Classifying Tag Relevance with Relevant Positive and Negative Examples

Xirong Li[1] and Cees G. M. Snoek[2]
[1]Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China, China
[2]Intelligent Systems Lab Amsterdam, University of Amsterdam, the Netherlands
xirong@ruc.edu.cn, cgmsnoek@uva.nl

ABSTRACT

Image tag relevance estimation aims to automatically determine what people label about images is factually present in the pictorial content. Different from previous works, which either use only positive examples of a given tag or use positive and random negative examples, we argue the importance of *relevant positive* and *relevant negative* examples for tag relevance estimation. We propose a system that selects positive and negative examples, deemed most relevant with respect to the given tag from crowd-annotated images. While applying models for many tags could be cumbersome, our system trains efficient ensembles of Support Vector Machines per tag, enabling fast classification. Experiments on two benchmark sets show that the proposed system compares favorably against five present day methods. Given extracted visual features, for each image our system can process up to 3,787 tags per second. The new system is both effective and efficient for tag relevance estimation.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Object recognition*

Keywords

Image tag relevance, relevant examples, fast classification

1. INTRODUCTION

We consider the problem of estimating the relevance of a user-provided image tag, as exemplified in Fig. 1. Although all the three images are labeled with the tag 'sheep', only image (a) is genuinely a picture of sheep. Since image (b) is clearly dissimilar to a typical scene wherein a sheep is present, using a few positive examples of sheep as a reference for visual similarity will help separate (a) and (b), as is commonly done in the literature, *e.g.*, [3, 5, 6, 8, 10]. However, as image (a) and image (c) are similar in terms of their visual appearance as well, using the positive examples alone is limited for distinguishing between the two images. To resolve the issue, negative examples, which are visually close to the positives, have to be taken into account. Exploiting both positives and negatives with respect to a specific tag, as we propose, is important for tag relevance estimation.

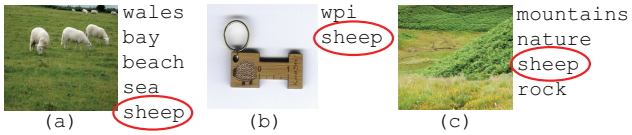

Figure 1: Examples of user-labeled images. Although all the three images are labeled with the tag 'sheep', only image (a) is truly a picture of sheep. While a few positive examples with respect to the tag will help separate (a) and (b), negative examples which are visually close to the positives are required to distinguish (a) from (c).

There are good efforts on selecting relevant positives [12] and relevant negatives [7] from crowd-annotated images in the context of visual categorization. However, what positive and negative examples to use for tag relevance estimation remains open. In fact, per-tag modeling has been challenged by [6], due to its quest for many well labeled training examples and the inefficiency in applying models for many tags. Recently, Chen *et al.* [3] propose to train Support Vector Machines per tag, and they adopt the linear kernel for reasons of efficiency. In their work, user-labeled images are directly used as positive examples, while negative examples are obtained by random sampling. Hence, the impact of relevant positives and relevant negatives on tag relevance estimation remains unclear.

In this paper we propose a *classification system* for tag relevance that takes into account both relevant positive and relevant negative examples. We build on the rich heritage from tag relevance estimation for positive example selection [6, 12]. We draw inspiration from recent advancements in visual categorization for negative example selection [7] and their efficient classification [7, 9]. Using the system we perform an empirical study to assess the value of relevant positive and negative examples for tag relevance estimation.

2. CLASSIFYING TAG RELEVANCE

Our goal is to build effective tag relevance estimators per tag, by exploiting the large amounts of crowd-annotated images on the Internet. For learning from large data, combin-

MM'13, October 21–25, 2013, Barcelona, Spain.

ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502129.

ing many classifiers built on small subsets of the data is a promising approach [2]. We thus follow this ensemble learning approach. To make our discussion more formal, we use ω to denote a tag of interest. Let x be an image. Its content-based representation is a d-dimensional feature vector. We refer to an image and its feature vector interchangeably, using $x(i)$ to indicate the i-th dimension of the vector. Let $G(x)$ be a tag relevance estimator for ω. We express $G(x)$ as an ensemble of T meta classifiers:

$$G(x) = \frac{1}{T}\sum_{t=1}^{T} g_t(x), \tag{1}$$

where $g_t(x)$ indicates the decision function of a meta classifier. We instantiate the meta classifiers using SVMs, for its well recognized performance on two-class learning:

$$g_t(x) = b_t + \sum_{j=1}^{n_t} \alpha_{t,j} \cdot y_{t,j} \cdot \mathcal{K}(x, x_{t,j}), \tag{2}$$

where b_t is the intercept, n_t the number of support vectors, $\alpha_{t,j}$ the positive coefficient of support vector $x_{t,j}$, $y_{t,j} \in \{1, -1\}$ a class label of $x_{t,j}$ with respect to ω, and $\mathcal{K}$ a kernel function.

Obtaining optimal $g_t(x)$ requires proper positive and negative training data. The relevance of negative examples with respect to ω depends on positive examples of the tag. In that regard, we first describe how to select relevant positive examples in Section 2.1, and then depict negative example selection in Section 2.2. While the selected positives and negatives lead to an effective ensemble of SVMs, the computational complexity of $G(x)$ is proportional to the size of the ensemble. We describe in Section 2.3 acceleration techniques which will make the complexity independent of the ensemble size. The proposed system is illustrated in Fig. 2.

2.1 Selecting Relevant Positive Examples

We choose to combine two state-of-the-art methods: semantic field [12] and neighbor voting [6]. As the two methods exploit textual and visual information respectively, they are orthogonal to each other. Combining them makes sense.

Given a specific tag ω, the semantic field method determines the positiveness of an image in light of the averaged semantic similarity between ω and the tags assigned to that image [12]. The semantic similarity between two tags is computed by combining the Flickr context similarity and the WordNet Wu-Palmer similarity. The Flickr similarity is based on the Normalized Google Distance, but with tag statistics acquired from Flickr image collections instead of Google indexed web pages. The WordNet similarity exploits path length in WordNet hierarchy to infer tag relatedness.

The neighbor voting method determines the positiveness of an image with respect to ω by exploiting tagging redundancies among multiple users [6]. The method retrieves k nearest neighbors from a large set of user-labeled images by content-based search. The number of neighbors labeled with ω is used as the positiveness score.

From the above discussion we see that the output scores of the two methods are of different scales. Hence, we use CombSUM with rank-based score normalization, a robust choice for multimedia fusion. Given images labeled with ω, we sort the images in descending order by their scores, and preserve the top l ranked results as relevant positives.

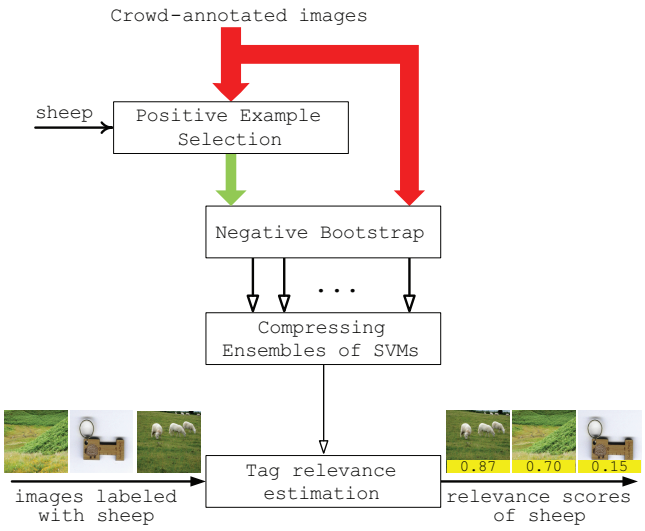

Figure 2: The proposed classification system for image tag relevance estimation. For each given tag, the system automatically selects a set of relevant positive examples from crowd-annotated images. Subsequently, relevant negative examples are selected via Negative Bootstrap [7], yielding an ensemble of SVMs. By compressing the ensemble to make the test complexity independent of the ensemble size, the system is both effective and efficient.

