

TagProp: Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-Annotation

Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek and Cordelia Schmid

LEAR, INRIA Grenoble

Laboratoire Jean Kuntzmann

firstname.lastname@inrialpes.fr

Abstract

Image auto-annotation is an important open problem in computer vision. For this task we propose TagProp, a discriminatively trained nearest neighbor model. Tags of test images are predicted using a weighted nearest-neighbor model to exploit labeled training images. Neighbor weights are based on neighbor rank or distance. TagProp allows the integration of metric learning by directly maximizing the log-likelihood of the tag predictions in the training set. In this manner, we can optimally combine a collection of image similarity metrics that cover different aspects of image content, such as local shape descriptors, or global color histograms. We also introduce a word specific sigmoidal modulation of the weighted neighbor tag predictions to boost the recall of rare words. We investigate the performance of different variants of our model and compare to existing work. We present experimental results for three challenging data sets. On all three, TagProp makes a marked improvement as compared to the current state-of-the-art.

1. Introduction

Image auto-annotation is an active subject of research [7, 15, 16, 18]. The goal is to develop methods that can predict for a new image the relevant keywords from an annotation vocabulary. These keyword predictions can be used either to propose tags for an image, or to propose images for a tag or a combination of tags. Such methods are becoming more and more important given the growing collections of user-provided visual content, e.g. on photo or video sharing sites, and desktop photo management applications. These large-scale collections feed the demand for automatic retrieval and annotation methods. Since the amount of images with more or less structured annotations is also increasing, this allows the deployment of machine learning techniques to leverage this potential by estimating accurate tag prediction models.

Although the general problem is a difficult one, progress

has been made in the research community by evaluations on standardized annotated data sets. In the next section we will detail the related work that is most closely linked to ours. The main shortcomings of existing work are twofold. First, models are often estimated to maximize generative likelihood of image features and annotations, which might not be optimal for tag prediction. Second, many parametric models are not rich enough to accurately capture the intricate dependencies between image content and annotations. Non-parametric nearest neighbor like methods have been found to be quite successful for tag prediction [5, 11, 13, 17, 22, 27]. This is mainly due to the high ‘capacity’ of such models: they can adapt flexibly to the patterns in the data as more data is available. However, existing nearest neighbor type methods do not allow for integrated learning of the metric that defines the nearest neighbors in order to maximize the predictive performance of the model. Either a fixed metric [5, 27] or adhoc combinations of several metrics [17] are used, despite many recent work showing the benefits of metric learning for many computer vision tasks such as image classification [12], image retrieval [10], or visual identification [9].

In this paper we present TagProp, short for Tag Propagation, a new nearest neighbor type model that predicts tags by taking a weighted combination of the tag absence/presence among neighbors. Our contributions are the following. First, the weights for neighbors are either determined based on the neighbor rank or its distance, and set automatically by maximizing the likelihood of annotations in a set of training images. With rank based weights the k -th neighbor always receives a fixed weight, whereas distance based weights decay exponentially with the distance. Our tag prediction model is conceptually simple, yet outperforms the current state-of-the-art methods using the same feature set. Second, contrary to earlier work, our model allows the integration of metric learning. This enables us to optimize e.g. a Mahalanobis metric between image features – or, less costly, a combination of several distance measures – to define the neighbor weights for the tag prediction task.

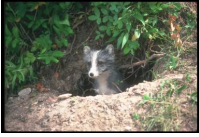





Corel 5k	ESP Game	IAPR TC12
 <div> arctic den fox grass tree (0.99) grass (0.94) rocks (0.91) flowers (0.82) tiger (0.82) </div>	 <div> box brown square square white box (1.00) square (1.00) brown (1.00) white (0.79) yellow (0.72) </div>	 <div> glacier mountain people tourist glacier (1.00) mountain (1.00) front (0.64) sky (0.58) people (0.58) </div>
 <div> iguana lizard marine rocks iguana (1.00) marine (1.00) lizard (1.00) water (0.67) sky (0.66) </div>	 <div> blue cartoon man woman man (0.98) anime (0.96) cartoon (0.92) people (0.89) woman (0.88) </div>	 <div> landscape lot meadow water llama (1.00) water (1.00) landscape (1.00) front (0.60) people (0.51) </div>

Figure 1. Example test images from the three data sets. Next to each image, we show the ground truth annotation (left), and the five tags with highest relevance predictions (correct ones are underlined) given by our TagProp model (σ ML variant with $K = 200$). Note the large diversity between the data sets, and that the ground truth annotations do not always contain all relevant tags (e.g. ‘water’ for the bottom left image), and sometimes contain tags for which one can argue whether they are relevant or not (e.g. ‘lot’ for the bottom right image).

