Unsupervised Multi-Feature Tag Relevance Learning for Social Image Retrieval

Xirong Li, Cees G.M. Snoek, and Marcel Worring
Intelligent Systems Lab Amsterdam, University of Amsterdam
Science Park 107, 1098XG, Amsterdam, The Netherlands
{x.li, cgmsnoek, m.worring}@uva.nl

ABSTRACT

Interpreting the relevance of a user-contributed tag with respect to the visual content of an image is an emerging problem in social image retrieval. In the literature this problem is tackled by analyzing the correlation between tags and images represented by specific visual features. Unfortunately, no single feature represents the visual content completely, e.g., global features are suitable for capturing the gist of scenes, while local features are better for depicting objects. To solve the problem of learning tag relevance given multiple features, we introduce in this paper two simple and effective methods: one is based on the classical Borda Count and the other is a method we name UniformTagger. Both methods combine the output of many tag relevance learners driven by diverse features in an unsupervised, rather than supervised, manner.

Experiments on 3.5 million social-tagged images and two test sets verify our proposal. Using learned tag relevance as updated tag frequency for social image retrieval, both Borda Count and UniformTagger outperform retrieval without tag relevance learning and retrieval with single-feature tag relevance learning. Moreover, the two unsupervised methods are comparable to a state-of-the-art supervised alternative, but without the need of any training data.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.2.4 [**Database Management**]: Systems—*Multimedia databases*

General Terms

Algorithms, Design, Experimentation

Keywords

Tag relevance learning, multi-feature, social image retrieval

CIVR '10, July 5-7, Xi'an, China

1. INTRODUCTION

Image sharing websites such as Flickr and Facebook are hosting billions of personal photos. Social image tagging, assigning tags to images by common users, is reshaping the way people manage and access such large-scale visual content. One might expect tag-based retrieval to be a natural and good starting point for search. Compared to content-based image retrieval [6], tag-based retrieval copes more easily with semantic queries. Moreover, its scalability has been verified by text retrieval research [2]. However, due to the diversity of knowledge and cultural background of its users, social tagging is often subjective and inaccurate. We consider a tag objective and relevant with respect to an image if the tag accurately describes objective aspects of the visual content. In other words, users with common knowledge relate the tag to the visual content easily and consistently. As a consequence, objective tags reflect visual concepts such as objects, scenes, and events. In contrast to free text descriptions, wherein tag relevance might be reflected by tag statistics [2], individual tags are used once per image in the social tagging paradigm. Hence, a fundamental problem for social image retrieval is how to objectively learn the relevance of a tag with respect to the visual content it is describing.

Recently, several papers have appeared in the literature to tackle the tag relevance learning problem [17–19]. In [17], for instance, we proposed a neighbor voting algorithm which estimates a tag's relevance by exploiting tagging redundancies among multiple users. The key idea is that if different persons label visually similar images using the same tags, these tags are likely to be relevant. Starting from the neighbor voting results, the authors in [19] further exploit pairwise similarity between tags to reinforce relevant tags. In general, a single feature is used to define visual (dis)similarity between images. Unfortunately, no single feature can represent the visual content completely [20], e.g., global features are suitable for capturing the gist of scenes [21], while local features are better for depicting objects [3]. Representing images by multiple features of different types might be beneficial as shown in previous studies, e.g., [24] for content-based image retrieval and [5, 8] for visual categorization. In [5, 8, 24], a considerable amount of training data is demanded to learn an optimized combination strategy per concept. Given the potentially unlimited array of query concepts in social image retrieval, an unsupervised or lightweight method which effectively and efficiently exploits diverse features for tag relevance learning is required.

In this paper, we introduce two such simple and effective solutions for multi-feature tag relevance learning: one

solution is based on the classical Borda Count [1] and the other is our proposed UniformTagger. Using a neighbor voting algorithm as a base learner [18], both methods combine the output of multiple tag relevance learners in a generic and unsupervised way. We evaluate the viability of the two methods on 3.5 million social-tagged images and two benchmark sets.

2. RELATED WORK

According to their query-dependence, we divide related work for social image retrieval into two types of methods: query-dependent methods and query-independent methods.

2.1 Query-dependent Methods

Given unsatisfactory image search results caused by subjective social tagging, query-dependent methods aim to improve image retrieval, either by re-ranking search results in terms of their visual consistency [10, 11, 14] or by aggregating search results returned by multiple sources of textual descriptions [22]. Re-ranking methods, for instance, assume the majority of top n, typically 1000, search results are relevant with respect to the query and relevant examples tend to have similar visual patterns such as color, texture, and shape. To find the dominant visual patterns, density estimation is often used, typically in the form of clustering [11] or a random walk on a graph wherein each node is a result image and each edge is weighted by pairwise visual (dis)similarity [14]. Density estimation tends to be inaccurate when feature dimensionality is high and samples are insufficient for computing the density [23], both of which often happen in the re-ranking scenario. Besides, density estimation is computationally expensive. Pre-computing search results is possible for some common queries. It is challenging to cover diverse user queries. Since social-tagged images might have extra meta-data such as notes and comments from users, some seek to improve retrieval accuracy by aggregating image search results ranked by retrieval systems built on individual meta-data. For example, the authors in [22] use Borda Count, a rank aggregation strategy for meta-search in text retrieval [1], to aggregate top image search results obtained with a set of user-provided notes. In summary, the fundamental problem of subjective social tagging is unaddressed in query-dependent methods.

