Image Annotation Using Metric Learning in Semantic Neighbourhoods

Yashaswi Verma and C.V. Jawahar

International Institute of Information Technology, Hyderabad, India - 500032
yashaswi.verma@research.iiit.ac.in, jawahar@iiit.ac.in

Abstract. Automatic image annotation aims at predicting a set of textual labels for an image that describe its semantics. These are usually taken from an annotation vocabulary of few hundred labels. Because of the large vocabulary, there is a high variance in the number of images corresponding to different labels ("class-imbalance"). Additionally, due to the limitations of manual annotation, a significant number of available images are not annotated with all the relevant labels ("weak-labelling"). These two issues badly affect the performance of most of the existing image annotation models. In this work, we propose 2PKNN, a two-step variant of the classical K-nearest neighbour algorithm, that addresses these two issues in the image annotation task. The first step of 2PKNN uses "image-to-label" similarities, while the second step uses "image-to-image" similarities; thus combining the benefits of both. Since the performance of nearest-neighbour based methods greatly depends on how features are compared, we also propose a metric learning framework over 2PKNN that learns weights for multiple features as well as distances together. This is done in a large margin set-up by generalizing a well-known (single-label) classification metric learning algorithm for multi-label prediction. For scalability, we implement it by alternating between stochastic sub-gradient descent and projection steps.

Extensive experiments demonstrate that, though conceptually simple, 2PKNN alone performs comparable to the current state-of-the-art on three challenging image annotation datasets, and shows significant improvements after metric learning.

1 Introduction

Automatic image annotation is a labelling problem which has potential applications in image retrieval [5,11,16], image description [23], etc. Given an unseen image, the goal is to predict multiple textual labels describing that image. Recent outburst of multimedia content has raised the demand for auto-annotation methods, thus making it an active area of research [5,11,16,19,20]. Several methods have been proposed in the past for image auto-annotation which try to model image-to-image [5,11,16], image-to-label [10,20] and label-to-label [11,20] similarities. Our work falls under the category of the supervised annotation models such as [4,5,10,11,13,16,19,22] that work with large annotation vocabularies.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part III, LNCS 7574, pp. 836–849, 2012.

Among these, K-nearest neighbour (or KNN) based methods such as [5,11,16] have been found to give some of the best results despite their simplicity. The intuition is that "similar images share common labels" [11]. In most of these approaches, this similarity is determined only using image features. Though this can handle label-to-label dependencies to some extent, it fails to address the "class-imbalance" (large variations in the frequency of different labels) and "weak-labelling" (many available images are not annotated with all the relevant labels) problems that are prevalent in the popular datasets (Sec. 4.1) as well as real-world databases. E.g., in an experiment on the Corel 5K dataset, we found that for the 20% least frequent labels, JEC [11] achieves an F-score of 19.7%, whereas it gives reasonably good performance for the 20% most frequent labels with F-score being 50.6%.

As per our knowledge, no attempt has been made in the past that directly addresses these issues. An indirect attempt was made in TagProp [16] to address class-imbalance, which we discuss in Sec. 2. To address these issues in a nearest-neighbour set-up, we need to make sure that (i) for a given image, the (subset of) training images that are considered for label prediction should not have large differences in the frequency of different labels; and (ii) the comparison criteria between two images should make use of both image-to-label and image-to-image similarities (image-to-image similarities also capture label-to-label similarities in a nearest-neighbour scenario). With this motivation, we present a two-step KNN-based method. We call this *2-Pass K-Nearest Neighbour* (or 2PKNN) algorithm. For an image, we say that its few nearest neighbours from a given class constitute its *semantic neighbourhood*, and these neighbours are its *semantic neighbours*. For a particular class, these are the samples that are most related with a new image. Given an unseen image, in the first step of 2PKNN we identify its semantic neighbours corresponding to all the labels. Then in the second step, only these samples are used for label prediction. This relates with the idea of "bottom-up pruning" common in day-to-day scenarios such as buying a car, or selecting a cloth to wear; where first the potential candidates are short-listed based on quick analysis, and then another set of criteria is used for final selection.

It is well-known that the performance of KNN based methods largely depends on how two images are compared [11]. Usually, this comparison is done using a set of features extracted from images and some specialized distance metric for each feature (such as L_1 for colour histograms, L_2 for Gist) [11,16]. In such a scenario, (i) since each base distance contributes differently, we can learn appropriate weights to combine them in the *distance space* [11,16]; and (ii) since every feature (such as SIFT or colour histogram) itself is represented as a multi-dimensional vector, its individual elements can also be weighted in the *feature space* [12]. As the 2PKNN algorithm works in the nearest-neighbour setting, we would like to learn weights that maximize the annotation performance. With this goal, we perform metric learning *over* 2PKNN by generalizing the LMNN [12] algorithm for multi-label prediction. Our metric learning framework extends LMNN in two major ways: (i) LMNN is meant for single-label classification (or simply classification) problems, while we adapt it for images annotation which

is a multi-label classification task; and (ii) LMNN learns a single Mahalanobis metric in the feature space, while we extend it to learn linear metrics for multiple features as well as distances together. Since we need to learn large number of weights, and iteratively perform pair-wise comparisons between large sample sets, scalability appears as one of the major concerns. To address this, we implement metric learning by alternating between stochastic sub-gradient descent and projection steps (similar to Pegasos [9]). This allows to optimize the weights iteratively using small number of comparisons at each iteration, thus making our learning easily scalable for large datasets with samples represented in very high dimensions. In our experiments on three benchmark image annotation datasets, our method (i.e., 2PKNN with metric learning) significantly outperforms the previous results.