Third, TagProp includes word-specific logistic discriminant models. These models use the tag predictions of the word-invariant models as inputs and are able, using just two parameters per word, to boost or suppress the tag presence probabilities for very frequent or rare words. This results in a significant increase in the number of words that are recalled, i.e. assigned to at least one test image.

To evaluate our models and to compare to previous work, we use three data sets – Corel 5k, IAPR TC12 and ESP Game – and standard measures including precision, recall, mean average precision and break-even point. In Figure 1 we show several examples of images with their annotations, and predictions from our model. On all data sets and measures we show significantly improved accuracy of our method as compared to earlier work.

The rest of this paper is organized as follows. In the next section we give an overview of the related work. Then, in Section 3, we present our tag prediction models, and how we estimate their parameters. In Section 4 we present the three data sets, evaluation criteria as well as the image features we use in our experiments. The experimental results are presented in Section 5. In Section 6 we present our conclusions and directions for further research.

2. Related Work

In this section we discuss models for image annotation and keyword based retrieval most relevant for our work. We identify four main groups of methods: those based on topic models or mixture models, discriminatively trained ones, and nearest neighbor type models.

The first group of methods are based on topic models such as latent Dirichlet allocation, probabilistic latent semantic analysis, and hierarchical Dirichlet processes, see e.g. [1, 20, 25]. These methods model annotated images as samples from a specific mix of topics, where each topic is a distribution over image features and annotation words. Parameter estimation involves estimating the topic mix for

each image, and estimating the data distributions of the topics. Most often, a multinomial distribution over words is used, and a Gaussian over visual features from different regions of the image. Methods inspired by machine translation [4], in this case translating from discrete visual features to the annotation vocabulary, can also be understood as topic models, using one topic per visual descriptor type.

A second family of methods uses mixture models to define a joint distribution over image features and annotation tags. To annotate a new image, these models compute the conditional probability over tags given the visual features by normalising the joint likelihood. Sometimes a fixed number of mixture components over visual features per keyword is used [2], while other models use the training images as components to define a mixture model over visual features and tags [5, 11, 13]. Each training image defines a likelihood over visual features and tags by a smoothed distribution around the observed values. These models can be seen as non-parametric density estimators over the co-occurrence of images and annotations. For visual features Gaussians are used, while the distributions over annotations are multinomials, or separate Bernoullis for each word.

Both families of generative models discussed above may be criticized because they maximize the generative data likelihood, which is not necessarily optimal for predictive performance. Therefore, discriminative models for tag prediction have also been proposed [3, 7, 10]. These methods learn a separate classifier for each tag, and use these to predict for each test image whether it belongs to the class of images that are annotated with each particular tag. Different learning methods have been used, including support vector machines, multiple-instance learning, and Bayes point machines. Notable is [7] which also addresses the problem of retrieving images based on multi-word queries.

Given the increasing amount of training data that is currently available, local learning techniques are becoming more attractive as a simple yet powerful alternative to

parametric models. Examples of such techniques include methods based on label diffusion over a similarity graph of labeled and unlabeled images [16, 22], or learning discriminative models in neighborhoods of test images [27]. A simpler adhoc nearest-neighbor tag transfer mechanism was recently introduced [17], showing state-of-the-art performance. There, nearest neighbors are determined by the average of several distances computed from different visual features. The authors also combine the base distances by learning a binary classifier separating image pairs that have several tags in common from images that do not share any tags. However, this linear distance combination did not give better results than an equally weighted combination.

3. Tag Relevance Prediction Models

Our goal is to predict the relevance of annotation tags for images. Given these relevance predictions we can annotate images by ranking the tags for a given image, or do keyword based retrieval by ranking images for a given tag. Our proposed method is based on a weighted nearest neighbor approach, inspired by recent successful methods [5, 11, 13, 17], that propagate the annotations of training images to new images. Our models are learnt in a discriminative manner, rather than using held-out data [5], or using neighbors in an adhoc manner [17]. We assume that some visual similarity or distance measures between images are given, abstracting away from their precise definition.

