



HAL
open science

Image re-ranking based on statistics of frequent patterns

Winn Voravuthikunchai, Bruno Crémilleux, Frédéric Jurie

► **To cite this version:**

Winn Voravuthikunchai, Bruno Crémilleux, Frédéric Jurie. Image re-ranking based on statistics of frequent patterns. ICMR '14 ACM International Conference on Multimedia Retrieval, Apr 2014, Glasgow, United Kingdom. 8 pp, 10.1145/2578726.2578743 . hal-01011237

HAL Id: hal-01011237

<https://hal.science/hal-01011237>

Submitted on 23 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Image re-ranking based on statistics of frequent patterns

Winn Voravuthikunchai

Bruno Crémilleux

Frédéric Jurie

University of Caen Basse-Normandie – CNRS UMR 6072 – ENSICAEN, France
firstname.name@unicaen.fr

ABSTRACT

Text-based image retrieval is a popular and simple framework consisting in using text annotations (*e.g.* image names, tags) to perform image retrieval, allowing to handle efficiently very large image collections. Even if the set of images retrieved using text annotations is noisy, it constitutes a reasonable initial set of images that can be considered as a bootstrap and improved further by analyzing image content. In this context, this paper introduces an approach for improving this initial set by *re-ranking* the so-obtained images, assuming that non-relevant images are scattered (*i.e.* they do not form clusters), unlike the relevant ones. More specifically, the approach consists in computing efficiently and on the fly *frequent closed patterns*, and in re-ranking images based on the number of patterns they contain. To do this, the paper introduces a simple but powerful new scoring function. The approach is validated on three different datasets for which state-of-the-art results are obtained.

Categories and Subject Descriptors

I.4 [IMAGE PROCESSING AND COMPUTER VISION]: Image Representation

General Terms

Algorithms, Experimentation, Measurement, Theory

Keywords

Data mining, Frequent patterns, Image re-ranking, Image search

1. INTRODUCTION

Web-image search has become a key feature of well-known search engines such as ‘Google’, ‘Yahoo’, ‘Bing’, *etc.* Given a text query, the search engine has to go through millions of images for retrieving, as quickly as possible, the relevant ones. Most of these search engines are primarily based on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMR '14 Glasgow, Scotland, UK

Copyright 2014 ACM 978-1-4503-2782-4/14/04 ...\$15.00.

| Image I_i | Trans. t_i | rel. | Patterns \mathcal{X}_j |
|-------------|---------------------|------|---|
| I_1 | $\{a_1, a_2, a_3\}$ | yes | $\mathcal{X}_1 = \{a_1\}$ $\mathcal{X}_2 = \{a_4\}$ $\mathcal{X}_3 = \{a_6\}$ $\mathcal{X}_4 = \{a_2, a_3\}$ |
| I_2 | $\{a_1, a_4, a_6\}$ | yes | |
| I_3 | $\{a_1, a_7, a_9\}$ | no | |
| I_4 | $\{a_2, a_3, a_6\}$ | yes | |
| I_5 | $\{a_4, a_5, a_8\}$ | no | |

(a) Initial ranking order.

| Image I_i | \mathcal{X}_j in t_i | $\#\mathcal{X}_j$ in t_i | rel. |
|-------------|---|----------------------------|------|
| I_2 | $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ | 3 | yes |
| I_1 | $\mathcal{X}_1, \mathcal{X}_4$ | 2 | yes |
| I_4 | $\mathcal{X}_3, \mathcal{X}_4$ | 2 | yes |
| I_3 | \mathcal{X}_1 | 1 | no |
| I_5 | \mathcal{X}_2 | 1 | no |

(c) Images re-ranked.

Figure 1: Toy example illustrating the re-ranking of images according to the count of frequent patterns. Details are given Section 3.1

the use of text meta-data such as keywords, tags, and/or text descriptions nearby the images. Since the meta-data do not always correspond to the visual content of the images, the retrievals are usually mixed up with undesirable non-relevant images. However, it has been observed that the so-retrieved images contains enough relevant images – they are made for users that are in general more interested by precision than recall – and that the precision can be improved by re-ranking the initial set of retrieved images.

This re-ranking stage can benefit from the use of the visual information contained in the images, as shown by [15]. Web-image re-ranking can be seen as a binary classification problem where the relevant images belong to the positive class. Although true labels are not provided, it is still possible to build class models based on the two following assumptions: (i) the initial text based search provides a reasonable initial ranking, which is to say that a majority of the top-ranked images are relevant to the query, meaning that classifiers such as SVMs can be trained by using the top-ranked images as (noisy) positive images while the images that are ranked below or even the images from other datasets are treated as negative images (see *e.g.* [4]). (ii) The relevant images are visually similar to each other (at least within groups) while the non-relevant images tend to be not similar to any other images. Graph based re-ranking approaches exploit this second assumption, by modeling the connectivity among retrieved images [18].

Besides the challenge to build a class model from noisy labeled data, a challenging aspect of web image re-ranking is the efficiency of the approach. Indeed, the re-ranking process has to be done on the fly – since the queries from the users are not known in advance – hence limiting the type of algorithm that can be used. Although many approaches in the literature have shown excellent re-ranking results, most of them are computationally too expensive and therefore unsuitable for real web image re-ranking applications.

