates and their similarity is then estimated using all available min-Hashes

3.1 min-Hash algorithm.

A number of random hash functions is given assigning $f_i : \mathscr{V} \rightarrow R$ a real number to each visual word. Let Xa and Xb be different words from the vocabulary $\mathscr{V}$. The random hash functions have to satisfy two conditions: f j(Xa) ≠ f j(Xb) and $P(f_j(X_a) < f_j(X_b)) = 0.5$ The functions f j also have to be independent. For small vocabularies,the hash functions can be implemented as a look up table, where each element of the table is generated by a random sample from Un(0,1). Note that each function f j infers an ordering on the set of visual words Xa <j Xb iff f j(Xa) < f j(Xb). We define a min-Hash as a smallest element of a set Ai under ordering induced by function f j

$$m(\mathscr{A}_i, f_j) = \arg\min_{X \in \mathscr{A}_i} f_j(X).$$

For each documentAi and each hash function f j the min-Hashes m($\mathscr{A}_i$, f j) are recorded. The method is based on the fact, which we show later on, that the probability of m($\mathscr{A}_1$, f j)= m($\mathscr{A}_2$, f j) is

$$P(m(\mathscr{A}_1, f_j) = m(\mathscr{A}_2, f_j)) = \frac{|\mathscr{A}_1 \cap \mathscr{A}_2|}{|\mathscr{A}_1 \cup \mathscr{A}_2|} = \mathrm{sim}_s(\mathscr{A}_1, \mathscr{A}_2)$$

To estimate sim_s ($\mathscr{A}_1$,$\mathscr{A}_2$), N independent hash functions f j are used. Let l be the number of how many times m($\mathscr{A}_1$, f j) = m(A2, f j). Then, l follows the binomial distribution Bi(N,sim_s(A1,$\mathscr{A}_2$)).The maximum likelihood estimate of sim_s ($\mathscr{A}_1$,$\mathscr{A}_2$) is l/N.

3.2 How does it work?

let $X = m(\mathscr{A}_1 \cup \mathscr{A}_2, f_j)$Since f j is a random hash function, each element of $\mathscr{A}_1 \cup \mathscr{A}_2$ has the same probability of being the least element. Therefore, we can think of X as being drawn at random from $\mathscr{A}_1 \cup \mathscr{A}_2$. If X is an element of both $\mathscr{A}_1$ and $\mathscr{A}_2$, i.e. X ε $\mathscr{A}_1 \setminus \mathscr{A}_2$, then m($\mathscr{A}_1$, f j) = m($\mathscr{A}_2$, f j) = X.Otherwise either X ε $\mathscr{A}_1 \setminus \mathscr{A}_2$and X = m($\mathscr{A}_1$, f j) ≠ m($\mathscr{A}_2$, f j); or X ε $\mathscr{A}_2 \setminus \mathscr{A}_1$and m($\mathscr{A}_1$, f j) ≠ m($\mathscr{A}_2$, f j) = X. The equation (5) states that X is drawn from $|\mathscr{A}_1 \cup \mathscr{A}_2|$ elements at random and the equality of min-Hashes occurs in $|\mathscr{A}_1 \cap \mathscr{A}_2|$ cases.

4.Experimental Results

We demonstrate our method for NDID on two data sets: the TrecVid 2006 data set and the University of Kentucky data set.

4.1TrecVid 2006

TrecVid [21] database consists of 146,588 JPEG keyframes automatically pre-selected from 165 hours (17.8M frames, 127 GB) of MPEG-1 news footage, recorded from different TV stations from around the world. Each frame is at a

resolution of 352×240 pixels and normally of quite low quality.

Figure 1 displays the number of sketch hits plotted against the similarity measures of the colliding documents. a vocabulary of 64K visual words, N =192 min-Hashes, sketch size n=3, and k =64 number of sketches were used.