3.1. Weighted Nearest Neighbor Tag Prediction

To model image annotations, we use Bernoulli models for each keyword. This choice is natural because keywords, unlike natural text where word frequency is meaningful, are either present or absent. The dependencies between keywords in the training data are not explicitly modeled, but are implicitly exploited in our model.

We use $y_{iw} \in \{-1, +1\}$ to denote the absence/presence of keyword w for image i , hence encoding the image annotations. The tag presence prediction $p(y_{iw} = +1)$ for image i is a weighted sum over the training images, indexed by j :

$$p(y_{iw} = +1) = \sum_j \pi_{ij} p(y_{iw} = +1|j), \quad (1)$$

$$p(y_{iw} = +1|j) = \begin{cases} 1 - \epsilon & \text{for } y_{jw} = +1, \\ \epsilon & \text{otherwise,} \end{cases} \quad (2)$$

where π_{ij} denotes the weight of image j for predicting the tags of image i . We require that $\pi_{ij} \geq 0$, and $\sum_j \pi_{ij} = 1$. We use ϵ to avoid zero prediction probabilities, and in practice we set $\epsilon = 10^{-5}$. To estimate the parameters that control the weights π_{ij} we maximize the log-likelihood of the predictions of training annotations. Taking care to set the weight of training images to themselves to zero, i.e.

$\pi_{ii} = 0$, our objective is to maximize

$$\mathcal{L} = \sum_{i,w} c_{iw} \ln p(y_{iw}), \quad (3)$$

where c_{iw} is a cost that takes into account the imbalance between keyword presence and absence. Indeed, in practice, there are many more tag absences than presences, and absences are much noisier than presences. This is because most tags in annotations are relevant, but often the annotation does not include all relevant tags. We set $c_{iw} = 1/n^+$ if $y_{iw} = +1$, where n^+ is the total number of positive labels, and likewise $c_{iw} = 1/n^-$ when $y_{iw} = -1$.

Rank-based weights. In the case of rank-based weights over K neighbors we set $\pi_{ij} = \gamma_k$ if j is the k -th nearest neighbor of i . The data log-likelihood (3) is concave in the parameters γ_k and can be estimated using an EM-algorithm, or a projected-gradient algorithm. The derivative of Eq. (3) with respect to γ_k equals

$$\frac{\partial \mathcal{L}}{\partial \gamma_k} = \sum_{i,w} \frac{c_{iw} p(y_{iw}|n_{ik})}{p(y_{iw})}, \quad (4)$$

where n_{ik} denotes the index of the k -th neighbor of image i . The number of parameters equals the neighborhood size K . We refer to this variant as RK, for “rank-based”.

Distance-based weights. The other possibility is to define the weights directly as a function of the distance, rather than the rank. This has the advantage that weights will depend smoothly on the distance, which is crucial if the distance is to be adjusted during training. The weights of training images j for an image i are redefined as

$$\pi_{ij} = \frac{\exp(-d_{\theta}(i, j))}{\sum_{j'} \exp(-d_{\theta}(i, j'))}, \quad (5)$$

where d_{θ} is a distance metric with parameters θ that we want to optimize. Note that the weights π_{ij} decay exponentially with distance d_{θ} to image i . Choices for d_{θ} include Mahalanobis distances d_M parametrized by a semi-definite matrix M , and $d_w(i, j) = \mathbf{w}^T \mathbf{d}_{ij}$ where \mathbf{d}_{ij} is a vector of base distances between image i and j , and \mathbf{w} contains the positive coefficients of the linear distance combination. The number of parameters equals the number of base distances that are combined. In the rest of the paper we focus on this particular case. When we use a single distance, referred to as the SD variant, \mathbf{w} is a scalar that controls the decay of the weights with distance, and it is the only parameter of the model. When multiple distances are used, the variant is referred to as ML, for “metric learning”.

Again, rather than using an EM-algorithm we directly maximize the log-likelihood using a projected gradient algorithm under positivity constraints on the elements of \mathbf{w} .