In this context, this paper proposes a new approach for image re-ranking, building on recent advances in data mining [1, 25]. More precisely, our approach makes use of *frequent pattern mining*, which allows the discovering of sets of image visual features shared by many images. The key idea is that frequent patterns are more likely to occur in relevant images since such images have related content (this is illustrated on a toy example Figure 1). Furthermore, once frequent patterns are extracted, the patterns found in each image infer the connectivity with other images in the database. More precisely, a new scoring function using the statistics of frequent patterns found in images is proposed. This scoring function allows to re-rank images without having to train any expensive classification models, contrarily to [18]. The encoding of images as pattern mining transactions is also critical. We used the adaptive thresholding process of [27], giving the same number of items (*i.e.* features) for each image. This prevents the risk that some non-relevant images have more frequent patterns only because they have more items than the relevant ones. In addition, we also propose efficient mining techniques either based on the extraction of closed patterns [20] in the transposed database or on multiple random projections. Consequently, the proposed approach not only gives excellent results in terms of accuracy, but it is also very fast. Therefore, it is suitable for re-ranking applications. The paper also shows that the approach can be used for more general outlier detection tasks. The approach is validated through extensive experiments on three different datasets, and compared to state-of-the-art algorithms.

The paper is organized as follows: Section 2 discusses related works while Section 3 describes our approach in detail including how to encode images as data mining transactions, how to extract frequent patterns, and how to re-rank images using frequent patterns. Section 4 provides the experimental validation, and the paper is concluded by some discussions and future work given Section 5.

2. RELATED WORK

Image re-ranking has attracted a lot of attention during the last five years. The different approaches in the literature differ in the way they model the class of relevant images, using (i) clustering and cluster centers, (ii) topic models (iii) classification based models, (iv) graph based models or (v) data mining models.

Clustering-based re-ranking algorithms exploit the property that relevant images form clusters, as they share some common visual properties [3, 14]. Images are then ranked according to their distance to one of the cluster centers (non-relevant images are expected to be far from cluster centers since these images are supposed to be very diverse and scattered). However, in practice, the relevant images have large visual variability (*e.g.* side-views and front-views of bicycles look very different to each other) so it is difficult to determine the number of relevant clusters for a given query.

Moreover, how to compare the relevance between the images in different clusters is still an open issue (*e.g.* ‘Is an image belonging to a small cluster but close to its center more relevant than another image which belongs to a bigger cluster but is far from its center?’).

Topic models have been used to deal with the diversity of relevant image content [5, 7]. Brought by the field of natural language processing, these models assume that the content of a document is generated by the set of topics it contains. Within this framework, each image is basically mapped to a lower dimensional topic space representing the strength of each topic in the image. The images are then ranked according to the dominating topics in the entire retrieval set, which means that if an image contains many dominating topics, it is likely to be relevant to the query.

Classification-based approaches have also been used for re-ranking *e.g.* [4, 9, 21]. In this case, a classifier can be trained using the initial top-ranked images as noisy positive training images. The trained classifier is used to give a new – and more relevant – score to each image. The problem is that the selected pseudo-positive and pseudo-negative images may not be truly-positive and truly-negative, and can damage the classification model. Moreover, a new classifier has to be learned for each new query. Krapac *et al.* [15] proposed a method based on query-relative visual word features to train a single generic classifier that can be used across different queries. It is possible to calculate which visual words are strongly associated with the query set, as the majority of the images are the relevant images; the visual words which occur often are the ones strongly associated to the query set. The statistics of the amount of strongly associated visual words can reflect the relevance of the image and can be used as generic features. Thollard and Quénot [24] proposed to combine an unsupervised re-ranking approach with a supervised re-ranking one. The unsupervised approach is based on the hypothesis that a relevant image is visually similar to some other relevant images, while a non-relevant image does not share similarity with any other images. The ranking score of the approach is based on the average distances between the K -nearest neighbors. Regarding the supervised re-ranking approach of [24], the idea is to train a single ‘junk’ classifier to filter out some non-relevant images that are typically found across all queries. These noisy images share some similar characteristics such as they commonly have very small size and are less textured (*e.g.* ‘icons’, ‘banners’). The two re-ranking methods and the original ranking are complimentary and can improve the overall re-ranking.

Graph-based methods perform well in image re-ranking [11, 13, 17, 29]. A fully connected graph is constructed from a query set, graph in which images are nodes and distance between images are vertices. A regularization scheme is applied, hence enforcing the scores to be smooth on the graph while keeping the score consistent with the prior information, (*i.e.* the initial text-based ranking). Unfortunately, such a graph-based approach has very high computational complexity, due to the computation of the distance between all image pairs and the computation of the pseudo-inverse of the adjacency matrix.

Finally, frequent pattern mining has been used for removing outliers in [18]. Each image is described as a transaction (or pattern). A pattern is made of items which are the visual words located on images’ interest points. Frequent pattern mining is applied to find frequent combinations of

visual attributes – constituting new image features – used by a one-class SVM to re-rank the images. Despite our approach also uses patterns, it is very different from [18]. One key difference lies in the way images are encoded as transactions. In [18], as the number of items per image varies a lot from an image to another, non-relevant images (which contain often more patterns as they are often rich in shape and texture) will contain more frequent patterns and will consequently have higher scores. In contrast, we adopted the encoding strategy of [27], for the aforementioned reasons. Another important difference with [18] lies in the way images are ranked: [18] trains a one-class SVM, which is slow when the dimensionality of the data is large, forcing themselves to use very poor image representation. In comparison, the simplicity and the efficiency of our scoring function allow to extract frequent patterns from very high dimension image features (*i.e.* 2,000 visual words and 21 SPM [16] grids, resulting 42,000 features) and use hundreds of thousands frequent patterns to rank images.