In the next section, we discuss some of the notable contributions in the field of image annotation. In Sec. 3, we formalize 2PKNN and the metric learning model; in Sec. 4, we discuss the experiments; and finally conclude in Sec. 5.

2 Related Work

The image annotation problem was initially addressed using generative models; e.g. translation models [2,3] and nearest-neighbour based relevance models [4,5]. Recently, a Markov Random Field [13] based approach was proposed that can flexibly accommodate most of the previous generative models. Though these methods are directly extendable to large datasets, they might not be ideal for the annotation task as their underlying joint distribution formulations assume independence of image features and labels, whereas recent developments [17] emphasize on using conditional dependence to achieve Bayes optimal prediction.

Among discriminative models, SML [10] treats each label as a class of a multi-class multi-labelling problem, and learns class-specific distributions. However, it requires large (class-)balanced training data to estimate these distributions. Also label interdependencies might result into corrupted distribution models. Another nearest-neighbour based method [19] tries to benefit from feature sparsity and clustering properties using a regularization based algorithm for feature selection. JEC [11] treats image annotation as retrieval. Using multiple global features, a greedy algorithm is used for label transfer from neighbours. They also performed metric learning in the distance space but it could not do any better than using equal weights. This is because they used a classification-based metric learning approach for the annotation task which is multi-label classification by nature. Though JEC looks simple at the modelling level, it reported the best results on benchmark annotation datasets when it was proposed. TagProp [16] is a weighted KNN based method that transfers labels by taking a weighted average of keywords' presence among the neighbours. To address the class-imbalance problem, logistic discriminant models are wrapped over the weighted KNN method with metric learning. This boosts the importance given to infrequent labels and suppresses it for frequent labels appearing among the neighbours.

Nearly all image annotation models can be considered as multi-label classification algorithms, as they associate multiple labels with an image. Recent

methods such as [14,21] treat it as a multi-label ranking problem. Given an image, instead of predicting some fixed number of labels, they generate a ranked list of *all* the labels based on their chances of getting assigned to that image. E.g., in [21] an algorithm was proposed to learn from incompletely labelled data; i.e., only a subset of the ground-truth labels of each training image is used at the time of learning. Of late, some annotations methods such as [15,20] have been proposed that try to model image features and labels as well as dependencies among them, but most of these work on small vocabularies containing few tens of labels, and hence class-imbalance and weak-labelling are not a big concern. Our work is more comprehensive and falls under the category of previous works on image annotation [5,10,11,13,16,19,22] that address a more realistic and challenging scenario where the vocabulary contains few hundreds of labels and the datasets seriously suffer from class-imbalance and weak-labelling issues.

3 Label Prediction Model

Here, first we describe the 2PKNN algorithm, and then formulate the metric learning over it.

3.1 The 2PKNN Algorithm

Let $\{I_1, \ldots, I_t\}$ be a collection of images and $\mathcal{Y} = \{y_1, \ldots, y_l\}$ be a vocabulary of l labels (or semantic concepts). The training set $\mathcal{T} = \{(I_1, Y_1), \ldots, (I_t, Y_t)\}$ consists of pairs of images and their corresponding label sets, with each $Y_i \subseteq \mathcal{Y}$. Similar to SML [10], we assume the conditional probabilities $P(A|y_i)$ that model the feature distribution of an image A given a semantic concept $y_i \in \mathcal{Y}$. Using this, we model image annotation as a problem of finding the posterior probabilities

$$P(y_i|A) = \frac{P(A|y_i)P(y_i)}{P(A)} \ , \tag{1}$$

where $P(y_i)$ is the prior probability of the label y_i. Then, given an unannotated image J, the best label for it will be given by

$$y^* = \arg\max_i P(y_i|J) \ . \tag{2}$$

Let $\mathcal{T}_i \subseteq \mathcal{T}$, $\forall i \in \{1, \ldots, l\}$ be the subset of training data that contains *all* the images annotated with the label y_i. Since each set $\mathcal{T}_i$ contains images with one semantic concept (or label) common among them, we consider it as a *semantic group* (similar to [20]). It should be noted that the sets $\mathcal{T}_i$'s are not disjoint, as an image usually has multiple labels and hence belongs to multipe semantic groups. Given an unannotated image J, from each semantic group we pick K_1 images that are most similar to J and form corresponding sets $\mathcal{T}_{J,i} \subseteq \mathcal{T}_i$. Thus, each $\mathcal{T}_{J,i}$ contains those images that are *most informative* in predicting the probability of the label y_i for J. The samples in each set $\mathcal{T}_{J,i}$ are the semantic neighbours of J corresponding to y_i. These semantic neighbours incorporate image-to-label