2.2 Selecting Relevant Negative Examples

While negative examples can be easily acquired by random sampling, see [3], such random negatives are inadequate for attacking challenging cases, like image (c) shown in Fig. 1. To separate (a) and (c), one might want to manually add positive examples of tags which resembles visual context of 'sheep', say 'grass' or 'hill'. However, the relevance of a negative example depends on the underlying visual features and classifiers, and is not necessarily consistent with what an observer may expect. It is thus difficult to specify relevant negatives by hand-crafted rules. In order to automatically select relevant negatives, we extend the Negative Bootstrap algorithm [7] to the tag relevance estimation problem. Different from [7] that departs from a few expert-labeled examples, we use purely crow-annotated examples.

Given the l positives selected in Section 2.1, Negative Bootstrap finds relevant negatives in an iterative manner. In the first iteration an initial classifier $g_1(x)$ is derived from the positives and l random negatives. In the t-th iteration, the algorithm randomly samples m examples to form a candidate set, and uses the ensemble of $t-1$ classifiers previously obtained to classify each candidate element. The top l most misclassified elements are selected and used together with the positives to derive a new meta classifier $g_t(x)$. Negative Bootstrap with T iterations produces an ensemble of T meta classifiers, which will be the tag relevance estimator for ω.

2.3 Compressing Ensembles of SVMs

As noted earlier, despite the effectiveness of ensemble learning, the intensive computation associated with applying all meta classifiers puts the practical use of per-tag modeling into question. To overcome the difficulty, we study how to

accelerate ensembles of SVMs. In particular, we consider linear SVMs (used in [3], but they do not consider ensembles), and histogram intersection kernel SVMs (HIKSVMs) for which there are works on fast classification of a single classifier [9] and an ensemble [7].

For linear SVMs, we sum over the i-th dimension of support vectors of the meta classifiers, i.e.,

$$C_i = \sum_{t=1}^{T} \sum_{j=1}^{n_t} \alpha_{t,j} \cdot y_{t,j} \cdot x_{t,j}(i). \quad (3)$$

Combining Eq. (2) and Eq. (3), we can rewrite $G(x)$ as

$$\frac{1}{T}(b_0 + \sum_{i=1}^{d} x(i) \cdot C_i), \quad (4)$$

where $b_0 = \sum_{t=1}^{T} b_t$. As both C_i and b_0 are constant with respect to x, computing Eq. (4) has an order of $O(d)$ only.

For HIKSVMs, the nonlinear kernel does not allow us to compress each dimension using Eq. (3). Instead, we follow [7], constructing the decision function per dimension as

$$H_i(z) = \sum_{t=1}^{T} \sum_{j=1}^{n_t} \alpha_{t,j} \cdot y_{t,j} \cdot \min(z, x_{t,j}(i)), \quad (5)$$

where z is a variable. The paper [7] shows that for any z, there exists a specific pair of ordered points $z_{i,r}$ and $z_{i,r+1}$ such that $H_i(z)$ can be computed by linear interpolation on $H_i(z_{i,r})$ and $H_i(z_{i,r+1})$. Further, by uniformly quantizing the value range of each dimension into q segments, and having $H_i(z)$ of the $q+1$ points precomputed, computing Eq. (1) boils down to doing a linear interpolation operation for each dimension, followed by summing over $d+1$ values. Hence, the time complexity of executing the compressed ensemble of HIKSVMs is now reduced to $O(d)$ also.

In sum, for a given tag, the proposed system automatically selects relevant positive and relevant negative examples from crowd-annotated images with no need of extra manual annotation. Such relevant examples result in ensembles of classifiers with better discrimination ability than their counterparts derived from random examples. Compressing the ensembles ensures fast classification, with a time complexity independent of the number of meta classifiers and the number of support vectors.

3. EMPIRICAL STUDY

3.1 Experimental Setup

Training data. To collect crowd-annotated images for training, we use over 25,000 WordNet tags as queries to uniformly sample Flickr images uploaded between 2005 and 2010. After removing batch-tagged images and those failed to extract visual features, we obtain 964,849 images.