3. RE-SCORING OF RETRIEVED IMAGES

As said in the introduction, the rationale for using frequent patterns (*i.e.* frequent groups of visual features jointly occurring images) for re-ranking images is that (1) patterns that occur frequently are likely to come from relevant images, as relevant images do share similarities (in contrast with non relevant images which are scattered) and (2) they can be computed on the fly very efficiently.

This section presents a new scoring function based on frequent patterns and explains how images can be represented as sets of binary items. This binarization step — required for mining patterns as data mining algorithms can only handle binary items — is critical as it provides the information from which the score will be computed.

3.1 Scoring function

Let $\mathcal{A} = \{a_1, \dots, a_k\}$ denotes the set of all possible items. In our case, items are visual words (*i.e.* quantized local features), and \mathcal{A} is the visual vocabulary. A set of items $\mathcal{X} \subseteq \mathcal{A}$, is called a pattern. The set of images is denoted by $\mathcal{I} = \{I_1, \dots, I_{|\mathcal{I}|}\}$, where $|\mathcal{I}|$ represents the number of images. Each image is represented by a pattern and forms a database entry, so-called a *transaction*. The transaction of image I_i is denoted as t_i . The set of transactions \mathcal{T} obtained from the retrieved images is called the *transaction database*, denoted as $\mathcal{T} = \{t_0, \dots, t_{|\mathcal{T}|}\}$. A pattern \mathcal{X} can be covered by (*i.e.* can be subset of) many transactions and the set of transactions covering \mathcal{X} is called the *cover* of \mathcal{X} with respect to the transaction database \mathcal{T} , denoted as $K_{\mathcal{T}}(\mathcal{X})$. More formally, $K_{\mathcal{T}}(\mathcal{X}) = \{k \in \{1, \dots, n\} \text{ s.t. } \mathcal{X} \subseteq t_k\}$. The frequency measure provides the number of occurrences of a pattern \mathcal{X} in the database. \mathcal{X} is considered to be a frequent pattern if its frequency $fr(\mathcal{X}) = |K_{\mathcal{T}}(\mathcal{X})|$ is above a minimum frequency threshold $minfr$ (in other words, $minfr$ is the minimum number of images that must contain a pattern \mathcal{X} in order to consider \mathcal{X} as a frequent pattern).

With all of these notations, we define the set of frequent patterns as :

$$\mathcal{F}(\mathcal{T}, minfr) = \{\mathcal{X} \subseteq \mathcal{A} \text{ s.t. } fr(\mathcal{X}) \geq minfr\} \quad (1)$$

Toy example.

In order to illustrate the approach, let us discuss the toy example shown in Figure 1. In this example, the images are supposed to be retrieved by a text based search engine and sorted by their initial relevant ranking (from 1 to 5). There are three relevant images, I_1 , I_2 , and I_4 and two non-relevant images, I_3 and I_5 (see Figure 1(a)). In the toy example, image I_2 is described by the items a_1 , a_4 and a_6 . The frequency of the pattern $\mathcal{X} = \{a_2, a_3\}$ is 2 and \mathcal{X} is covered by I_1 and I_4 (*i.e.* $K_{\mathcal{T}}(\{a_2, a_3\}) = \{1, 4\}$).

Scoring function.

A first possible scoring function can be:

$$S(I_i) = |\mathcal{F}(\mathcal{T}, minfr) \subseteq t_i| \quad (2)$$

which is the number of frequent patterns included in the transaction representing image I_i . Following our toy example, Figure 1(b) gives the frequent closed patterns which have been extracted with $minfr = 2$ (Section 3.2.2 for the details the mining step). Note that the frequent patterns are found mostly in the relevant images. At last, Figure 1(c) shows the re-ranking of the images according to the number of frequent closed patterns they contain.

The scoring function (Eq. 2) can be improved by using the original ranking. Since the ranking given by the text-based representation is reasonably good, and since this ranking is known, we can use it to weight the frequent patterns. Thus, we do weight each pattern \mathcal{X} by $w(\mathcal{X})$, computed as the sum of the inverse of the original rank of the images containing \mathcal{X} . More formally, $w(\mathcal{X}) = \sum_{k \in K_{\mathcal{T}}(\mathcal{X})} \frac{1}{k}$. As an example, if the pattern \mathcal{X} is found in I_1 , I_3 , and I_5 which are at the images in the first, third, and fifth position of the original ranking, $w(\mathcal{X}) = \frac{1}{1} + \frac{1}{3} + \frac{1}{5}$. According to this weighting scheme, the frequent patterns found in top images should contribute more to the final score.

According to this observation, the improved scoring function is thus:

$$S(I_i) = \sum_{\mathcal{X} \in \{\mathcal{F}(\mathcal{T}, Fmin) \subseteq t_i\}} w(\mathcal{X}) \quad (3)$$

Experiments show that this scoring function significantly improves the performance (Section 4.2).

3.2 Mining image transactions

3.2.1 Representing images by transactions

In order to extract frequent patterns from images, images have to be represented as sets of binary items. Starting from the Bag of Words (BoW) histogram, which is considered as a good choice for representing images [10, 22, 28], the idea is to obtain binary items by thresholding the bins of the BoW histogram. Visual words whose frequencies are above the threshold are set to ‘one’ and are considered as the items of the image. More formally, in the BoW representation, each image I is described as a histogram of visual words $\vec{h} = (p(w_0|I), \dots, p(w_d|I))$ where d is the size of the dictionary. The binary vector is described as $h_i^b = 1 \iff h_i \geq \tau$, where τ is the threshold.