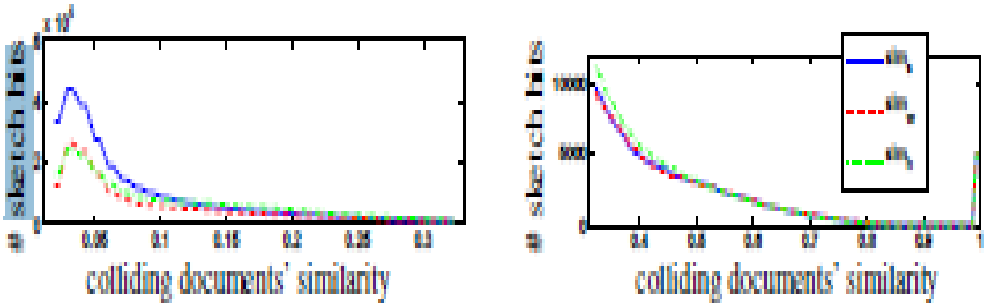

Figure 1: The number of sketch hits as a function of the document similarity for different similarity measures (TrecVid data set). Left: similarities 0 – 0.35, right: high similarities (different scale).

For document pairs with high value of the similarity measure, the number of hits is roughly equal for sims and simw and slightly higher for simh. This means that about the same number of near duplicate images will be recovered by the first two methods and the histogram intersection detects slightly higher number of near duplicates. The detected near duplicate results appear similar after visual inspection and no significant discrepancy can be observed between the results of the methods.

However, for document pairs with low similarity (pairs that are of no interest) using simw and simh similarity significantly reduces the number of sketch hits. In the standard version of the algorithm, even uninformative visual words that are common to many images are equally likely to become a min-Hash. When this happens, a large number of images is represented by the same frequent min-Hash. In the proposed approach, common visual words are down-weighted by a low value of idf. As a result, a lower number of sketch collisions of documents with low similarity is observed. The average number of documents examined per query is 8.5, 7.1, and 7.7 for sims, simw, and simh respectively. Compare this to 43,997.3 of considered documents . using tf-idf inverted file retrieval using a vocabulary of the same size.

4.2 University of Kentucky database

This database contains 10,200 images in sets of 4 images of one object / scene. Querying the database with each image should return three more examples. This is used to score the retrieval by the average number of correctly returned images in top four results (the query image is to be retrieved too). We are probing lower values of the similarity measures due to larger variations between images of the same scene in this data set. Therefore more min-Hashes and more sketches have to be recorded. we varied several parameters of the method: the size of the vocabulary (30k and 100k), the number of independent random hash functions, and the number of hashed sketches. The number of min-Hashes per sketch was set to n = 2. The average number of documents considered (the average number of sketch hits)3 and the average number of correctly retrieved images in the top 4 ranked images were recorded. The results consistently show that the number of sketch hits is significantly decreased while the retrieval score is improved when the idf-weighting is used. The results are further improved when the histogram intersection is used

Some example queries and results are shown in figure 2. It can be seen on the results, that sims often retrieves images based on the object background. The background is repeated on many images and is down-weighted by both simw and simh idf weighting. For comparison, the number of considered documents using standard tf-idf retrieval with inverted files would be 10,089.9 and 9,659.4 for vocabulary sizes 30k and 100k respectively. We are not trying to compete with image or specific object retrieval. The method is designed to find images with high similarity by 'trying out' only a few possibilities. This database is too small to highlight the advantages of rapid retrieval and reduced image representation. Despite this, the scores for the histogram intersection similarity measure simh exceed the score of 3.16 for flat tf-idf .

6 ACKNOWLEDGMENTS

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2011-0029924) and MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program(NIPA-2013-H0301-13-3006) supervised by the NIPA (National IT Industry Promotion Agency).