Using the new definition of the weights, the gradient of the log-likelihood Eq. (3) with respect to \mathbf{w} equals

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \sum_{i,j} W_i (\pi_{ij} - \rho_{ij}) \mathbf{d}_{ij}, \quad (6)$$

where $W_i = \sum_w c_{iw}$ and ρ_{ij} denotes the weighted average over all words w of the posterior probability of neighbor j for image i given the annotation:

$$\rho_{ij} = \sum_w \frac{c_{iw}}{W_i} p(j|y_{iw}). \quad (7)$$

To reduce the computational cost of training the model, we do not compute all pairwise π_{ij} and ρ_{ij} . Rather, for each i we compute them only over a large set, and assume the remaining π_{ij} and ρ_{ij} to be zero. For each i , we select K neighbors such that we maximise $k^* = \min\{k_d\}$, where k_d is the largest neighbor rank for which neighbors 1 to k of base distance d are included among the selected neighbors. In this way we are likely to include all images with large π_{ij} regardless of the distance combination \mathbf{w} that is learnt. Therefore, after determining these neighborhoods, our algorithm scales linearly with the number of training images.

Note the relation of our model to the multi-class metric learning approach of [6]. In that work, a metric is learnt such that weights π_{ij} as defined by Eq. (5) are as close as possible in the sense of Kullback-Leibler (KL) divergence to fixed set of target weights ρ_{ij} . The target weights were defined to be zero for pairs from different classes, and set to a constant for all pairs from the same class. In fact, when deriving an EM-algorithm for our model, we find the objective of the M-step to be of the form of a KL divergence between the ρ_{ij} (fixed to values computed in the E-step) and the π_{ij} . For fixed ρ_{ij} this KL divergence is convex in \mathbf{w} .

3.2. Word-specific Logistic Discriminant Models

Weighted nearest neighbor approaches tend to have relatively low recall scores, which is easily understood as follows. In order to receive a high probability for the presence of a tag, it needs to be present among most neighbors with a significant weight. This, however, is unlikely to be the case for rare tags. So, even if we are lucky enough to have a few neighbors annotated with the tag, we will predict the presence with a low probability.

To overcome this, we introduce word-specific logistic discriminant models that can boost the probability for rare tags and decrease it for very frequent ones. The logistic model uses weighted neighbor predictions by defining

$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w), \quad (8)$$

$$x_{iw} = \sum_j \pi_{ij} y_{jw}, \quad (9)$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ and x_{iw} is the weighted average of annotations for tag w among the neighbors of i , which is equivalent to Eq. (1) up to an affine transformation. The word-specific models add 2 parameters to estimate for each word. The resulting modulated variants are referred to as σ RK, σ SD and σ ML, respectively.

For fixed π_{ij} the model is a logistic discriminant model, and the log-likelihood is concave in $\{\alpha_w, \beta_w\}$, and can be trained per keyword. Using the new model, the gradient of the log-likelihood of the training annotations with respect to the parameters θ that control the weights equals

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{i,w} c_{iw} \alpha_w p(-y_{iw}) y_{iw} \frac{\partial x_{iw}}{\partial \theta}, \quad (10)$$

and for the model based on rank or distance respectively

$$\frac{\partial x_{iw}}{\partial \gamma_k} = y_{n_{ik}w}, \quad (11)$$

$$\frac{\partial x_{iw}}{\partial \mathbf{w}} = \sum_j \pi_{ij} (x_{iw} - y_{jw}) \mathbf{d}_{ij}. \quad (12)$$

In practice we estimate the parameters θ and $\{\alpha_w, \beta_w\}$ in an alternating fashion. We observe rapid convergence, typically after alternating the maximization three times.

4. Data Sets and Experimental Setup

In this section we first present the data sets used in our experiments, then in Section 4.2 we describe the different features that we extract from images to compute distance measures between images, and in Section 4.3 we discuss the evaluation measures for image annotation and retrieval.

4.1. Data Sets

We consider three publicly available data sets that have been used in previous work, and allow for direct comparison. Table 1 summarizes some statistics of these data sets, example images are shown in Figure 1.

Corel 5k. This data set was first used in [4]. Since then, it has become an important benchmark for keyword based image retrieval and image annotation. It contains around 5000 images manually annotated with 1 to 5 keywords. The vocabulary contains 260 words. A fixed set of 499 images are used as test, and the rest is used for training.