In the context of image re-ranking, the choice of τ is critical. We discuss two possible thresholding alternatives namely *fixed thresholding*, and *adaptive thresholding*.

Fixed thresholding. The simplest way would be to have a fixed constant global threshold $\tau = C$. However, images

with flat BoW could be entirely set to ‘zeros’ or ‘ones’ depending on the threshold value. This could be partly solved by having one threshold per bin/feature. For example, the threshold could be set according to the mean value f_i or the median value \tilde{f}_i of the considered feature, but the number of attributes after binarization (*i.e.* the number of ones) would still vary a lot. As images with very few items will have low score by construction (remind that the score is computed from the number of frequent patterns), this alternative is not adequate with the proposed scoring function.

Adaptive thresholding. We summarize here the adaptive thresholding procedure introduced in [27]. This method sets τ to the top- K most frequent visual word value, $\tau = h_k^s$ where h^s is denoted as a vector in which its components correspond to the bins of h sorted in descending order. Selecting the visual words which have the highest counts is favorable since they are the most representative features in the image. The main advantage of this binarization is that images will have exactly the same number of items, which ensures that the number of frequent patterns found in each image truly reflects the connectivity to other images. Using other binarization methods, may allow non-relevant images to have more frequent patterns than the relevant ones, due to the possibility of having more items in the transactions. Furthermore, having a fixed number of attributes also makes the implementation more efficient. For example, it allows to adopt fixed array sizes which is much more efficient in terms of both computation time and memory usage (the exact need of memory is known in advance).

3.2.2 Efficient frequent pattern mining

Once images are represented as sets of items, the next step is to extract *frequent patterns* [1] (see definition Section 3.1). As each subset of a frequent pattern is also a frequent pattern, the entire set of frequent patterns can be very large and can include redundant information (*i.e.* many frequent patterns are extracted from the same set of images). To reduce this redundancy, we consider two condensed-representations of frequent patterns: (i) *frequent closed patterns* [19], and (ii) *frequent maximal pattern* [2]. Regarding the definitions, \mathcal{X} is a frequent closed pattern if \mathcal{X} is frequent and $\nexists Y \supset \mathcal{X} \mid fr(\mathcal{X}) = fr(\mathcal{Y})$ where \mathcal{Y} is any superset of \mathcal{X} . A closed pattern summarizes the frequency of a subset of patterns having the same frequency value. On the other hand, \mathcal{X} is a maximal frequent pattern, if \mathcal{X} is frequent and $\nexists Y \supset \mathcal{X} \mid fr(\mathcal{Y}) \geq minfr$ where \mathcal{Y} is any superset of \mathcal{X} . Maximal patterns are the longest patterns (w.r.t. the items) satisfying the *minfr* threshold. All frequent patterns can be derived from both condensed representations but the difference is the frequency values: the exact frequencies of all the frequent patterns can be derived from the frequent closed patterns but not from the frequent maximal patterns. Mining closed or maximal patterns also significantly enhances the computing effort [25]. Indeed, specific pruning techniques make the mining of closed and maximal frequent patterns much more efficient than the mining of the whole set of frequent patterns. Moreover, since the number of frequent closed patterns is much less than all frequent patterns, the computational cost of the re-ranking stage is reduced. In practice, we experimented with frequent patterns, frequent closed patterns, and frequent maximal patterns and concluded from these experiments that frequent closed patterns gave the best results both in terms of efficiency and perfor-

| Image I_i | Trans. \mathcal{T}_i | Items a_j | Trans. \mathcal{T}_j |
|-------------|------------------------|-------------|------------------------|
| I_1 | $\{a_1, a_2, a_3\}$ | a_1 | $\{I_1, I_2, I_3\}$ |
| I_2 | $\{a_1, a_4, a_6\}$ | a_2 | $\{I_1, I_4\}$ |
| I_3 | $\{a_1, a_7, a_9\}$ | a_3 | $\{I_1, I_4\}$ |
| I_4 | $\{a_2, a_3, a_6\}$ | a_4 | $\{I_2, I_5\}$ |
| I_5 | $\{a_4, a_5, a_8\}$ | a_5 | $\{I_5\}$ |
| | | a_6 | $\{I_2, I_4\}$ |
| | | a_7 | $\{I_3\}$ |
| | | a_8 | $\{I_5\}$ |

(a) Original database.

(b) Transposed database.

Figure 2: Toy example of the original transaction database in (a) and the transposition of the transaction database in (b).

mance.

Improving mining efficiency.

The mining complexity is linear with the number of images but can grow exponentially with the number of items, which can be unfortunately very large. We investigated two solutions to make the mining process more efficient: (i) mining frequent patterns from the transposed data, (ii) reducing the number of items per transaction by applying multiple random projections.

Matrix transposition. [20] proposed to use the *Galois connection* property to solve the problem of mining patterns from high number of items with respect to the number of transactions. The principle is to transpose the original data matrix and then to extract the closed patterns from the transposed data matrix. Thanks to the Galois connection, we can infer the results that would be extracted from the initial data matrix by associating the closed patterns from the transposed matrix with the closed patterns from the initial matrix. Finally, the same set of closed patterns with their frequencies are extracted, but much more efficiently. We use this transposition trick when the query dataset has few images. In other words, instead of considering an image as a data mining transaction with binary items, each image item is now considered as a data mining transaction containing a few images. Figure 2 shows the transposition of the data. In the following description of the method, we still use the terms of the original database.