similarity. Once $\mathcal{T}_{J,i}$'s are determined, we merge them all to form a set $\mathcal{T}_J = \{\mathcal{T}_{J,1} \bigcup \ldots \bigcup \mathcal{T}_{J,l}\}$. This way, we obtain a subset of the training data $\mathcal{T}_J \subseteq \mathcal{T}$ *specific to* J that contains its semantic neighbours corresponding to all the labels in the vocabulary $\mathcal{Y}$. This is the *first pass* of 2PKNN. It can be easily noted that in $\mathcal{T}_J$, each label appears (at least) K_1 times, thus addressing the class-imbalance issue. To understand how this step also handles the issue of weak-labelling, we analyze the cause of this. Weak-labelling occurs because "obvious" labels are often missed by human annotators while building a dataset, and hence many images depicting any such concept are actually not annotated with it. Under this situation, given an unseen image, if we use only its *few* nearest neighbours from the *entire* training data (as in [11,16]), then such labels may not appear among these neighbours and hence not get apt scores. In contrary, the first pass of 2PKNN finds a neighbourhood where all the labels are present explicitly. Therefore, now they (i.e., the "obvious" labels) have better chances of getting assigned to a new image, thus addressing the weak-labelling issue.

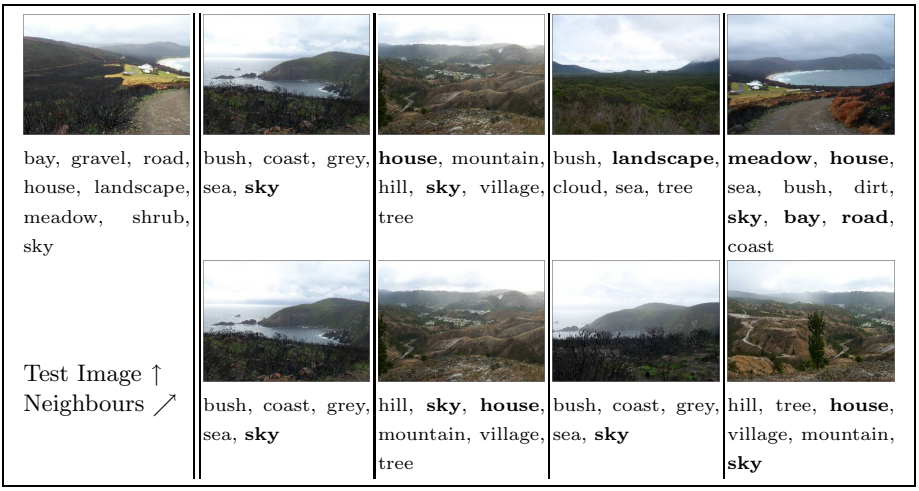

Fig. 1. For a test image from the IAPR TC-12 dataset (first column), the first row on the right section shows its 4 nearest images (and their labels) from the training data after the first pass of 2PKNN ($K_1 = 1$), and the second row shows the 4 nearest images using JEC [11]. The labels in bold are the ones that match with the ground-truth labels of the test image. Note the frequency (9 vs. 6) and diversity ({sky, house, landscape, bay, road, meadow} vs. {sky, house}) of matching labels for 2PKNN vs. JEC.

Figure 1 shows an example from the IAPR TC-12 dataset (Sec. 4.1) illustrating how the first pass of 2PKNN addresses both class-imbalance and weak-labelling. For a given test image (first column) along with its ground-truth labels, we can notice the presence of rare labels {"landscape", "bay", "road", "meadow"}

among its four nearest images found after the first pass of 2PKNN (first row on the right), without compromising with the frequent labels {"sky", "house"}. In contrary, the neighbours obtained using JEC [11] (second row) contain only the frequent labels. It can also be observed that though the labels {"landscape", "meadow"} look obvious for the neighbours found using JEC, these are actually absent in their ground-truth annotations (weak-labelling), whereas the first pass of 2PKNN explicits their presence among the neighbours.