ESP Game. This data set is obtained from an online game where two players, that can not communicate outside the game, gain points by agreeing on words describing the image [24]. This way the players are encouraged to provide important and meaningful tags to images. We use the subset of 20,000, out of the 60,000 images publicly available, that was also used in [17]. This data set is very challenging, as it contains a wide variety of images including: logos, drawings, and personal photos.

	Corel 5k	ESP Game	IAPR TC12
Vocabulary size	260	268	291
Nr. of images	4,493	18,689	17,665
Words per img.	3.4 / 5	4.7 / 15	5.7 / 23
Img. per word	58.6 / 1004	362.7 / 4553	347.7 / 4999

Table 1. Statistics of the training sets of the three data sets. Image and word counts are given in the format mean / maximum. Statistics for the test sets resemble closely those of the training sets.

IAPR TC12. This set of 20,000 images accompanied with descriptions in several languages was initially published for cross-lingual retrieval [8]. It can be transformed into a format comparable to the other sets by extracting common nouns using natural language processing techniques. We use the same resulting annotation as in [17].

4.2. Feature Extraction

We extract different types of features commonly used for image search and categorisation. We use two types of global image descriptors: Gist features [21], and color histograms with 16 bins in each color channel for RGB, LAB, HSV representations. Local features include SIFT as well as a robust hue descriptor [23], both extracted densely on a multi-scale grid or for Harris-Laplacian interest points. Each local feature descriptor is quantized using k-means on samples from the training set. Images are then represented as a ‘bag-of-words’ histogram. All descriptors but Gist are L1-normalised and also computed in a spatial arrangement [14]. We compute the histograms over three horizontal regions of the image, and concatenate them to form a new global descriptor, albeit one that encodes some information of the spatial layout of the image. To limit color histogram sizes, here, we reduced the quantization to 12 bins in each channel. Note that this spatial binning differs from segmented image regions, as used in some previous work.

This results in 15 distinct descriptors, namely one Gist descriptor, 6 color histograms and 8 bag-of-features (2 features types x 2 descriptors x 2 layouts). To compute the distances from the descriptors we follow previous work and use L2 as the base metric for Gist, L1 for global color histograms, and χ^2 for the others.

4.3. Evaluation Measures

We evaluate our models with standard performance measures, used in previous work, that evaluate retrieval performance per keyword, and then average over keywords.

Precision and recall for fixed annotation length. Following [4], each image is annotated with the 5 most relevant keywords. Then, the mean precision P and recall R over keywords are computed. N_+ is used to denote the number of keywords with non-zero recall value. Note that each

image is forced to be annotated with 5 keywords, even if the image has fewer or more keywords in the ground truth. Therefore, even if a model predicts all ground-truth keywords with a significantly higher probability than other keywords, we will not measure perfect precision and recall.

Precision at different levels of recall. We also evaluate precision at different levels of recall as in [7]. The break-even point (**BEP**), or R-precision, measures for each keyword w the precision among the top n_w relevant images, where n_w is the number of images annotated with this keyword in the ground truth. The mean average precision (**mAP**) over keywords is found by computing for each keyword the average of the precisions measured after each relevant image is retrieved.

5. Experimental Results

In this section we present a quantitative evaluation of TagProp and compare to previous work, qualitative results can be found in Figure 1. We first give a detailed presentation of results obtained on the Corel 5k data set, and compare them to previous work. In Section 5.2 we present our results for the IAPR TC12 and ESP Game data sets. Results for multi-word image retrieval are presented in Section 5.3.

5.1. Results for the Corel 5k data set

In a first set of experiments we compare the different variants of TagProp and compare them to the original results of [17], referred to as JEC, and also using our own features (JEC-15). That is, we take an equally weighted combination of our 15 normalized base distances to define image similarity.

From the results in Table 2 we can make several observations. First, using the tag transfer method proposed in [17] with our own features we obtain results very similar to the original work. Thus, other performance differences obtained using our methods must be due to the tag prediction methods. Our models that use this fixed distance combination to define weights (either directly in SD or using ranks in RK) perform comparably. Among these results, the ones of the sigmoidal model using distance-based weights (σ SD) are the best, and they show a modest improvement over the results obtained with the more adhoc JEC-15.