2.2 Query-independent Methods

Query-independent methods target at improving social tagging accuracy by predicting objective tags which reflect visual concepts visible in images. As a consequence of more accurate tagging, better image search results might be achieved. Predicting relevant tags for unlabeled images, or image auto-tagging, has been intensively studied in the last decade [8, 16]. Learning tag relevance for social-tagged images is however relatively new, and distinguishable from automatic image tagging in the following two aspects: 1) a small number of candidate tags to predict for individual images, and 2) a large number of concepts to model for a collection. First, given a social-tagged image, relevant tags are identified only from the associated tags. It is thus "easier" than tagging unlabeled images. Second, given the diversity of social tagging and numerous loosely tagged visual data, the number of concepts to be modeled is larger than the relatively small number of predefined concepts in a typical image auto-tagging scenario [16]. Therefore, a simple and efficient solution to social tag relevance learning is desirable.

As the first attempt to learn social tag relevance, we proposed a neighbor voting algorithm which infers the relevance of a tag with respect to an image from the tagging behavior of its visual neighbors [17]. By updating tag frequency using the learned tag relevance value in a tag-based retrieval paradigm, better search results are obtained, while simultaneously scalability is maintained. Despite its simplicity, the neighbor voting algorithm is quite effective as confirmed by several other papers, e.g., [15, 19]. In [19], the authors base a tag ranking method on the neighbor voting results. The authors in [15] find that objective tags can be identified by visual neighbor voting. In [26,27], the authors combine both tag similarity and image similarity. In the literature, however, only a single feature is used for tag relevance learning. How to leverage multiple features in the existing frameworks remains unclear, which will be the focus of this paper.

3. MULTI-FEATURE TAG RELEVANCE

We first introduce some notation. Let x be an image and $F = \{f_1, \ldots, f_m\}$ a set of feature extraction functions yielding visual features for image representation. Given a large collection of N social-tagged images and query concept w, let $D_w = \{x_1, \ldots, x_{N_w}\}$ be images in the collection labeled with w, where D_{w+} and D_{w-} represent images relevant and irrelevant to w, respectively. For $f_i \in F$, we define a tag relevance measure $g_{i,j}(f_i(x), \theta_j, w) \in [0,1]$, where $\{\theta_j | j = 1, \ldots, n\}$ is a set of model parameters.

We aim to derive a ranking function $G(x, w) \in [0, 1]$ which ranks any relevant images ahead of any irrelevant images. To make this operational, we formulate the goal as finding the G maximizing the objective function of RankBoost [7],

$$r(G) = \sum_{x \in D_{w+}} \sum_{x' \in D_{w-}} \lambda(x, x') \left(G(x, w) - G(x', w) \right), \quad (1)$$

where $\{\lambda(x, x') | \sum \lambda(x, x') = 1\}$ are non-negative weights indicating the importance of correctly ranking the pair (x, x'). Intuitively, $r(G)$ prefers G which generates correct ranking with high confidence. As G is supposed to be better than random guess, we have $r(G) \in (0,1]$. Since linear combination or additive modeling has proven to be a solid choice for combining classification/ranking models [7, 9], we adopt the choice for combining multi-feature tag relevance learning. In particular, we seek a convex combination,

$$G(x, w) := \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_{i,j} \cdot g_{i,j}(f_i(x), \theta_j, w), \quad (2)$$

to maximize $r(G)$, where $\{\alpha_{i,j} | \sum_{i=1}^{m} \alpha_{i,j} = 1, \alpha_{i,j} \geq 0\}$ are weighting parameters indicate the importance of individual tag relevance learners. For convenience, we abbreviate $g_{i,j}(f_i(x), \theta_j, w)$ to $g_{i,j}(x, w)$ hereafter.

3.1 Single-feature Tag Relevance Learning

Before surmounting the multi-feature problem, we first introduce a single-feature tag relevance learner. Since learning tag relevance by neighbor voting is shown to be effective by previous studies [15,17–19], we follow this idea to derive the single-feature learners. We employ a neighbor voting algorithm which estimates the relevance of a tag with respect to an image by taking into account both the tag's frequency in the image's visual neighborhood and the tag's prior frequency [18]. Concretely, given the visual feature extracted

