Learning to Rank Bag-of-Word Histograms for Large-scale Object Retrieval

Danfeng Qin
http://www.vision.ee.ethz.ch/~qind/
Yuhua Chen
yuhchen@ee.ethz.ch
Matthieu Guillaumin
http://www.vision.ee.ethz.ch/~mguillau/
Luc Van Gool
http://www.vision.ee.ethz.ch/~vangool/

Computer Vision Laboratory
ETH Zurich
Switzerland

Retrieving images of a particular query object in a large database of images is an important problem for computer vision with applications in object discovery, 3D reconstruction, location recognition and mobile visual search. Most recent state-of-the-art large-scale image retrieval systems rely on local features, in particular the SIFT descriptor and its variants. Typically, those local descriptors are aggregated into a histogram-based representation of the image referred to as the Bag-of-Words model (BoW) [4]. BoW models considerably reduce the computational burden and the memory footprint of the systems, because local descriptors are quantised into *visual words*.

For BoW histograms, it is common to use simple similarity functions such as the inner product or cosine similarity. However, such functions are not optimal for modelling the visual similarity between BoW features and thus lead to sub-optimal performance for retrieval [2, 3, 6]. The potential problems are the following: a) The evidence coming from co-missing visual features is under-estimated [2]; b) The similarity between a query image and a database image should not be symmetric [6]; c) Statistical properties of visual words are not taken into account [1, 3, 5].

Even though different methods have been proposed to address each of these problems individually, none provides a satisfying solution to properly account for all of them. Moreover, most authors propose ad-hoc solutions by means of functions controlled by very few parameters. These parameters are then hand-tuned or exhaustively searched on validation/test data to adapt them to each dataset. In this work, our goal is to replace those ad-hoc similarities in measuring histograms with ones that are specifically trained to maximize the retrieval accuracy. We propose to use a simple and very general linear model whose weights directly represent the similarity values. We devise a variant of rank-SVM to learn those weights automatically from training data with fast convergence and we propose techniques to limit the weights to a tractable number to avoid overfitting. Importantly, the flexibility of our model allows us to seamlessly incorporate well-known image retrieval schemes such as burstiness, negative evidence and idf weighting, and still exploit inverted files for efficiency in the large-scale setting. In our experiments, as shown in Table. 1, our approach consistently and significantly outperforms the similarities used in several state-of-the-art systems on 4 standard benchmark datasets.

Most of existing similarity measures [2, 3, 6] can be written in a very general form as:

$$s(q,d) = \tau(q)\tau(d)\sum_{i=1}^{K} s_i(q_i,d_i). \tag{1}$$

Rather than trying to design τ and s_i manually, we propose to resort to learning and discover the patterns of a good similarity function for image search, automatically from training data. Looking at Eq. (1), we aim at learning the values $s_i(q_i,d_i)$ directly. This is notably impractical, as each q_i and d_i can be arbitrarily large. However, state-of-the-art methods use very large visual codebook ($K \approx 10^6$) leading to sparse of BoW representations, with few occurrences of any visual word in any given image. As a result, using a *truncated histogram* $\hat{q}_i = \min(q_i, n)$ with $n \in \mathbb{N}^+$ will provide an excellent approximation of the original histogram while limiting the number of possible values of $s_i(\hat{q}_i,\hat{d}_i)$ to $(n+1)^2$. Additionally, because we learn the values of $s_i(\hat{q}_i,\hat{d}_i)$ directly, these terms can be learned to incorporate a *contribution to the normalisation functions*. This leads to a modified similarity $\hat{s}_i$ and our approximated model becomes additive and writes as:

$$s(q,d) = \tau(q)\tau(d)\sum_{i=1}^{K} s_i(q_i,d_i) \approx \sum_{i=1}^{K} \hat{s}_i(\hat{q}_i,\hat{d}_i), \tag{2}$$