The *second pass* of 2PKNN is a weighted sum over the samples in $\mathcal{T}_J$ to assign importance to labels based on image similarity. This gives the posterior probability for J given a label $y_k \in \mathcal{Y}$ as

$$P(J|y_k) = \sum_{(I_i,Y_i)\in\mathcal{T}_J} \theta_{J,I_i}.P(y_k|I_i) = \sum_{(I_i,Y_i)\in\mathcal{T}_J} exp(-D(J,I_i)).\delta(y_k \in Y_i), \quad (3)$$

where $\theta_{J,I_i} = exp(-D(J,I_i))$ (see Eq. (4) for the definition of $D(J,I_i)$) denotes the contribution of image I_i in predicting the label y_k for J depending on their visual similarity; and $P(y_k|I_i) = \delta(y_k \in Y_i)$ denotes the presence/absence of label y_k in the label set Y_i of I_i, with $\delta(\cdot)$ being 1 only when the argument holds true and 0 otherwise. Assuming that the first pass of 2PKNN will give a subset of the training data where each label has comparable frequency, we set the prior probability in Eq. (1) as a uniform distribution; i.e., $P(y_i) = \frac{1}{|\mathcal{T}|}$, $\forall i \in \{1,\ldots,l\}$. Putting Eq. (3) in Eq. (1) generates a ranking of all the labels based on their probability of getting assinged to the unseen image J. Note that along with image-to-image similarities, the second pass of 2PKNN implicitly takes care of label-to-label dependencies, as the labels appearing together in the same neighbouring image will get equal importance.

Both conceptually as well practically, 2PKNN is entirely different from the previous two-step variants of KNN such as [1,7]. They use few (global) nearest neighbours of a sample to apply some other more sophisticated technique such as linear discriminant analysis [1] or SVM [7]. Whereas, the first pass of 2PKNN considers all the samples but in *localized* semantic groups. Also, the previous variants were designed for the classification task, while 2PKNN addresses the more challenging problem of image annotation (which is multi-label classification by nature), where the datasets suffer from high class-imbalance and weak-labelling (note that weak-labelling is not a concern in classification problems).

3.2 Metric Learning (ML)

Most of the classification metric learning algorithms try to increase inter-class and reduce intra-class distances, thus treating each pair of samples in a binary manner (recall that similar approach was used in JEC [11] but could not improve the annotation performance). Since image annotation is a multi-label classification task, here two samples relate in the continuous space $[0,1]$; hence classification metric learning cannot be applied directly. As part of metric learning, our aim is to learn (non-negative) weights over multiple features as well as base distances that maximize the annotation performance for 2PKNN. For this

purpose, we extend the classical LMNN [12] algorithm for multi-label prediction. Let there be two images A and B, each represented by n features $\{\mathbf{f_A^1}, \ldots, \mathbf{f_A^n}\}$ and $\{\mathbf{f_B^1}, \ldots, \mathbf{f_B^n}\}$ respectively. Each feature is a multi-dimensional vector, with the dimensionality of a feature $\mathbf{f^i}$ being $\mathcal{N}_i$, i.e. $\mathbf{f^i} \in \mathcal{R}^{\mathcal{N}_i}$ for $i = 1, \ldots, n$. We denote an entry of a vector $\mathbf{x}$ as $\mathbf{x}(\cdot)$. The distance between two images is computed by finding the distance between their corresponding features using some specialized distance measure for each feature (such as L_1 for colour histograms, χ^2 for SIFT features, etc.), and then combining them all. In order to learn weights in the feature space, it should be noted that some of the popular distance measures such L_1, (squared) L_2 and χ^2 can be written as a dot product of two vectors. E.g., given any two corresponding feature vectors $\mathbf{f_A^i}$ and $\mathbf{f_B^i}$, if we consider a vector $\mathbf{d_{AB}^i} \in \mathcal{R}_+^{\mathcal{N}_i}$ such that $\mathbf{d_{AB}^i}(j) = |\mathbf{f_A^i}(j) - \mathbf{f_B^i}(j)|$, $\forall j \in \{1, \ldots, \mathcal{N}_i\}$, then the L_1 distance between the two feature vectors can be written as $L_1(\mathbf{f_A^i}, \mathbf{f_B^i}) = \mathbf{v^i} \cdot \mathbf{d_{AB}^i}$; where $|\cdot|$ gives the absolute value, and $\mathbf{v^i} \in \mathcal{R}_+^{\mathcal{N}_i}$ is usually taken as a normalized unit vector[1]. Note that $\mathbf{v^i}$ can replaced by *any* non-negative real-valued normalized vector that assigns appropriate weights to *individual* dimensions of a feature vector in the *feature space*. Moreover, we can also learn weights $\mathbf{w} \in \mathcal{R}_+^n$ in the *distance space* to optimally combine multiple feature distances. Based on this, we write the distance between A and B as

$$D(A, B) = \sum_{i=1}^{n} \mathbf{w}(i). \sum_{j=1}^{\mathcal{N}_i} \mathbf{v^i}(j). \mathbf{d_{AB}^i}(j) \ . \tag{4}$$