Two test sets. We use two public test sets: Social20 [6] and NUSWIDE [4]. The Social20 set[1] consists of 19,971 Flickr images, with ground truth available for 20 tags corresponding to visual objects and scenes such as 'airplane', 'sheep', and 'street'. The NUSWIDE test set[2] contains 103,688 Flickr images, with ground truth available for 81 tags covering a range of objects, scenes, and events. The training and test sets were collected independently (data statistics shown in Table 1).

[1] `mediamill.nl`

[2] `http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm`. While the NUSWIDE test set originally contains 107,859 images, 4171 images are no longer available on Flickr.

Table 1: The training set and the two test sets.

	Training	Social20 [6]	NUSWIDE [4]
No. images	964,849	19,971	103,688
No. users	145,029	10,972	32,415
No. test tags	N.A.	20	81

Evaluation criteria. For each test tag, we sort images labeled with this tag in descending order by their relevance scores, and compute Average Precision on the entire ranked list. For overall comparison, we report mean Average Precision (mAP).

We extract a 1,024-dimensional bag of visual codes feature, by quantizing densely sampled SIFT descriptors [11]. For each test tag, we empirically preserve the top 500 ranked examples from the training set as its relevant positives. We then seek relevant negatives by running negative bootstrap with 10 iterations. Meta classifiers are trained using LIBSVM [1] with its default cost parameter because the positive and negative examples are perfectly balanced.

Baselines. We compare with the following five present day methods: tag position [10], tag ranking [8], neighbor voting [6], semantic field [12], and linear SVMs [3]. For a fair comparison, all methods use the same training data.

3.2 Experiments

Experiment 1. The Impact of Relevant Examples. In order to justify the need of relevant examples, we compare the following four strategies: 1) random positives with random negatives; 2) relevant positives with random negatives; 3) random positives with relevant negatives, and 4) relevant positives with relevant negatives. For a fair comparison, whenever applicable we will make the four strategies share the same input and parameters.

As shown in Table 2, for both linear SVMs and HIKSVMs, relevant positives in combination with relevant negatives perform best. Substituting relevant positives for random positives is useful, improving mAP of HIKSVMs from 0.781 to 0.794 on Social20 and from 0.624 to 0.633 on NUSWIDE. Since the relevance of negative examples depends on positive examples, unreliable positives could confuse negative bootstrap. Consequently, we observe that given random positives, substituting relevant negatives for random negatives is useless in general. We conclude from the results that both relevant positives and relevant negatives are important for tag relevance estimation, and they have to be used together.

Experiment 2. Efficiency Analysis. Having visual features extracted, running the compressed ensemble of HIKSVMs for a given tag costs merely 0.264 millisecond per image, on average, on our machine with 2.4 GHz multi-core cpu and 24 GB memory. The corresponding number for linear SVMs is 0.221 millisecond. This means, for each image, the proposed system (with fast ensembles of HIKSVMs) can process 3,787 tags per second. Noticing that the averaged number of distinct user tags per image is around 6 [6], our per-tag modeling approach is practical.

Experiment 3. Comparison with Present Day Methods. As shown in Table 3, the proposed system compares favorably over all five baselines. Compared to the best baseline, *i.e.*,

Table 2: The impact of relevant examples on tag relevance estimation. Combining relevant positive examples (RelPos) and relevant negative examples (RelNeg) works best.

SVMs	RelPos	RelNeg	Social20	NUSWIDE
Linear	No	No	0.680	0.572
	Yes	No	0.694	0.582
	No	Yes	0.667	0.569
	Yes	**Yes**	**0.734**	**0.604**
Histogram intersection kernel	No	No	0.781	0.624
	Yes	No	0.794	0.633
	No	Yes	0.781	0.636
	Yes	**Yes**	**0.817**	**0.655**

Table 3: Comparing different methods for tag relevance estimation on the two test sets.