Multiple random projections. It has been shown that representing high-dimensional data by multiple projections leads to good approximations of the data *e.g.* [12]. Therefore, we propose a binarization framework consisting in (i) projecting the high dimensional feature space into several low dimensional sub-spaces by applying P random projections, (ii) binarizing features using top- K binarization, (iii) extracting and counting frequent patterns found in each subspace. Note that the combination of the first two-steps bears similarity with local sensity hashing (LSH) [8]. The difference is that in LSH a fixed threshold is used instead of top- K . We have mentioned the advantages of top- K over the fixed threshold in Section 3.2. Moreover, in practice, the projection is done by randomly selecting p visual words from the original BoW. This can be seen as projecting the original d -dimensional data to a p -dimensional subspace where the projection matrix is obtained by randomly selecting p basis vectors among the d ones from the original space (more complex projections have been investigated, without improving

the performance). As an alternative to multiple random projections, we also experimented with principal component analysis (PCA) to reduce the dimensionality of input histograms. However, we obtained much worse results.

4. EXPERIMENTS

This section provides the experimental validation of the proposed approach. We first describe the datasets and then present and analyze the experimental results.

For extracting frequent closed patterns, we use the LCM mining tool [25], setting $minfr = 2$ (*i.e.* a pattern is frequent if it appears in at least two images).

The visual binary attributes are obtained by binarizing a bag-of-words (BoW) representation. To compute the BoW, we use multi-scale SIFT as local descriptors, computed by the VLFeat library [26] (12 scales from 3 up to 14 pixels with the default step size of 2 pixels). The visual words are pooled from 3-level SPM [16] grids (1×1 , 2×2 , 4×4). To binarize the BoW representation we use the adaptive thresholding of [27].

4.1 Datasets and evaluation protocol

The approach is validated on the three following datasets: the INRIA Web Queries dataset [15], the QUAERO’s visual concepts image dataset [23], and finally the eBay Motorbike dataset [6].

The **INRIA Web Queries dataset** consists of top-ranked images from text queries returned by a web search engine. In total, there are 71,478 images from 353 queries, having about 200 images per query. For each query, about 40% of the images are relevant to the query. The queries are very diverse, ranging from general object classes or scenery classes such as ‘car’, ‘bird’, ‘mountain’, *etc.* to specific names of objects, places, events, or persons such as ‘Nike Logo’, ‘Eiffel tower’, ‘Cannes festival’, ‘Cameron Diaz’, *etc.* For each query, the annotation giving the relevance to the query is provided. The evaluation protocol is as follows. For each query, the images are sorted according to their ranking score. The *Average Precision AP* is calculated per each query and the *mean Average Precision mAP* is reported.

The **QUAERO’s visual concepts image dataset** is similar to the INRIA Web Queries dataset in the way the images are obtained. The diversity of the queries, type of images, as well as the intra variations in each query, are also similar to the INRIA Web Queries dataset. The main difference is that the number of images per concept is larger with about 950 images per concept. Moreover, since the dataset was created after the INRIA Web Queries dataset, web-search engines had been improved, resulting in cleaner data. For each concept, about 55% of the images are relevant images. The number of concepts is also larger consisting of 519 concepts. In total, this dataset contains 187,029 images. We follow the same evaluation protocol as used in [24] by using 100 concepts as test data to report the *mAP* over the 100 test concepts.

The **eBay Motorbike dataset** contains 5,245 images of different types of motorbikes collected from eBay. The dataset was designed for studying the problem of removing outlier images from a large dataset, rather than re-ranking web images. Since the images are for advertising, the quality of the images are better than web images. The dataset is also much cleaner with 97% of relevant images (*i.e.* motorbike images) compare to the INRIA Web Queries and

the QUAERO’s visual concepts datasets which have about 40-60% of relevant images per query. The metric used to evaluate the performance is the *Equal Error Rate (EER)*.

4.2 On setting the system

This section justifies the chosen options regarding the algorithm and studies the sensitivity of the parameters. For doing this, we used a sub-part of the INRIA Web Queries dataset, by randomly selecting 30 queries (out of the 353 queries).

Transposed VS Random projections. As mentioned in Section 3.2.2, we proposed two methods for making frequent pattern mining more efficient, either by mining the transposed database or by doing random projections. We compared the two alternatives, computed from the same BoW representation. We run grid-search to obtain the best hyper-parameters (*i.e.* K for the first method, and P , p , K for the second) which gives the highest score for each methods on a validation set.

In terms of effectiveness, the performance of both methods are comparable (the difference in *mAP* is of 0.8). However, in terms of complexity, the random projections approach is better since it is linear according to the number of images. The computational time of pattern extractions process between the two methods are compared in Table 3. Note that for the random projections method, all sub-processes (*i.e.* transaction encoding and pattern extraction of each random projection) are independent to each other, and therefore, the computation for the sub-processes can be run in parallel. In Table 3, the number of CPUs used is equal to the number of random projections which is 50. Note that for the matrix transposition approach, the computational time for extracting frequent patterns from some queries (*i.e.* ‘Map World’ and ‘Logo Chelsea’) is very large. This is due to the number of items (*i.e.* images) of some transactions (*i.e.* features) can be very large. The reason is that these queries contain a lot of duplicates or near duplicates, and some features can appear in all of them. On the other hand, the computational time of the random projection approach is quite stable across different queries, since the number of items (*i.e.* features) per transaction (*i.e.* images) is fixed. To conclude this experiment, when multi-process resource is available and especially when the number of images is large, the random projection method is a better choice. The remaining experiments used this approach.