Now we describe how to learn the weights appearing in Eq. (4). For a given labelled sample $(Ip, Yp) \in \mathcal{T}$, we define its (i) *target* neighbours as its K_1 nearest images from the semantic group $\mathcal{T}_q$, $\forall q$ s.t. $y_q \in Y_p$, and (ii) *impostors* as its K_1 nearest images from $\mathcal{T}_r$, $\forall r$ s.t. $y_r \in \mathcal{Y} \setminus Y_p$. Our objective is learn the weights such that the distance of a sample from its target neighbours is minimized, and is also less than its distance from any of the impostors (i.e., *pull* the target neighbours and *push* the impostors). In other words, given an image I_p along with its labels Y_p, we want to learn weights such that its nearest (K_1) semantic neighbours from the semantic groups $\mathcal{T}_q$'s (i.e., the groups corresponding to its ground-truth labels) are pulled closer, and those from the remaining semantic groups are pushed far. With this goal, for sample image I_p, its target neighbour I_q and its impostor I_r, the loss function will be given by

$$E_{loss} = \sum_{pq} \eta_{pq} D(I_p, I_q) + \mu \sum_{pqr} \eta_{pq}(1 - \lambda_{pr})[1 + D(I_p, I_q) - D(I_p, I_r)]_+ \ . \tag{5}$$

Here, $\mu > 0$ handles the trade-off between the two error terms. The variable η_{pq} is 1 if I_q is a target neighbour of I_p and 0 otherwise. $\lambda_{pr} = \frac{|Y_p \cap Y_r|}{|Y_r|} \in [0, 1]$, with Y_r being the label set of an impostor I_r of I_p. And $[z]_+ = max(0, z)$ is the hinge loss which will be positive only when $D(I_p, I_r) < D(I_p, I_q) + 1$ (i.e., when for a sample I_p, its impostor I_r is nearer than its target neighbour I_q). To make sure

[1] $\mathbf{d_{AB}^i}$ can similarly be computed for other measures such as squared L_2 and χ^2.

that a target neighbour I_q is much closer than an impostor I_r, a margin (of size 1) is used in the error function. Note that λ_{pr} is in the continuous range $[0, 1]$, thus scaling the hinge loss depending on the overlap between the label sets of a given image I_p and its impostor I_r. This means that for a given sample, the amount of push applied on its impostor varies depending on its conceptual similarity with that sample. An impostor with large similarity will be pushed less, whereas an impostor with small similarity will be pushed more. This makes it suitable for multi-label classification tasks such as image annotation. The above loss function is minimized by the following constrained optimization problem:

$$\begin{aligned}
&\min_{\mathbf{w},\mathbf{v}} \textstyle\sum_{pq} \eta_{pq} D(I_p, I_q) + \mu \sum_{pqr} \eta_{pq}(1 - \lambda_{pr})\xi_{pqr} \\
&s.t.\ \ D(I_p, I_r) - D(I_p, I_q) \geq 1 - \xi_{pqr} \quad \forall p, q, r \\
&\qquad \xi_{pqr} \quad \geq 0 \quad \forall p, q, r \\
&\qquad \mathbf{w}(i) \ \geq 0 \quad \forall i;\ \ \textstyle\sum_{i=1}^{n} \mathbf{w}(i) = n \\
&\qquad \mathbf{v^i}(j) \geq 0 \quad \forall i, j;\ \ \textstyle\sum_{j=1}^{\mathcal{N}_i} \mathbf{v^i}(j) = 1 \quad \forall i \in \{1, \ldots, n\}
\end{aligned} \tag{6}$$

Here, the slack variables ξ_{pqr} represent the hinge loss in Eq. (5), and $\mathbf{v}$ is a vector obtained by concatenating all the $\mathbf{v^i}$'s for $i = 1, \ldots, n$. We solve the above optimization problem in the primal form itself. Since image features are usually in very high dimensions, the number of variables is large ($= n + \sum_{i=1}^{n} \mathcal{N}_i$). This makes the scalability of the above optimization problem difficult using conventional gradient descent. To overcome this issue, we solve it by alternatively using stochastic sub-gradient descent and projection steps (similar to Pegasos [9]). This gives an approximate optimal solution using small number of comparisons, and thus helps in achieving high scalability. To determine the optimal weights, we alternate between the weights in distance space and feature space.

Our extension of LMNN conceptually differs from its previous extensions such as [18] in at least two significant ways: (i) we adapt LMNN in its choice of target/impostors to learn metrics for multi-label prediction problems, whereas [18] uses the same definition of target/impostors as in LMNN to address classification problem in multi-task setting, and (ii) in our formulation, the amount of push applied on an impostor varies depending on its conceptual similarity w.r.t. a given sample, which makes it suitable for multi-label prediction tasks.

4 Experiments

4.1 Data Sets and Their Characteristics

We have used three popular image annotation datasets Corel 5K, ESP Game and IAPR TC-12 to test and compare the performance of our method with previous approaches. Corel 5K was first used in [3], and since then it has become a benchmark for comparing annotation performance. ESP Game contains images annotated using an on-line game, where two (mutually unknown) players are randomly given an image for which they have to predict same keyword(s) to

score points [6]. This way, many people participate in the manual annotation task thus making this dataset very challenging and diverse. IAPR TC-12 was introduced in [8] for cross-lingual retrieval. In this, each image has a detailed description from which only nouns are extracted and treated as annotations.