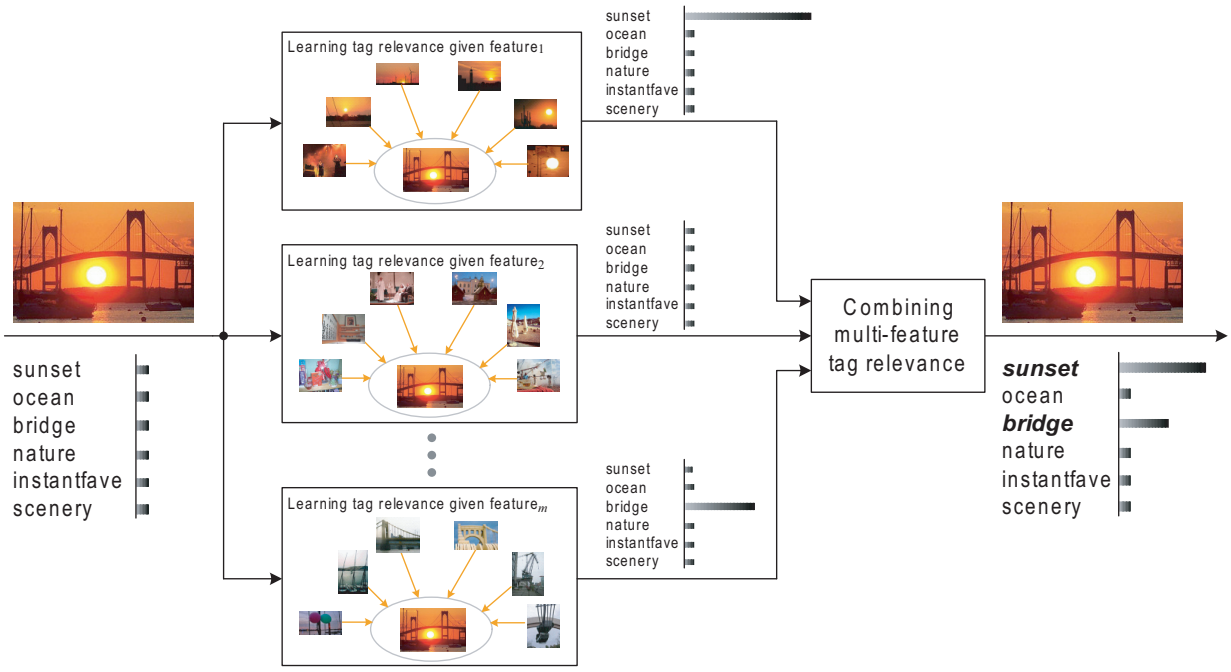


Figure 1: Multi-feature tag relevance learning. Using a neighbor voting algorithm as a single-feature base learner [18], we propose to improve tag relevance learning by combining the output of many base learners obtained with different features and model parameters.

by $f_i \in F$ and some distance metric, let $k_{j,w}$ be the number of images labeled with w in the k_j-nearest neighbor set of image x found in the collection. The tag relevance measure $g_{i,j}(x, w)$, normalized by dividing by k_j, is

$$max(\varepsilon, \frac{k_{j,w}}{k_j} - \frac{N_w}{N}), \tag{3}$$

where ε is a very small positive constant. We use the *max* function to reject unreliable estimates. As shown in Eq. 3, the more neighbor images labeled with the tag, the larger the tag relevance value will be, meanwhile high-frequency tags are penalized for their high prior. We employ the popular Euclidean distance throughout this work. With such simplification, the model parameters $\{\theta_j\}$ correspond to the number of neighbors $\{k_j\}$.

3.2 Combining Multi-feature Tag Relevance

Varying both the features and the model parameters, we will obtain multiple tag relevance estimates. If these estimates complement each other, combining them yields a better result, as illustrated in Fig. 1. To be precise, we aim for a combination method which maximizes the performance of image retrieval using the learned tag relevance value as a ranking criterion. We now elucidate how to perform such combination in an unsupervised manner without any training data and in a supervised manner with training data.

3.2.1 Unsupervised Combination Methods

As we target at the potentially unlimited array of concepts existing in social tagging where well-labeled training data are likely unavailable, we seek a generic and unsupervised combination algorithm. Since we have no prior knowledge of which tag relevance learners are most appropriate for a given concept, we propose to combine many base learners with different features and model parameters in a uniform manner, which we term UniformTagger. The rationale for this simple strategy is as follows. Envisage that the parameters $\{\alpha_{i,j}\}$ follow certain, yet unknown, probability distributions. According to the principle of maximum entropy [13], a postulate in probability theory, when no information is given about the distribution, the best, or safest, choice is the one with largest entropy, or in other words, the uniform distribution. Consequently, we have the ranking function $G(x, w)$ of UniformTagger as

$$\frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} g_{i,j}(x, w). \tag{4}$$

As an alternative to UniformTagger, we consider Borda Count, a well-known rank aggregation algorithm [1], to combine image search results ranked by individual tag relevance learners. Different from UniformTagger, Borda Count quantizes continuous tag relevance values into discrete ranks. The ranking function $G(x, w)$ of Borda Count is expressed as

$$\frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(N_w - rank(g_{i,j}(x, w))\right), \tag{5}$$

where $rank(g_{i,j}) \in \{1, ..., N_w\}$ returns the rank of image x after ranking the image set D_w in descending order according to the base learner $g_{i,j}$. The constant N_w is the number of images in D_w, as defined earlier.

3.2.2 Supervised Combination Methods

When well-labeled training data of a given concept are available, the weighting parameters $\{\alpha_{i,j}\}$ in Eq. 2 can be optimized to fit that concept. To this end, we consider three

Table 1: (Un)Supervised methods to combine multi-feature tag relevance learning results for social image retrieval. These methods implement the component "Combining multi-feature tag relevance" in Fig. 1. We shorten $g_{i,j}(x,w)$ to $g_{i,j}$ for better view of the table.

	Method	Input	Weights $\{\alpha_{i,j}\}$	Ranking function $G(x,w)$
Supervised	Best Single Learner [18]	$\{g_{i,j}\}$	$i^*,j^* = \underset{i,j}{\operatorname{argmax}}\, r(g_{i,j}),\ \alpha_{i^*,j^*}=1$	g_{i^*,j^*}
	Weighted Borda Count [1]	$\{rank(g_{i,j})\}$	$\frac{1}{2}\ln\left(\frac{1+r(g_{i,j})}{1-r(g_{i,j})}\right)$	$\sum_{i=1}^{m}\sum_{j=1}^{n}\alpha_{i,j}\cdot(N_w - rank(g_{i,j}))$
	RankBoost [7]	$\{g_{i,j}\}$	$\frac{1}{2}\ln\left(\frac{1+r(g_{i,j})}{1-r(g_{i,j})}\right)$	$\sum_{i=1}^{m}\sum_{j=1}^{n}\alpha_{i,j}\cdot g_{i,j}$
Unsupervised	Borda Count [1]	$\{rank(g_{i,j})\}$	$\frac{1}{m\cdot n}$	$\frac{1}{m\cdot n}\sum_{i=1}^{m}\sum_{j=1}^{n}(N_w - rank(g_{i,j}))$
	proposed UniformTagger	$\{g_{i,j}\}$	$\frac{1}{m\cdot n}$	$\frac{1}{m\cdot n}\sum_{i=1}^{m}\sum_{j=1}^{n} g_{i,j}$

state-of-the-art supervised methods, i.e., Best Single Learner [18], Weighted Borda Count [1], and RankBoost [7].