		documents considered						top 4 score					
		vocab 30k			vocab 100k			vocab 30k			vocab 100k		
mh	ske	sim_s	sim_w	sim_h	sim_s	sim_w	sim_h	sim_s	sim_w	sim_h	sim_s	sim_w	sim_h
512	256	553.8	362.2	**207.0**	143.8	87.3	**49.3**	2.54	2.54	**2.67**	2.43	2.42	**2.57**
512	512	908.6	664.1	**394.6**	281.3	181.1	**94.4**	2.70	2.72	**2.85**	2.65	2.68	**2.80**
512	1024	1671.9	1200.1	**730.9**	543.0	340.2	**178.7**	2.74	2.79	**2.94**	2.80	2.85	**2.97**
512	1536	2325.4	1626.8	**1041.3**	871.4	469.6	**260.7**	2.75	2.80	**2.96**	2.81	2.90	**3.03**
640	320	657.4	434.3	**255.6**	177.0	107.2	**60.4**	2.65	2.65	**2.77**	2.54	2.53	**2.67**
640	640	1141.9	810.2	**488.3**	340.2	206.5	**117.7**	2.76	2.81	**2.93**	2.73	2.77	**2.89**
640	1280	1924.3	1443.4	**889.4**	642.9	396.5	**225.5**	2.80	2.86	**3.01**	2.84	2.92	**3.04**
640	1920	2691.4	1949.0	**1258.4**	969.7	567.0	**330.7**	2.80	2.87	**3.02**	2.88	2.96	**3.09**
768	384	748.5	520.5	**302.8**	215.4	127.7	**72.0**	2.71	2.73	**2.84**	2.62	2.62	**2.74**
768	768	1362.3	957.0	**578.2**	419.9	244.7	**140.3**	2.83	2.86	**2.99**	2.81	2.85	**2.95**
768	1536	2242.9	1669.1	**1035.7**	761.2	637.8	**264.1**	2.85	2.90	**3.05**	2.90	2.98	**3.08**
768	2304	2978.1	2230.6	**1423.1**	1154.0	816.5	**382.1**	2.85	2.91	**3.06**	2.91	3.01	**3.13**
896	448	979.0	595.2	**352.5**	251.2	145.5	**83.6**	2.77	2.79	**2.90**	2.69	2.68	**2.80**
896	896	1578.5	1082.2	**683.8**	481.6	275.4	**163.0**	2.86	2.90	**3.03**	2.86	2.90	**3.00**
896	1792	2743.1	1878.6	**1371.7**	869.5	515.1	**318.8**	2.88	2.93	**3.08**	2.94	3.02	**3.13**
896	2688	3398.8	2496.4	**1790.8**	1238.7	734.9	**452.8**	2.87	2.93	**3.09**	2.96	3.05	**3.17**

Table 1: University of Kentucky data set. Number of min-Hashes (mh), number of sketches (ske), number of considered documents, and average number of correct images in top 4 are shown for three similarity measures sim_s, sim_w, and sim_h. Better results (lower for documents considered, higher for top 4 score) are highlighted among the methods.

Figure 2: University of Kentucky data set: sample queries (left column), results (three rows each) for sim_s (top row), sim_w (middle row), and sim_h (bottom row).

7. Conclusions

We have proposed two novel similarity measures whose retrieval performance is approaching the well established tf-idf weighting scheme for image / particular object re- trieval. We show that pairs of images with high values of similarity can be efficiently (in time proportional to the number of retrieved images) retrieved using the min-Hash algorithm. We have shown experimental evidence that the idf word weighting improves both the search efficiency and the quality of the results. The weighted histogram intersection is the best similarity measure (out of the three examined) in both retrieval quality and search efficiency. Promising results on the retrieval database encourage the use of the hashing scheme beyond near duplicate detection, for example in clustering of large database of images

8.REFERENCE

[1] R. Ahmed and G. Karypis, "Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks," Knowledge and Information Systems, vol. 33, no. 3, pp. 603-630, Dec. 2012.

[2] M.H. Alam, J.W. Ha, and S.K. Lee, "Novel Approaches to Crawling Important Pages Early," Knowledge andInformation Systems, vol. 33, no. 3, pp 707-734, Dec. 2012.

[3] S. Aral and D. Walker, "Identifying Influential and Susceptible Members of Social Networks," Science, vol. 337, pp. 337-341, 2012.

[4] A. Machanavajjhala and J.P. Reiter, "Big Privacy: Protecting Confidentiality in Big Data," ACM Crossroads, vol. 19, no. 1, pp. 20-23, 2012.

[5] S. Banerjee and N. Agarwal, "Analyzing Collective Behavior from Blogs Using Swarm Intelligence," Knowledge and Information Systems, vol. 33, no. 3, pp. 523-547, Dec. 2012.

[6] T. C. Hoad and J. Zobel. Fast video matching with signature alignment. In MIR, pages 262–269, 2003.

[7] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation.In IEEE Symposium on Foundations of CS, 2000.

[8] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In Proc. CVPR, 2008.

[9] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In Proc. CVPR, 2007.

[10] A. Joly, O. Buisson, and C. Fr´elicot. Content-based copy detection using distortion-based probabilistic similarity search. IEEE Transactions on Multimedia, to appear, 2007.

[11] A. Joly, C. Frelicot, and O. Buisson. Robust content-based video copy identification in a large reference database. In Proc. CIVR, 2003.

[12] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In ACM Multimedia, 2004.