In Table 1, columns $2-5$ show the general statistics of the three datasets; and in columns $6-8$, we highlight some interesting statistics that provide better insights about the structure of the three datasets. It can be noticed that for each dataset, around 75% of the labels have frequency less than the mean label frequency (column 8), and also the median label frequency is far less than the corresponding mean frequency (column 7). This verifies the claim we previously made in Sec. 1 (i.e., datasets badly suffer from the class-imbalance problem).

Table 1. General (columns 2-5) and some insightful (columns 6-8) statistics for the three datasets. In column 6 and 7, the entries are in the format "mean, median, maximum". Column 8 ("Labels#") shows the number of labels whose frequency is less than the mean label frequency.

Dataset	Number of images	Number of labels	Training images	Testing images	Labels per image	Images per label (or label frequency)	Labels#
Corel 5K	5,000	260	4,500	500	3.4, 4, 5	58.6, 22, 1004	195 (75.0%)
ESP Game	20,770	268	18,689	2,081	4.7, 5, 15	326.7, 172, 4553	201 (75.0%)
IAPR TC-12	19,627	291	17,665	1,962	5.7, 5, 23	347.7, 153, 4999	217 (74.6%)

Though it is not straightforward to quantify weak-labelling, we try to analyze it from the number of labels per image (column 6). We argue that large gap between mean (or median) and maximum number of labels per image indicates that many images are not labelled with all the relevant labels. Based on this, we can infer that both ESP Game and IAPR TC-12 datasets suffer from weak-labelling. For Corel 5K dataset, we examined the images and their corresponding annotations to realize weak-labelling.

4.2 Features and Evaluation Measures

Features. To compare our model's performance with the previous methods, we use the similar features as in [16]. These are a combination of local and global features. The local features include the SIFT and hue descriptors obtained densely from multi-scale grid, and from Harris-Laplacian interest points. The global features comprise of histograms in RGB, HSV and LAB colour spaces, and the Gist features. To encode some spatial information about an image, all but the Gist features are also computed over three equal horizontal partitions for each image. To calculate distance between two features, L_1 measure is used for the colour histograms, L_2 for the Gist, and χ^2 for the SIFT and hue descriptors.

Evaluation Measures. To analyze the annotation performance, we compute precision and recall of each label in a dataset. Suppose a label y_i is present in the

ground-truth of m_1 images, and it is predicted for m_2 images during testing out of which m_3 predictions are correct ($m_3 \leq m_2$ and $m_3 \leq m_1$), then its precision will be $= m_3/m_2$ and recall will be $= m_3/m_1$. We average these values over all the labels of a dataset and get percentage mean precision P and percentage mean recall R, similar to the previous annotation methods such as [11,16,19,22]. Using these two measures, we get the percentage F1-score F1 = 2.P.R/(P+R), which takes care of the trade-off between precision and recall. The different image annotation methods are compared using two criteria: first F1; and second N+ which is the number of labels that are correctly assigned to at least one test image (i.e., the number of labels with positive recall).

To compare with [14], we use three measures (see [14] for more details): (i) "One-error" is similar to classification error, which tells the number of times the label predicted with the highest probability is not present in the ground-truth, (ii) "Coverage" is a measure of the worst rank assigned to any of the ground-truth labels, and (iii) "Average Precision" (not to be confused with percentage mean precision P used to measure annotation performance) gives area under precision-recall curve used to evaluate a ranked list of labels. To compare with [21], we adopt Area Under ROC curve (or AUC) as the evaluation measure.

For One-error and Coverage, smaller value implies better performance, while for all other measures, higher value implies better performance.

4.3 Experimental Details

A randomly-sampled subset of training data consisting of 3000 samples is used to learn the weights $\mathbf{w}$ and $\mathbf{v^i}$'s in leave-one-out manner, and a separate (random) validation set of 400 samples is used for early-stopping. This is repeated five times, and the model that performs best during validation is used to evaluate the performance on test data. For the first pass of 2PKNN, we set K_1 as 4, 2 and 2 for the Corel 5K, ESP Game and IAPR TC-12 datasets respectively. The results corresponding to 2PKNN are determined using an L_1-normalized one-vector for each $\mathbf{v^i}$; & the vector $\mathbf{w}$ is replaced by a scalar that controls the decay of θ_{J,I_i} (Eq. (3)) with distance (similar to [16]). The results for 2PKNN+ML are obtained by using weighted features and base distances in 2PKNN, with weights being determined after metric learning. The probability scores are appropriately scaled such that the relevance of a label for any image is never above one.

4.4 Comparisons

Now we present the quantitive analysis on the three datasets. First, for the sake of completeness, we compare with recent multi-label ranking methods [14,21], and then we show detailed comparisons with the previous annotation methods.