More importantly, using our models that integrate metric learning (ML and σ ML), much larger improvements are obtained, in particular using the σ ML variant. Compared to the current state-of-the-art method using the same features, we obtain marked improvements of 5% in precision, 9% in recall, and count 20 more words with positive recall. This result shows clearly that nearest neighbor type tag prediction can benefit from metric learning. Interestingly, earlier efforts to exploit metric learning did not succeed [17], *c.f.* Section 2. The key to our successful use of metric learning is its integration in the prediction model.

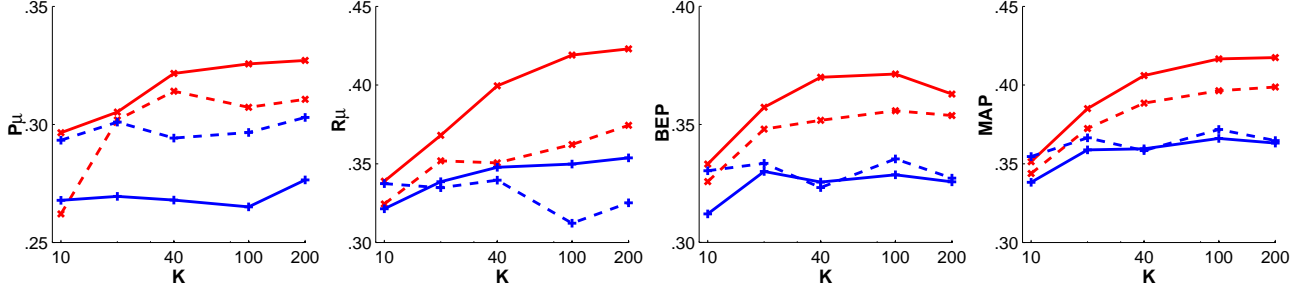


Figure 2. Performance on the Corel data set of our distance-based models in terms of P , R , BEP , and mAP with respect to the neighborhood size K . The dashed curves correspond to the ML (red) and SD (blue) variants. The solid curves are the sigmoidal modulated extensions: σML (red) and σSD (blue).

	Previously reported results								TagProp					
	CRM [13]	InfNet[19]	NPDE [26]	SML [2]	MBRM [5]	TGLM [16]	JEC [17]	JEC-15	RK	σRK	SD	σSD	ML	σML
P	16	17	18	23	24	25	27	28	28	26	30	28	31	33
R	19	24	21	29	25	29	32	33	32	34	33	35	37	42
$N+$	107	112	114	137	122	131	139	140	136	143	136	145	146	160

Table 2. Overview of performance in terms of P , R , and $N+$ of our models (using $K = 200$), and those reported in a selection of earlier work. JEC-15 refers to our implementation of [17] using our 15 distances. We show results for our variants: RK and SD using the equal distance combination, ML which integrates metric learning, as well as their modulated extensions (σRK , σSD and σML , respectively).

In Figure 2 we show performance in terms of P , R , BEP , and mAP , for the distance based models as a function of the number of neighbors over which the weights are evaluated. From these results we conclude that consistently over all number of neighbors, with or without σ , the metric learning finds distance combinations that outperform the equal weighted combination. Furthermore, we see that σML has a major impact on the R measure, and also leads to improvements in the other measures. We observe that using large numbers of neighbors improves performance, at least up to 100. This is true in particular for the ML variants. This can be explained by the fact that, in ML, the ranking of neighbor images change with the learnt metric. Thus, in order to ensure that all useful training images are included in the initial neighborhoods (computed from the base distances, *c.f.* Section 3), these sets should be large enough.

In Figure 3 we further analyze which words benefit most from the improved recall in σ variants. As expected, the improvement is higher for rare words and only the few most frequent words are penalized in terms of recall, but even in this situation the performance remains high and many relevant images are still retrieved.

From these first experiments we identify our distance based variants as the most promising, and use them below for the other two data sets. We use $K = 200$ as a default choice for the number of neighbors as it shows good performance for all measures.

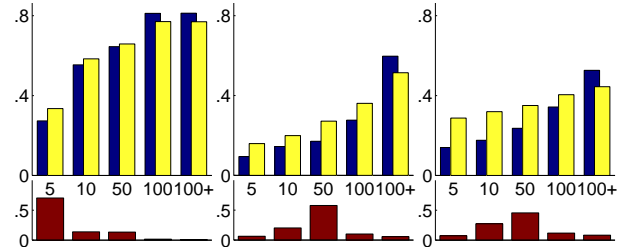


Figure 3. Mean recall of words in the three data sets (from left to right: Corel 5k, ESP Game, and IAPR TC12) for ML (blue) and σML (yellow), grouped with respect to their frequency in the data set. Keywords are binned based on how many images they occur in: the first bin groups words with less than 5 relevant images, the second bin words with between 6 and 10 images, and so on. The lower bars show the fraction of keywords in each bin, the upper bars show the average recall for the words in a bin.