The Best Single Learner method selects as the ranking function an individual learner g_{i^*,j^*} maximizing the objective function $r(\cdot)$ in Eq. 1. In contrast to selecting one learner, Weighted Borda Count aggregates multiple ranked lists obtained with $\{g_{i,j}\}$, wherein each list is weighted according to the training performance of the corresponding $g_{i,j}$. RankBoost sequentially combines the output of individual learners with an adaptive weighting scheme to emphasize base learners capable of correcting mis-ranking made in previous learning rounds. To be precise, if a pair of positive example x and negative example x' is mis-ranked in the current round, $\lambda(x,x')$ in Eq. 1 increases so that an unused learner correctly ranking (x,x') will be selected in the next round. By contrast, all pairs in Best Single Learner and Weighted Borda Count are equally important, namely $\{\lambda(x,x') = \frac{1}{|D_{w+}|\cdot|D_{w-}|}\}$. For the three methods, a weighting function is required to convert the training performance into the weights. To make a fair comparison, we compute $\alpha_{i,j}$ using the weighting function of RankBoost,

$$\frac{1}{2}\ln\left(\frac{1+r(g_{i,j}(x,w))}{1-r(g_{i,j}(x,w))}\right). \quad (6)$$

Since good tag relevance learners generate high $r(\cdot)$, they receive large weights when combined.

We summarize both unsupervised and supervised methods in Table 1. Note that RankBoost and (Weighted) Borda Count have to maintain $m\times n$ copies of tag indexing data structures in memory to fuse search results during query time. In contrast, since UniformTagger converts multiple tag relevance estimates into a single value before search, it has to keep only one copy of the indexing structure in memory. Hence, the proposed UniformTagger requires a smaller memory footprint and is more efficient for image retrieval.

4. EXPERIMENTAL SETUP

For multi-feature tag relevance learning, there are two questions to answer: 1) is multi-feature better than single-feature, and 2) are the unsupervised methods competitive to the supervised alternatives? We answer these two questions in the context of tag-based social image retrieval.

4.1 Data Collections

A large social-tagged image set. For neighbor voting, we use 3.5 million social-tagged images randomly collected from Flickr in our earlier work [18].

Two evaluation sets. For evaluation, we use the following two sets: Social20[1] [18] and NUS-SCENE [4]. The Social20 set has 20 visual concepts and 19,972 images. Each concept has 1000 images labeled with that concept by social tagging. These images have been re-labeled in terms of their relevance, and evenly divided into the training data and the testing data. The NUS-SCENE set has 33 concepts covering a range of scenes. The number of training examples per concept ranges from 140 to 7,142, with an average value of 1,484. The statistics of the test set are similar.

4.2 Experiments

Experiment 1: Unsupervised image retrieval. We evaluate the two unsupervised methods: UniformTagger and Borda Count. For each concept, we obtain the Tag baseline by ranking images according to the concept's occurrence frequency in descending order. Then, for each image in the two test sets, we estimate the relevance of its associated tags by single-feature tag relevance learning with various settings. The multi-feature counterpart is obtained by combining the single-feature results using the two methods, respectively. To perform tag-based retrieval with tag relevance learning, we update tag frequency using the learned tag relevance values, and rank images in terms of the updated tag frequency. By doing so, we study how multi-feature tag relevance learning improves social image retrieval.

Experiment 2: Supervised image retrieval. For each concept, we train the three supervised combination methods, i.e., Best Single Learner, Weighted Borda Count, and RankBoost, on the training set. We then apply the trained models on the testing set.

Evaluation criteria. To assess image retrieval accuracy, we use Average Precision (AP), a common evaluation criterion in multimedia retrieval. To evaluate the overall performance, we use Mean Average Precision (MAP), the mean value of AP scores over all concepts.

[1]The Social20 set is available at `http://staff.science.uva.nl/~xirong/tagrel/`.

4.3 Implementation

Visual features. As an instantiation of feature extraction functions $\{f_1, \ldots, f_m\}$, we use three types of visual features: Color64, GIST, and Dense-SURF. The Color64 is a 64-d global feature combining the 44-d color correlogram [12], the 14-d texture moments [28], and the 6-d RGB color moments. The GIST is a 980-d global feature representing dominant spatial structure of a scene by a set of perceptual dimensions such as naturalness, openness, and roughness [21]. Finally, the Dense-SURF is a 4000-d bag-of-keypoints feature depicting local information of the visual content. We adopt dense sampling for keypoint detection and SURF [3] for keypoint description, using a fast implementation of the Dense-SURF [25]. For all features, we use the Euclidian distance to measure visual dissimilarity.