Comparison with Multi-label Ranking Methods. In order to show that our method does not compromise with the performance on frequent labels, we quantitatively compare our method with the best reported results of MIML [14]

in Table 2 using the same conventions as theirs. To be specific, we use the Corel 5K dataset and consider only the 20 most frequent labels, which results in $3,947$ training and 444 test images with average 1.8 labels per image. As we can see, our method performs significantly better than [14] on all the three measures. Notable is the considerable reduction in the coverage score, which indicates that in most of the cases, all the ground-truth annotations are included among the top 4 labels predicted by our method.

Table 2. Comparison between MIML [14] and 2PKNN combined with metric learning on a subset of the Corel 5K dataset using only 20 most frequent labels as in [14]

	One-error	Coverage	Average Precision
MIML [14]	0.565	5.507	0.535
2PKNN+ML (This work)	**0.427**	**3.38**	**0.644**

To show that our method addresses weak-labelling issue, we compare with [21]. Though [21] addresses the problem of incomplete labelling (see Sec. 2), conceptually it overlaps with the weak-labelling issue as both work under the scenario of inavailability of all relevant labels. Following their protocol, we select those images from the entire ESP Game dataset that are annotated with at least 5 labels, and then test on four cases. First we use all the labels in the ground-truth, and then randomly remove 20%, 40% and 60% labels respectively from the ground-truth annotation of each training image. On an average, they achieved 83.7475% AUC for these cases, while we get 85.4739% which is better by 1.7264%.

Comparison with Image Annotation Methods. To compare the image annoation performance, we follow the pre-defined partitions for training and testing as used in [11,16]. To each test image, we assign the top five labels predicted using Eq. (2). Our results as well as those reported by the previous models for image annotation are summarized in Table 3. We can see that on all the three datasets, our base method 2PKNN itself performs comparable to the previous best results. Notable is the significant increase in the number of labels with positive recall (credit goes to the first pass of 2PKNN). After combining metric learning with it (2PKNN+ML), the performance significantly improves. Precisely, for the Corel 5K, ESP Game and IAPR TC-12 datasets, we gain 6.7%, 3.8%, and 4.1% respectively in term of F1; and 19, 13 and 12 respectively in terms of N+ over the current state-of-the-art. We also found that F1 improves by upto 2% in general on using both $\mathbf{w}$ and $\mathbf{v^i}$'s as compared to using any one of them.

These empirical evaluations conclude that our method consistently shows superior performance than the previous models under multiple evaluation criteria, thus establishing its overall effectiveness.

Table 3. Comparison of annotation performance among different methods. The top section shows the performances reported by the previous methods. The bottom section shows the performance achieved by our method (2PKNN), and that combined with metric learning (2PKNN+ML). The best results in both parts are highlighted in bold.

	Corel 5K				ESP Game				IAPR TC-12			
Method	P	R	F1	N+	P	R	F1	N+	P	R	F1	N+
CRM [4]	16	19	17.4	107	–	–	–	–	–	–	–	–
MBRM [5]	24	25	24.5	122	18	19	18.5	209	24	23	23.5	223
SML [10]	23	29	25.7	137	–	–	–	–	–	–	–	–
JEC [11]	27	32	29.3	139	22	25	23.4	224	28	29	28.5	250
GS [19]	30	33	31.4	146	–	–	–	–	32	29	30.4	252
MRFA [13]	31	36	33.3	**172**	–	–	–	–	–	–	–	–
CCD (SVRMKL+KPCA) [22]	**36**	41	**38.3**	159	36	24	28.8	232	44	29	35.0	251
TagProp(ML) [16]	31	37	33.7	146	**49**	20	28.4	213	**48**	25	32.9	227
TagProp(σML) [16]	33	**42**	37.0	160	39	**27**	**31.9**	**239**	46	**35**	**39.8**	**266**
2PKNN (This work)	39	40	39.5	177	51	23	31.7	245	49	32	38.7	274
2PKNN+ML (This work)	**44**	**46**	**45.0**	**191**	**53**	**27**	**35.7**	**252**	**54**	**37**	**43.9**	**278**

4.5 Discussion

In Figure 2, we analyze how 2PKNN addresses the class-imbalance issue as compared to the traditional weighted KNN method (used in TagProp [16]). For this purpose, we use the annotation performance in terms of mean recall. The labels are partitioned into two groups based on their frequency. The first partition consists of the 50% least frequent labels and the second partition consists of the 50% most frequent labels. Three observations can be made by looking at this figure. First, for all the three datasets, unlike weighted KNN, 2PKNN performs comparable for both the label-partitions despite the large differences in their frequency (compare the median label frequency with mean and maximum label frequency in Table 1, column 7). This suggests that 2PKNN actually addresses the class-imbalance problem in the challenging datasets with large vocabularies. Second, 2PKNN does *not* compromise with the performance on frequent labels compared to weighted KNN, and always performs better than it. This shows that 2PKNN can be a better option than weighted KNN for the complicated image annotation task. And third, after metric learning, the performance always improves for both the label-partitions for all the datasets. This confirms that our metric learning approach benefits both rare as well as frequent labels. In Figure 3, we show some qualitative annotation results from the three datasets. Each image is an example of a weakly-labelled image. It can be seen that for all these images, our method predicts all the ground-truth labels. Moreover, the additional labels predicted are actually depicted in the corresponding images, but missing in their ground-truth annotations. Recall that for quantitative analysis, we experimentally compared our method with [21] (Sec. 4.4) and achieved better performance. These show that our method is capable of addressing the weak-labelling issue prevalent in the real-world datasets.