5.2. Results for ESP Game and IAPR TC12

Next we present experimental results obtained on the ESP Game and IAPR TC12 data sets. Table 3 compares the current state-of-the-art for these data sets to our distance based variants, using equal and learned distance combinations. For reference we also include our implementation of [17] using our feature set. Again we find that our models significantly improve over the current state-of-the-art per-

		IAPR			ESP Game		
		<i>P</i>	<i>R</i>	N+	<i>P</i>	<i>R</i>	N+
MBRM [17]		24	23	223	18	19	209
JEC [17]		28	29	250	22	25	224
JEC-15		29	19	211	24	19	222
TagProp	SD	50	20	215	48	19	212
	σ SD	41	30	259	39	24	232
	ML	48	25	227	49	20	213
	σ ML	46	35	266	39	27	239

Table 3. Comparison of performance in terms of *P*, *R*, and *N+* on ESP and IAPR of the state-of-the-art [17] to our TagProp variants with distance based weights, using $K=200$ neighbors.

formance. In contrast to the Corel data set, here we find most significant improvement for precision rather than recall: 18% (resp. 17%) on *P*, 6% (resp. 2%) on *R*, and 16 (resp. 15) on *N+* on IAPR (resp. ESP).

As for Corel, we show in Figure 4 the influence of the neighborhood size on our different measures for both datasets for our σ SD and σ ML variants. We can again observe the benefit of integrated metric learning.

5.3. Image Retrieval from Multi-word Queries

Up to this point, we looked at image retrieval performance for single word queries. Most existing work concentrates on this problem, as it is difficult as such. However, any realistic image retrieval system should support multi-word queries as well. Here we present performance in terms of **BEP** and **mAP** on the Corel data set that include multi-word queries. To allow for direct comparison, we follow the setup of [7], which uses a subset of 179 words of the 260 annotation words of Corel 5k that appear at least twice in the test set. Images are considered relevant for a query when they are annotated with all words, and we consider all 2241 queries composed of one or more words such that the test set includes at least one image that is relevant. Further, the queries are divided into ‘difficult’ ones (1820) for which there are only one or two relevant images, and easy ones (421) with three or more relevant images.

To predict relevance of images for a multi-word query we compute the probability according to our model to observe all keywords in the query. Due to the probabilistic output of TagProp, this is easily done by taking the product over the single keyword probabilities, as our model does not explicitly account for dependencies between words. In Table 4 we summarize our results, and compare to those of PAMIR [7]. Our results improve by about 10% the **mAP** performance over all query types. Also in terms of **BEP** we gain 10% compared to PAMIR, which was found in [7] to outperform a number of alternative approaches.

		All-mAP	Single	Multi	Easy	Difficult	All-BEP
TagProp	PAMIR [7]	26	34	26	43	22	17
	SD	32	40	31	49	28	24
	σ SD	31	41	30	49	27	23
	ML	36	43	35	53	32	27
	σ ML	36	46	35	55	32	27

Table 4. Comparison of TagProp variants (using $K=200$) and PAMIR in terms of **mAP** and **BEP**. The **mAP** performance is also broken down over single-word and multi-word queries, easy and difficult ones. Only the 179 words that appear at least twice in test images are used, as in [7].

TagProp	<i>P</i>	<i>R</i>	<i>N+</i>	BEP	mAP
Corel 5K	32.7	42.3	160	36.3	41.8
IAPR	46.0	35.2	266	40.9	39.9
ESP-Game	39.2	27.4	239	31.3	28.1

Table 5. Overview of performance for the three data sets obtained with TagProp (σ ML variant with $K=200$).

6. Conclusions

We have introduced new models for image annotation and keyword based image retrieval. These models combine a weighted nearest-neighbor approach with metric learning capabilities in a discriminative framework. We add word-specific logistic discriminant modulation to deal with the varying word frequencies in a data-driven manner.