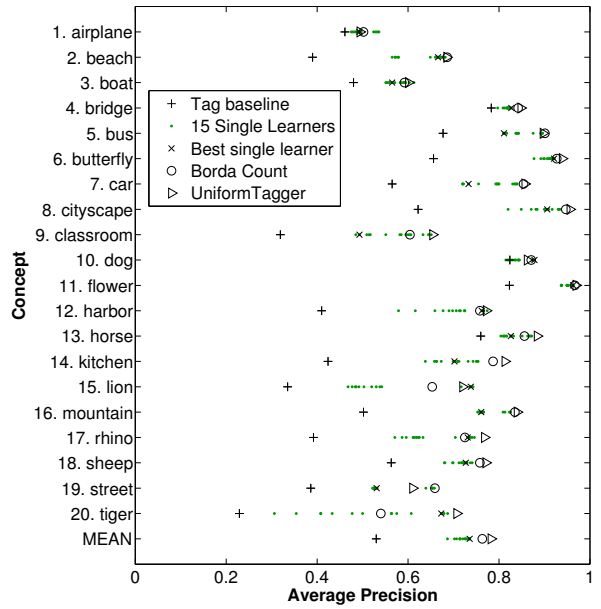
Parameters of base tag relevance learners. For each of the above three features, we vary the model parameter θ, i.e., the number of neighbors k in the neighbor voting algorithm, to create multiple tag relevance learners $g_{i,j}$ defined in Eq. 3. k We choose k from {500,1000, 1500,2000,2500}, and thus create 3×5=15 base learners in total.

Approximate visual neighbor search. To implement the neighbor voting algorithm, we perform approximate k-Nearest-Neighbor search on the 3.5 million collection as a tradeoff between accuracy and efficiency. We build a visual index for each feature by dividing the entire collection into smaller subsets by K-means clustering. Each subset is indexed by a cluster center and cached into main memory. For a query image, neighbor search is conducted within those subsets whose centers are closest to the query. For high-dimensional features such as the 4000-d Dense-SURF, the amount of computation and memory cost can still be considerable even in much reduced subsets. To tackle such difficulty, we employ Principal Component Analysis to reduce the original feature dimensionality to 30. To summarize, we first run K-means clustering on the original feature to create a coarse index of 1000 clusters. Neighbor search is then executed based on the reduced feature. The computational complexity of ranking cluster centers in terms of their distance to a query is $O(d \cdot K + K \cdot \log K)$, where d is the original feature dimensionality and K the number of cluster centers. Suppose we select p points from the closest clusters, the computational complexity for finding k nearest neighbors within the selected points is $O(\tilde{d} \cdot p + p \cdot \log k)$, where $\tilde{d}$ is the reduced feature dimensionality.

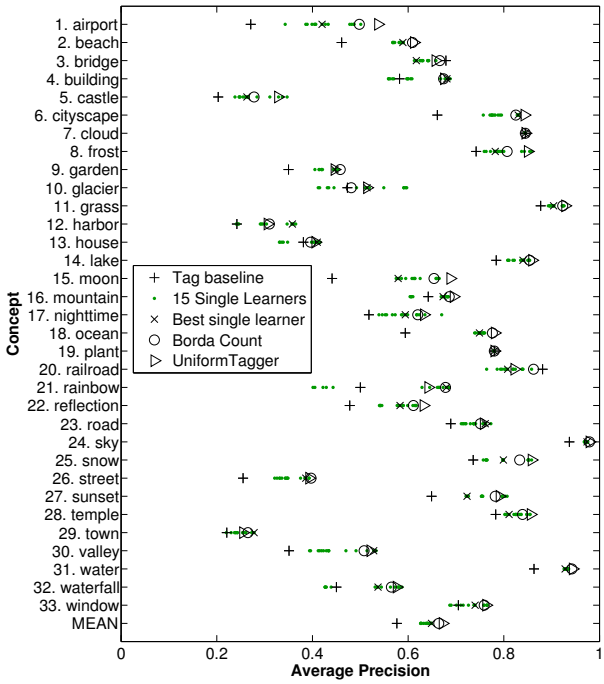
5. RESULTS

5.1 Experiment 1: Unsupervised image retrieval

We summarize the unsupervised retrieval results on Social20 and NUS-SCENE in Fig. 2(a) and Fig. 2(b). Both UniformTagger and Borda Count outperform the Tag Baseline and the single-feature runs. On Social20, for instance, UniformTagger reaches 47.6% and 8.6% improvements of MAP, compared to the Tag Baseline and the average result of the 15 single-feature runs, respectively. While the corresponding numbers on NUS-SCENE are 16.5% and 5.4%. For most of the individual concepts, UniformTagger is also superior to the average result of the Single-feature runs. As shown in Fig. 2(a), 14 out of the 20 concepts in the Social20 set have more than 5% improvement in terms of MAP. As shown in Fig. 2(b), 16 out of the 33 concepts in the NUS-SCENE set have more than 5% improvement in terms of MAP. Moreover, UniformTagger surpasses the best single-feature learner obtained by comparing all single-feature runs on the test sets. All these results verify the effectiveness of



(a)



(b)

Figure 2: Experiment 1. Comparing unsupervised multi-feature tag relevance learning with single-feature cases for tag-based social image retrieval. Multi-feature tag relevance learning outperforms the single-feature cases on the two test sets, namely (a) Social20 and (b) NUS-SCENE.

multi-feature tag relevance learning.