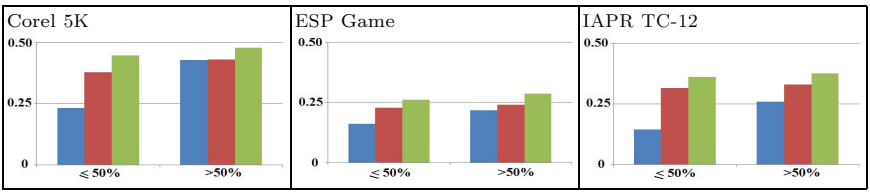

Fig. 2. Annotation performance in terms of mean recall (vertical axis) for the three datasets obtained using weighted KNN as in TagProp [16] (blue), using 2PKNN (red), and 2PKNN combined with metric learning (green). The labels are grouped based on their frequency in a dataset (horizontal axis). The first bin corresponds to the 50% least frequent labels and the second bin corresponds to the 50% most frequent labels.

Fig. 3. Annotations for example images from the three datasets. The second row shows the ground-truth annotations and the third row shows the labels predicted using our method 2PKNN+ML. The labels in **blue** (bold) are those that match with ground-truth. The labels in *red* (italics) are those that, though depicted in the corresponding images, are missing in their ground-truth annotations and are predicted by our method.

5 Conclusion

We showed that our variant of the KNN algorithm, i.e. 2PKNN, combined with metric learning performs better than the previous methods on three challenging image annotation datasets. It also addresses class-imbalance and weak-labelling issues that are prevalent in the real-world scenarios. This can be useful for natural image databases where tags obey the *Zipf's law*. We also demonstrated how a classification metric learning algorithm can be effectively adapted for the more complicated multi-label classification problems such as annotation. We hope that this work throws some light on the possibilites of extending the popular discriminative margin-based classification methods for the multi-label prediction tasks.

References

1. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. PAMI 18(6), 607–616 (1996)
2. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: First International Workshop on Multimedia Intelligent Storage and Retrieval Management (1999)
3. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
4. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: NIPS (2003)
5. Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple Bernoulli relevance models for image and video annotation. In: CVPR, pp. 1002–1009 (2004)
6. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: ACM SIGCHI (2004)
7. Zhang, H., Berg, A.C., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006)
8. Grubinger, M.: Analysis and Evaluation of Visual Information Systems Performance. PhD thesis, Victoria University, Melbourne, Australia (2007)
9. Shwartz, S.S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient solver for SVM. In: ICML (2007)
10. Carneiro, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. PAMI 29(3), 394–410 (2007)
11. Makadia, A., Pavlovic, V., Kumar, S.: A New Baseline for Image Annotation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 316–329. Springer, Heidelberg (2008)
12. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. JMLR 10(2), 207–244 (2009)
13. Xiang, Y., Zhou, X., Chua, T.-S., Ngo, C.W.: A Revisit of generative models for automatic image annotation using markov random fields. In: CVPR (2009)
14. Jin, R., Wang, S., Zhou, Z.H.: Learning a distance metric from multi-instance multi-label data. In: CVPR, pp. 896–902 (2009)
15. Wang, H., Huang, H., Ding, C.: Image annotation using multi-label correlated Green's function. In: ICCV (2009)
16. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbour models for image auto-annotation. In: ICCV (2009)
17. Dembczyński, K., Cheng, W., Hüllermeier, E.: Bayes optimal multilabel classification via probabilistic classifier chains. In: ICML (2010)
18. Parameswaran, S., Weinberger, K.Q.: Large margin multi-task metric learning. In: NIPS (2010)
19. Zhang, S., Huang, J., Huang, Y., Yu, Y., Li, H., Metaxas, D.N.: Automatic image annotation using group sparsity. In: CVPR (2010)
20. Wang, H., Huang, H., Ding, C.: Image annotation using bi-relational graph of images and semantic labels. In: CVPR (2011)
21. Bucak, S.S., Jin, R., Jain, A.K.: Multi-label learning with incomplete class assignments. In: CVPR, pp. 2801–2808 (2011)
22. Nakayama, H.: Linear distance metric Learning for large-scale generic image recognition. PhD thesis, The University of Tokyo, Japan (2011)
23. Gupta, A., Verma, Y., Jawahar, C.V.: Choosing linguistics over vision to describe images. In: AAAI (2012)