Hyper-parameters. In this set of experiments, the influence of the parameters as well as the justification for the chosen values are given. There are three parameters to be considered in our approach (see Section 3.2 for their definitions): (i) the value K in the adaptive thresholding process, (ii) the dimensionality of the projected features p , and (iii) the number of random projections P . The performance is expected to increase with P since more information of the original representation is used. We first set P to the arbitrary value of 20, and run grid-search to find K and p . As shown in Figure 3, to obtain good performance, K has to be set to about 10% of p . The performance is not sensitive to the dimensionality of the projected features p , and a large range of values can be used. Nevertheless, small values for p are more desirable since they lead to small values for K . Remind that the mining complexity grows exponentially with K . To see the influence of P , we next fix $K = 20$ and $p = 800$ and evaluate the results with different values of P .

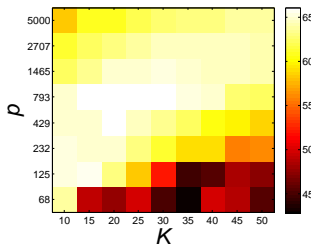


Figure 3: mAP as a function of p and K .

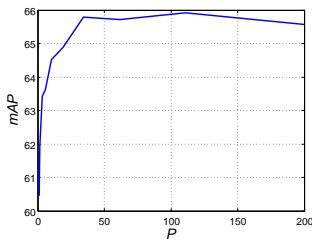


Figure 4: mAP as a function of P .

| Type | mAP | #Patterns |
|--------------------------|-------|-----------|
| Frequent pattern | 65.1 | 2M |
| Frequent closed pattern | 65.7 | 450k |
| Frequent maximal pattern | 54.5 | 87k |

Table 1: Mean Average Precision and number of patterns for each type of patterns.

As shown by Figure 4, the performance increases with the number of projections and saturated around $P = 50$.

Types of patterns. In this set of experiments, we investigate how the performance is related to the type of pattern, chosen as being either (i) frequent patterns, (ii) frequent closed patterns or (iii) maximal frequent patterns (see previous section for their definitions). As shown Table 1, using frequent *closed* patterns gives similar results than using frequent patterns, despite there is about 5 times less patterns. This was expected as frequent closed patterns carry the same information as frequent patterns. We also observe that the number of frequent *maximal* patterns is 5 times lower than the number of frequent closed patterns, but the performance is significantly worse. We do not report here run-times, but they are approximately proportional to the number of patterns that are produced.

Vocabulary size. In this experiment, different vocabulary sizes are investigated. As shown by Table 2, the performance increases with the size of the vocabularies. However, increasing the size of the vocabulary also increases the encoding time so that we limited the size to 2,000 visual words.

Use of the original ranking. Our proposed scoring function taking into account the original ranking order as introduced in Section 3.1 (Eq. 3), improving the performance of the re-ranking algorithm (Eq. 2) by 2.4% mAP .

Best P random projections. As described in Section 3.2.2, we evaluated the performance given by the use of 5,000 random projections, and kept the best few projections which perform best. We made 3-fold cross-validation on the validation data and found that the performance saturates after adding the 20 best random projections. The improvement is of 2% mAP over using all the projections. Consequently, this feature selection not only makes the process faster but also improves performance. For the remaining experiments, we selected the best 20 (out of 5,000) random

| Vocab. size | 100 | 1,000 | 2,000 |
|-------------|------|-------|-------|
| mAP | 62.9 | 65.7 | 66.4 |

Table 2: Performance for different vocabulary sizes.

projections on the validation data and use them on the test data.

4.3 Complexity and computational time

Regardless of the time for computing the BoW representation – which can be done off-line and is the same for any re-ranking approach – the complexity of pattern extraction is linear according to the number of images. In addition, the computational time for scoring an image is linear with the number of patterns. The computational time for pattern extractions and scoring the images are given Table 3. As shown by Table 3, the approach is fast enough for being used on the fly. Moreover, the computation time is stable across different queries. On average, it takes about 0.15 seconds, which is comparable to the computational times reported by [24].

| Query | Pat. Extract(s) | | Scoring(s) | | #Pat. | |
|------------------------|-----------------|-------------|-------------|-------------|------------|------------|
| | <i>Tr.</i> | <i>Rp.</i> | <i>Tr.</i> | <i>Rp.</i> | <i>Tr.</i> | <i>Rp.</i> |
| Maradona | 0.18 | 0.10 | 0.01 | 0.03 | 2k | 7k |
| Giraffe | 0.30 | 0.10 | 0.02 | 0.04 | 4k | 8k |
| Times square | 0.49 | 0.14 | 0.03 | 0.07 | 7k | 14k |
| Grand canyon | 0.81 | 0.12 | 0.05 | 0.06 | 10k | 11k |
| Logo Chelsea | 3.53 | 0.13 | 0.40 | 0.06 | 42k | 11k |
| Map World | 8.62 | 0.13 | 1.03 | 0.07 | 100k | 12k |
| <i>Mean 30 queries</i> | <i>1.58</i> | <i>0.11</i> | <i>0.17</i> | <i>0.04</i> | <i>17k</i> | <i>9k</i> |
| <i>Std. 30 queries</i> | <i>6</i> | <i>0.02</i> | <i>0.02</i> | <i>0.02</i> | <i>30k</i> | <i>3k</i> |

Table 3: Computational time (in sec.) for pattern extraction and image scoring), as well as number of patterns. Values are given both for the method using transposed data (*Tr.*) and the one using multiple random projections (*Rp.*). Computed on a validation set.