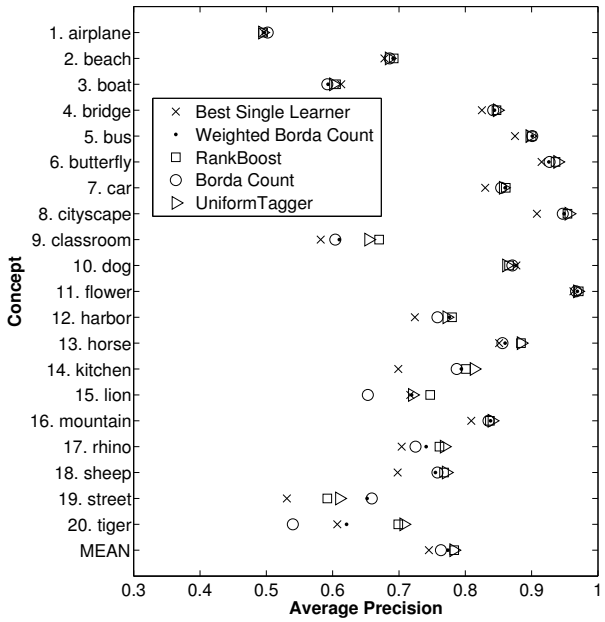
For the two unsupervised methods, UniformTagger is slightly better than Borda Count. Since Borda Count quantizes tag relevance values into ranks, it is more robust to outliers. Consider the concepts *street* and *tiger* in Social20 for example. The GIST feature is superior to the Color64 feature for *street*, while inferior for *tiger*. For both concepts, base learners using Color64 generate larger tag relevance values than using GIST. Canceling out larger yet inaccurate estimations by quantization, Borda Count obtains better results for *street*. While for *tiger*, by preserving the original tag relevance estimations, UniformTagger surpasses Borda Count. Therefore, only when there are some base learners producing large yet inaccurate estimates, Borda Count is preferred. Otherwise, UniformTagger is the best choice.

5.2 Experiment 2: Supervised image retrieval

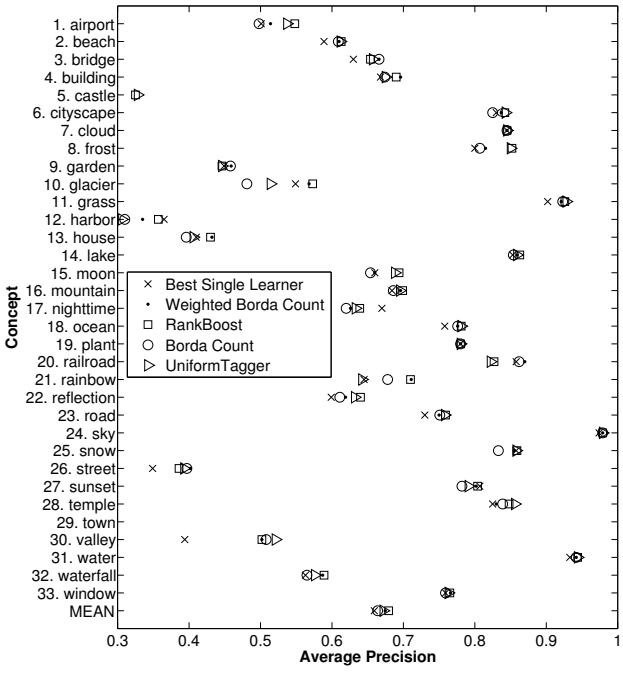
We show the supervised retrieval results on Social20 and NUS-SCENE in Fig. 3(a) and Fig. 3(b). Among the three supervised methods, RankBoost is the best, while Best Single Learner is inferior to the two competitors for the majority of the concepts. Since the best learner is determined by training, the divergence between the training and the testing data might result in a suboptimal option. In contrast the best single strategy, combining multiple tag relevance learning results tends to be more reliable and accurate.

Further, we compare UniformTagger, the best unsupervised method, with RankBoost. In general, UniformTagger, with an MAP of 0.782 on Social20 and 0.672 on NUS-SCENE, is on par with RankBoost, which has an MAP of 0.783 on Social20 and 0.679 on NUS-SCENE. For Social-20, the performance difference between the two methods for each concept is less than 5.0% in terms of AP. For NUS-SCENE, RankBoost outperforms UniformTagger for the following four concepts: *harbor* (16.8%), *glacier* (11.5%), *rainbow* (10.5%), and *house* (5.9%), where the numbers in the parentheses are relative improvements in terms of MAP. Looking into search results ranked by the individual base learners, we observe the common phenomenon that when one feature is significantly better than the other two features, e.g., GIST for *harbor* and *rainbow*, and Color64 for *glacier*, base learners with the best features are reinforced by training. Notice the relatively small improvement for concept *house*. We explain this result by the observation that both Color64 and GIST are suitable for *house* and form the majority even in UniformTagger. Note that the good performance of RankBoost and the other supervised methods is gained at the expense of acquiring a considerable amount of training data. Surprisingly, UniformTagger, as well as Borda Count, are comparable to these supervised alternatives, yet without resorting to any training effort.

Our explanation of such counter-intuitive results is as follows. Recall that RankBoost weights the base learners with respect to their training performance measured by the objective function r, which favors a ranker predicting good ranking with high confidence, as aforementioned. It is thus possible to over-emphasize a ranker which gives large tag relevance estimates yet suboptimal ranking. In such case, "optimized" weights indeed make the performance degenerate, as opposed to the uniform weights (see concept *street* in Social20 for instance). Though UniformTagger and Borda Count might not be the best option for some concepts, they are as effective as the supervised alternatives in general.



(a)



(b)

Figure 3: Experiment 2. Comparing unsupervised and supervised methods, on combining multi-feature tag relevance learning for tag-based social image retrieval on the two test sets, (a) Social20 and (b) NUS-SCENE. UniformTagger and Borda Count are comparable to the supervised alternatives, but without any training effort. The horizontal axis starts at 0.3 for a better view of small differences.