Method	Social20	NUSWIDE
tag position [10]	0.565	0.560
tag ranking [8]	0.643	0.575
neighbor voting [6]	0.772	0.617
semantic field [12]	0.647	0.577
linear SVMs [3]	0.680	0.572
This work	**0.817**	**0.655**

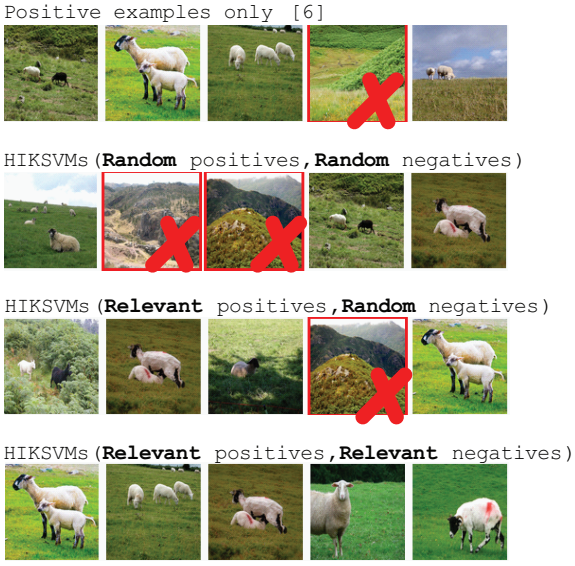

Figure 3: The top 5 results of 'sheep' sorted by tag relevance scores produced by different estimators.

neighbor voting which scores an mAP of 0.772 on Social20 and 0.617 on NUSWIDE, our system reaches an mAP of 0.817 and 0.655 on the two test sets. Concerning the choice of SVMs, HIKSVMs clearly outperform linear SVMs. In addition, since the positive selection method presented in Section 2.1 can also be used for tag relevance estimation, we compare that method, and find that our system beats it as well. Fig. 3 shows a qualitative result. These results justify the effectiveness of the proposed system for tag relevance estimation.

4. CONCLUSIONS

This paper presents a per-tag classification approach to image tag relevance estimation. We build a system which exploits both relevant positive and relevant negative examples with respect to individual tags. Experiments on two benchmark sets support the following conclusions. As the main contribution of this work, we empirically show that relevant positives and relevant negatives are important for constructing effective tag relevance estimators, and such relevant examples can be automatically selected from crowd-annotated images with no need of extra manual annotation. We find that relevant positives and relevant negatives have to be used simultaneously. Replacing relevant positives (or negatives) by random positives (or negatives) would yield suboptimal performance. Concerning the choices of classifiers, histogram intersection kernel SVMs (HIKSVMs) are superior to linear SVMs. Equipped with ensembles of HIKSVMs trained on relevant positives and relevant negatives, the system beats five present day methods. Moreover, with fast classification of the ensembles, our system can process up to 3,787 tags per second. The effectiveness and efficiency make the proposed tag relevance estimation system promising for real-world deployment.

Acknowledgements. This research was supported by the Basic Research funds in Renmin University of China from the central government (No. 13XNLF05), the Dutch national program COMMIT, and the STW STORY project.

5. REFERENCES

[1] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *TIST*, 2011.

[2] N. Chawla, L. Hall, K. Bowyer, and W. Kegelmeyer. Learning ensembles from bites: A scalable and accurate approach. *JMLR*, 2004.

[3] L. Chen, D. Xu, I. Tsang, and J. Luo. Tag-based web photo retrieval improved by batch mode re-tagging. In *CVPR*, 2010.

[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] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu. Visual-textual joint relevance learning for tag-based social image search. *TIP*, 2013.

[6] X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *TMM*, 2009.

[7] X. Li, C. Snoek, M. Worring, D. Koelma, and A. Smeulders. Bootstrapping visual categorization with relevant negatives. *TMM*, 2013.

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

[9] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[10] A. Sun, S. Bhowmick, K. Nguyen, and G. Bai. Tag-based social image retrieval: An empirical evaluation. *JASIST*, 2011.

[11] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *TPAMI*, 2010.

[12] S. Zhu, Y.-G. Jiang, and C.-W. Ngo. Sampling and ontologically pooling web images for visual concept learning. *TMM*, 2012.