4.4 Performance evaluation and comparison with other methods

In this section, we compare our re-ranking method with state-of-the-art re-ranking methods available in the literature. In these experiments, three hyper-parameters are set as $P = 20$, $p = 800$, $K = 20$. The size of vocabularies is of 2,000. The original ranking is integrated excepted for the eBay Motorbike dataset for which the initial positions of the images are meaningless. The selection of the best random projections is activated. Note that we have cross-validated the settings on all datasets, and, interestingly, the choice of the hyper-parameters is very stable. Moreover, even the 20 best random projections apply well to all datasets. It explains why we eventually used the same settings for the 3 datasets.

INRIA Web Queries dataset. On this dataset, as shown by Table 4, our re-ranking approach improves the original search engine ranking by about 13% mAP and is better than any classifier-based approaches [15, 24]. Comparing with the graph ranking of [17], our approach has comparable performance in terms of mAP . However as explained in Section 2, our approach is real-time while the graph based ranking approach has a significantly higher complexity and cannot be used on the fly.

We can observe Figure 5 that after applying our re-ranking the top results are very clean compare to the original ranking, especially for queries in which images have small vari-

| Method | $mAP(\%)$ |
|----------------------------|-------------|
| Original Search Engine | 56.9 |
| Query-ind.+Query-dep. [24] | 65.5 |
| LogReg (visual) [15] | 64.9 |
| SpecFilter+MRank [17] | 73.8 |
| Ours | 72.2 |

Table 4: Comparison to other re-ranking approaches on the INRIA Web Queries dataset.

| Method | $mAP(\%)$ |
|----------------------------|-------------|
| Original ranking | 70.4 |
| Query-ind.+Query-dep. [24] | 72.7 |
| Ours | 76.1 |

Table 5: Comparison to other existing re-ranking approaches on the QUAERO’s visual concepts image dataset.

| Method | $EER(\%)$ |
|--------------------------|-------------|
| Implicit Shape Model [6] | 71.0 |
| FP+SVM [18] | 72.6 |
| Ours | 80.0 |

Table 6: Comparison to other existing re-ranking approaches on the Ebay Motorbike dataset.

ations *e.g.* ‘logos’, ‘maps’, and ‘flags’ up to immediate variations *e.g.* ‘landmarks’, ‘celebrities’. For queries in which images have a large range of diversity *e.g.* ‘Generic objects’, ‘Animals’, the top images are less clean. Nevertheless, our re-ranking approach can still improve the original ranking from the text-based search.

QUAERO’s visual concepts image dataset. In this dataset, as shown by Table 4, our re-ranking approach improves the original search engine ranking by about 6% mAP , and is 3% mAP better than the best result reported on this dataset [24].

eBay Motorbike dataset. As shown by Table 6 and Figure 6, our approach is 7.4% EER better than the state-of-the-art approach of [18].

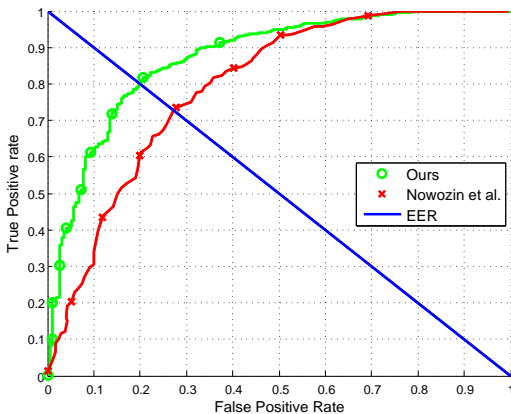


Figure 6: ROC curves comparing our re-ranking system with [18] on the Ebay dataset.

5. CONCLUSIONS

This paper proposes a new approach for image re-ranking relying on the effective use of pattern mining techniques. A new scoring function for re-ranking images in the context of text-based image retrieval is defined. Relying on the hypothesis that non-relevant images are more scattered than relevant ones, the proposed scoring function updates the original scores by measuring the amount of frequent patterns contained in images. In addition of being fast enough for on the fly usage, the approach gives state-of-the-art results on three different challenging datasets.

Acknowledgments

This work was partially funded by the QUAERO project supported by OSEO, French State agency for innovation.

6. REFERENCES

- [1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.
- [2] R. J. Bayardo Jr. Efficiently mining long patterns from databases. In *ACM Sigmod Record*, volume 27. ACM, 1998.
- [3] N. Ben-Haim, B. Babenko, and S. Belongie. Improving web-based image search via content based clustering. In *CVPR Workshop*, 2006.
- [4] T. Berg and D. Forsyth. Animals on the web. In *CVPR*, 2006.
- [5] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, 2004.
- [6] M. Fritz and B. Schiele. Towards unsupervised discovery of visual categories. In *DAGM*, 2006.
- [7] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *CVPR*, 2008.
- [8] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB, 1999.
- [9] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *PAMI*, 30:1371–1384, 2008.
- [10] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.
- [11] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Reranking methods for visual search. *Multimedia, IEEE*, 14:14–22, 2007.
- [12] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE PAMI*, 34(9):1704–1716, 2012.
- [13] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *PAMI*, 30:1877–1890, 2008.
- [14] L. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *WWW*. ACM, 2008.
- [15] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web-image search results using query-relative classifiers. In *CVPR*, 2010.