We reported extensive experimental results on three data sets, using several performance measures. From these results we conclude that our σ ML sigmoidal variant of TagProp that uses distance based weights and integrates metric learning performs best. It combines good recall with high precision over all data sets. We refer to Table 5 for an overview of the performance obtained with σ ML with all five evaluation measures. Consistently, it outperforms previously reported results for all three data sets. It gives significant improvements over the same model applied to uniformly combined distances, σ SD. This contrasts with earlier attempts to use metric learning for tag prediction, see e.g. [17], that were unsuccessful because metric learning was not integrated in the prediction model. Our word specific modulation significantly improves recall for rare words as well as the overall performance. On all three data sets, and all five evaluation measures, our model achieves performance well above all previously published results.

In future work, we will consider extending the model to assign tags to image regions, in order to address tasks such as image region labelling and object detection from image-wide annotations.

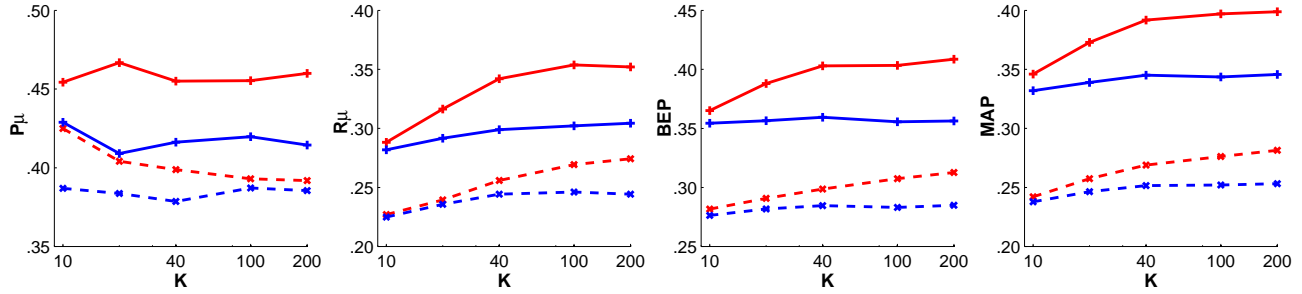


Figure 4. Performance of our sigmoidal modulated distance-based variants of TagProp in terms of P , R , BEP , and mAP , on the ESP (solid) and IAPR (dashed) datasets with respect to the neighborhood size K . Red curves correspond to the learnt combination, σML , and blue ones to an equal combination, σSD .

Acknowledgements

This work was supported by the European funded research project CLASS.

References

- [1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003.
- [2] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3):394–410, 2007.
- [3] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *Proceedings Internet imaging (SPIE)*, volume 5304, 2004.
- [4] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.
- [5] S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *CVPR*, 2004.
- [6] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS*, 2006.
- [7] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *PAMI*, 30(8):1371–1384, 2008.
- [8] M. Grubinger. *Analysis and Evaluation of Visual Information Systems Performance*. PhD thesis, Victoria University, Melbourne, Australia, 2007.
- [9] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *ICCV*, 2009.
- [10] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *CVPR*, 2004.
- [11] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.
- [12] R. Jin, S. Wang, and Z.-H. Zhou. Learning a distance metric from multi-instance multi-label data. In *CVPR*, 2009.
- [13] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.
- [14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
- [15] J. Li and J. Wang. Real-time computerized annotation of pictures. *PAMI*, 30(6):985–1002, 2008.
- [16] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*, 42(2):218–228, 2009.
- [17] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, 2008.
- [18] T. Mei, Y. Wang, X. Hua, S. Gong, and S. Li. Coherent image annotation by learning semantic distance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [19] D. Metzler and R. Manmatha. An inference network approach to image retrieval. In *CIVR*, 2004.
- [20] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: Constraining the latent space. In *ACM Multimedia*, 2004.
- [21] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.
- [22] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *ACM SIGKDD*, 2004.
- [23] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *ECCV*, 2006.
- [24] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM SIGCHI*, 2004.
- [25] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical Dirichlet process model. In *Workshop on Multimedia Data Mining ACM SIGKDD*, 2008.
- [26] A. Yavlinsky, E. Schofield, and S. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *CIVR*, 2005.
- [27] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, pages 2126–2136, 2006.