Finally, we present some image search results in Fig. 4. Search performances for concepts having strong visual clues,

e.g., *snow*, can be easily improved by single-feature tag relevance learning. For *airplane*, since users tend to label images of aerial views taken from airplane windows as *airplane*, learning tag relevance by neighbor voting does not yield much improvement. While for concepts having larger intra-concept visual diversity such as *kitchen* or concepts having larger inter-concept visual ambiguity such as *rainbow* versus colorful things like balloons, UniformTagger performs best by combining multi-feature tag relevance learning results.

6. CONCLUSIONS

Given subjective social tagging, how to objectively interpret the relevance of a user-contributed tag with respect to the visual content it is describing is an emerging problem in social image retrieval. In this paper, we investigate both unsupervised and supervised methods for multi-feature tag relevance learning. As a main contribution, we propose the UniformTagger method. Using a neighbor voting algorithm as the base single-feature learner, UniformTagger combines tag relevance estimations of many base learners in a uniform and unsupervised manner. Compared to an unsupervised alternative, namely Borda Count, and three supervised alternatives, i.e., Best Single Learner, Weighted Borda Count, and RankBoost, UniformTagger achieves better or comparable performance. Meanwhile, it requires a smaller memory footprint and is more efficient for tag-based image retrieval.

Experiments on 3.5 million social-tagged images and two realistic test sets verify the effectiveness of multi-feature tag relevance learning. Compared to a retrieval baseline without tag relevance learning, UniformTagger achieves a relative improvement of 47.6% and 16.5% in terms of MAP, on the two test sets. Compared to the average result of retrieval using single-feature tag relevance learning, UniformTagger gains an improvement of 8.6% and 5.4% in terms of MAP. Moreover, UniformTagger is comparable to RankBoost, a state-of-the-art supervised alternative, but without any training effort. In light of the current trend towards large-scale visual search, we consider the simplicity and the lack of supervision, coupled with the good performance, the most valuable assets of the proposed method.

Acknowledgements

This work was supported by the EC-FP6 VIDI-Video project and the STW SEARCHER project.

7. REFERENCES

[1] J. Aslam and M. Montague. Models for metasearch. In *SIGIR*, 2001.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[4] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. NUS-WIDE: A real-world web image database from National University of Singapore. In *CIVR*, 2009.

[5] M. Cooper. Image categorization combining neighborhood methods and boosting. In *ACM MM Workshop on LS-MMRM*, 2009.

[6] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Survey*, 40(2):1–60, 2008.

[7] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[8] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[10] E. Hörster, M. Slaney, M. Ranzato, and K. Weinberger. Unsupervised image ranking. In *ACM MM Workshop on LS-MMRM*, 2009.

[11] W. Hsu, L. Kennedy, and S.-F. Chang. Reranking methods for visual search. *IEEE MM*, 14(3):14–22, 2007.

[12] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR*, 1997.

[13] E. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[14] Y. Jing and S. Baluja. VisualRank: Applying PageRank to large-scale image search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1877–1890, 2008.

[15] L. Kennedy, M. Slaney, and K. Weinberger. Reliable tags using image similarity: mining specificity and expertise from large-scale multimedia databases. In *ACM MM Workshop on WSMC*, 2009.

[16] J. Li and J. Wang. Real-time computerized annotation of pictures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):985–1002, 2008.

[17] X. Li, C. Snoek, and M. Worring. Learning tag relevance by neighbor voting for social image retrieval. In *ACM MIR*, 2008.

[18] X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Trans. MM*, 11(7):1310–1322, 2009.

[19] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, 2009.

[20] Y. Lu, L. Zhang, Q. Tian, and W.-Y. Ma. What are the high-level concepts with small semantic gaps? In *CVPR*, 2008.

[21] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[22] X. Olivares, M. Ciaramita, and R. van Zwol. Boosting image retrieval through aggregating search results based on visual annotations. In *ACM MM*, 2008.

[23] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.

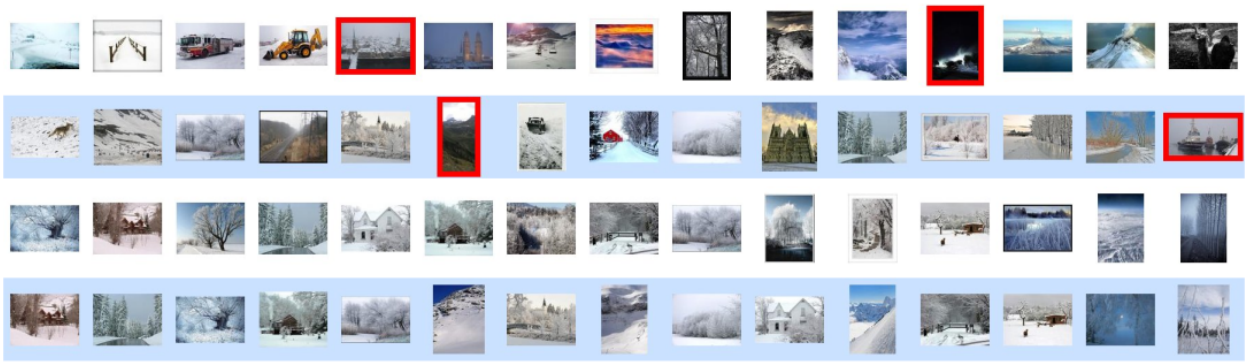
[24] K. Tieu and P. Viola. Boosting image retrieval. *Int. J. Comput. Vision*, 56(1-2):17–36, 2004.