where $\hat{s}_i(j,l)$ for $j,l \in [0,n]$ are the $K\cdot(n+1)^2$ parameters to learn. Notably, this additive approximation allows to rewrite Eq. (2) as a linear combination of indicator functions:

$$\hat{s}_i(\hat{q}_i,\hat{d}_i) = w_{i\hat{q}_i\hat{d}_i} = \sum_{j=0}^{n}\sum_{l=0}^{n} w_{ijl}\mathbb{I}(\hat{q}_i = j)\mathbb{I}(\hat{d}_i = l), \tag{3}$$

where $w_{ijl} = \hat{s}_i(j,l)$. In other words, if we define $\Psi(q,d)$ as the binary vector indexed by (i,j,l) such that $\Psi_{ijl}(q,d) = \mathbb{I}(\hat{q}_i = j)\mathbb{I}(\hat{d}_i = l)$ and define $\mathbf{w} = [w_{ijl}]_{i,j,l}$, then:

$$s(q,d) \approx \mathbf{w}^\top \Psi(q,d). \tag{4}$$

Importantly, Eq. (4) highlights that Ψ acts as a feature encoding for the query-document pair (q,d) in a linear prediction model. Despite its simplicity, this model is very general and flexible, and is able to incorporate many of the properties discussed in [2, 3, 6], and potentially others, without having to explicitly model them. To illustrate this, let us first consider the simple case of $n = 1$. In such case, the truncated histogram $\hat{q}$ simply encodes the absence or presence of visual words (an encoding often referred to as *binary bag-of-words*), and there are only 4 weights to learn per visual word: co-absence $\hat{s}_i(0,0)$, co-occurrence $\hat{s}_i(1,1)$ and either case of mutual exclusion $\hat{s}_i(0,1)$ and $\hat{s}_i(1,0)$. If we learn that $\hat{s}_i(0,0) > \hat{s}_i(0,1)$, then not only have we implicitly learned that co-absence of the visual word i contribute more to the similarity than mutual exclusion (as argued by [2]) but also exactly by which amount. If we learn that $\hat{s}_i(0,1) \neq \hat{s}_i(1,0)$, then this implies that the ideal similarity is indeed asymmetric [6]. Finally, learning all the weights together allows to identify which visual words are more important than others, as indicated by the relative weight of $\hat{s}_i(1,1)$ and $\hat{s}_j(1,1)$. Hence, it automatically models re-weighting schemes such as IDF. Finally, when $n > 1$, phenomena such as burstiness [3] are also learnt.

	Oxford5k^s	Oxford105k^s	Holidays^s	UKbench^s
Cosine Similarity	0.819 (0)	0.725 (0)	0.862 (0)	3.51 (0)
Burstiness Weighting [3]	0.826 (0)	0.748 (0)	0.858 (0)	3.54 (0)
Negative Evidence [2]	0.830 (0)	0.684 (0)	0.848 (0)	3.44 (0)
Adaptive Asymmetric [6]	0.839 (1)	0.758 (0)	0.795 (0)	3.38 (0)
This paper	**0.870 (9)**	**0.816 (10)**	**0.871 (10)**	**3.70 (10)**

Table 1: Comparison to alternative similarities. We report the average performance over the 10 splits of the data (mAP or top-4 score depending on the dataset) and in parenthesis the number of runs where the method is the best. In bold is the best result for each dataset.

[1] Ondrej Chum, James Philbin, and Andrew Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008.

[2] Hervé Jégou and Ondřej Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *Computer Vision–ECCV 2012*, pages 774–787. Springer, 2012.

[3] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1169–1176. IEEE, 2009.

[4] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[5] Liang Zheng, Shengjin Wang, Ziqiong Liu, and Qi Tian. Lp-norm idf for large scale image search. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1626–1633. IEEE, 2013.

[6] Cai-Zhi Zhu, Hervé Jégou, and Shin'ichi Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV-International Conference on Computer Vision*, 2013.