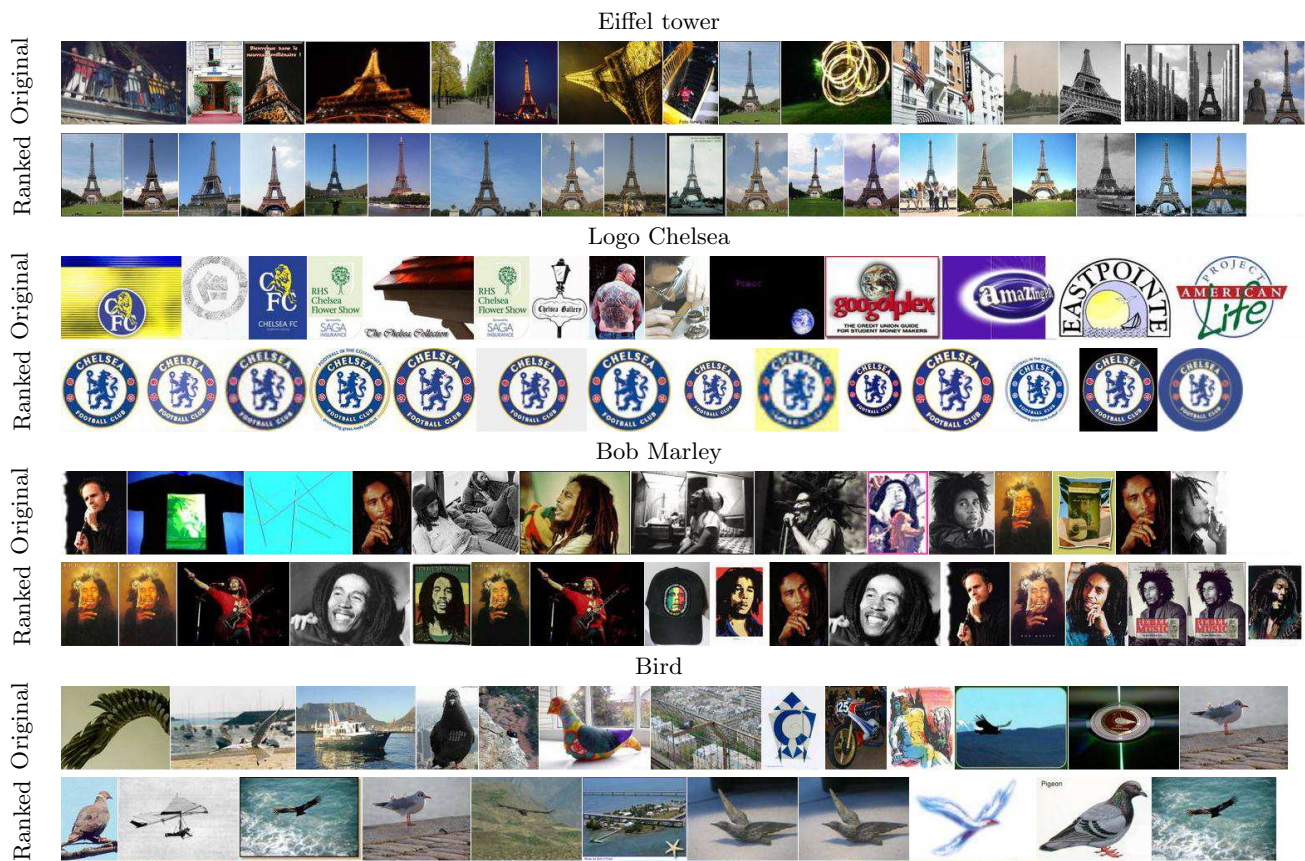


Figure 5: Qualitative re-ranking results.

- [16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
- [17] W. Liu, Y. Jiang, J. Luo, and S. Chang. Noise resistant graph ranking for improved web image search. In *CVPR*, 2011.
- [18] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. BakIr. Weighted substructure mining for image analysis. In *CVPR*, 2007.
- [19] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *ICDT*. 1999.
- [20] F. Rioult, J.-F. Boulicaut, B. Crémilleux, and J. Besson. Using transposition for pattern discovery from microarray data. In *8th ACM SIGMOD Workshop in DMKD*, 2003.
- [21] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.
- [22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.
- [23] Y. Su and F. Jurie. Visual word disambiguation by semantic contexts. In *ICCV*, 2011.
- [24] F. Thollard and G. Quénot. Content-based re-ranking of text-based image search results. In *ECIR*, 2013.
- [25] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *DS*, 2004.
- [26] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [27] W. Voravuthikunchai, B. Crémilleux, and F. Jurie. Histograms of pattern sets for image classification and object recognition. In *CVPR*, 2014.
- [28] C. Wallraven, B. Caputo, and A. B. A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, 2003.
- [29] J. Wang, Y.-G. Jiang, and S.-F. Chang. Label diagnosis through self tuning for web image search. In *CVPR*, 2009.