[25] J. Uijlings, A. Smeulders, and R. Scha. Real-time bag-of-words, approximately. In *CIVR*, 2009.

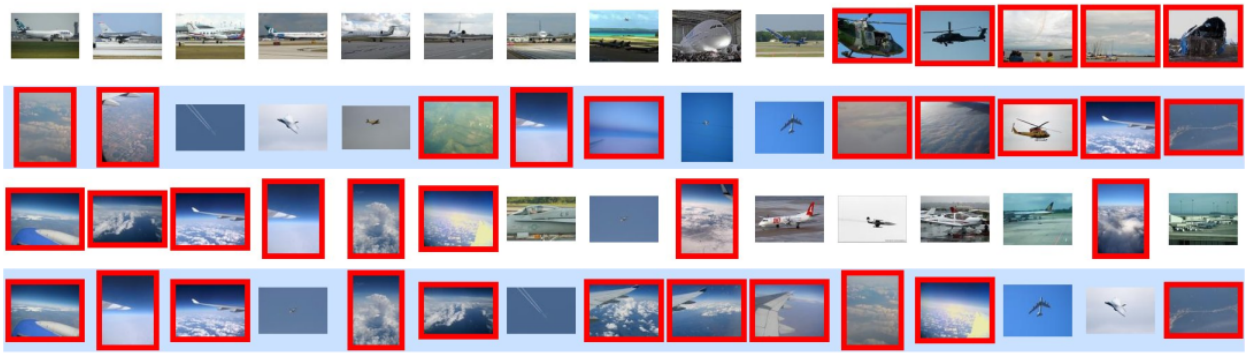
[26] L. Wu, L. Yang, N. Yu, and X.-S. Hua. Learning to tag. In *WWW*, 2009.

[27] H. Xu, J. Wang, X.-S. Hua, and S. Li. Tag refinement by regularized LDA. In *ACM MM*, 2009.

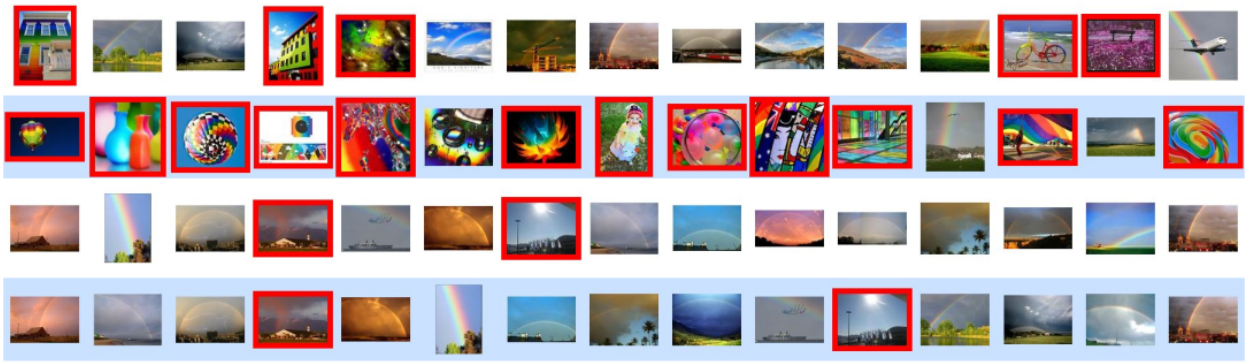
[28] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moment for content-based image retrieval. In *ICIP*, 2002.



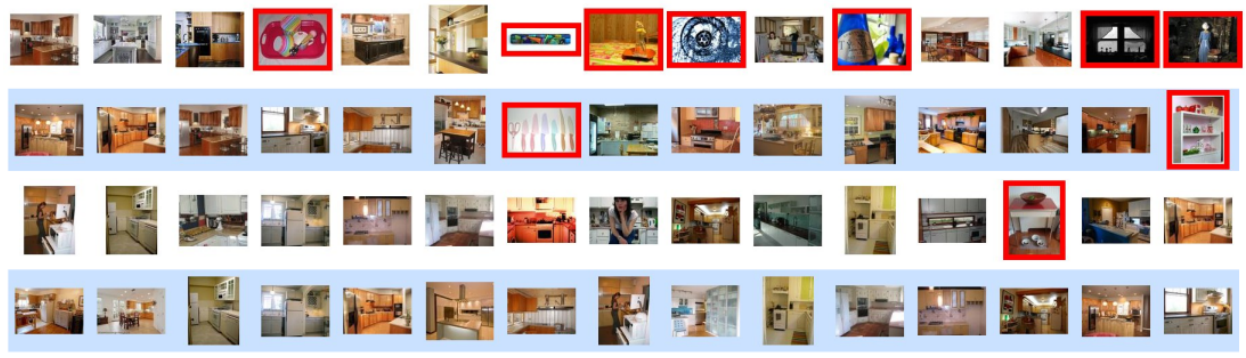
(a) Concept *snow*



(b) Concept *airplane*



(c) Concept *rainbow*



(d) Concept *kitchen*

Figure 4: Tag-based image retrieval results with and without (multi-feature) tag relevance learning for query concepts (a) *snow*, (b) *airplane*, (c) *rainbow*, and (d) *kitchen*. From the top row to the bottom row, each sub-figure shows top 15 results of a concept returned by the Tag Baseline, the worst single-feature tag relevance learner, the best single-feature tag relevance learner, and the proposed UniformTagger. Images with (red) borders are false positives with respect to the concept.