Exploring SVM for Image Annotation in Presence of Confusing Labels

Yashaswi Verma
http://researchweb.iiit.ac.in/~yashaswi.verma/
C. V. Jawahar
http://www.iiit.ac.in/~jawahar/

CVIT
IIIT-Hyderabad
Hyderabad, India
http://cvit.iiit.ac.in

Abstract

We address the problem of automatic image annotation in large vocabulary datasets. In such datasets, for a given label, there could be several other labels that act as its *confusing labels*. Three possible factors for this are (i) incomplete-labeling (*"cars"* vs. *"vehicle"*), (ii) label-ambiguity (*"flowers"* vs. *"blooms"*), and (iii) structural-overlap (*"lion"* vs. *"tiger"*). While previous studies in this domain have mostly focused on nearest-neighbour based models, we show that even the conventional one-vs-rest SVM significantly outperforms several benchmark models. We also demonstrate that with a simple modification in the hinge-loss of SVM, it is possible to significantly improve its performance. In particular, we introduce a tolerance-parameter in the hinge-loss. This makes the new model more tolerant against the errors in the classification of samples tagged with confusing labels as compared to other samples. This tolerance parameter is automatically determined using visual similarity and dataset statistics. Experimental evaluations demonstrate that our method (referred to as SVM with Variable Tolerance or *SVM-VT*) shows promising results on the task of image annotation on three challenging datasets, and establishes a baseline for such models in this domain.

1 Introduction

Automatic image annotation is an interesting problem, where each image is associated with a set of labels and the goal is to learn a model that assigns multiple labels to a new image. This has applications in several tasks such as image retrieval, object recognition, robot navigation, etc. Hence this has emerged as an important research area during the last decade [2, 5, 6, 8, 11, 19, 23]. In annotation datasets with large vocabularies of few hundred or more labels, there exist three practical issues: (a) **Incomplete-labeling**: The training samples are not exhaustively tagged with *all* relevant labels from vocabulary. This is because while building a dataset, human annotators find some labels as "obvious" and miss them while preparing the ground-truth. E.g., an image tagged with *"car"* might not be tagged with *"vehicle"*. (b) **Label-ambiguity**: There are some labels that convey same semantic meaning and thus can be used interchangeably, due to which usually only one of them is assigned by annotator. E.g., an image tagged with *"flowers"* may not be tagged with *"blooms"* as both convey the same meaning. (c) **Structural-overlap**: There are some labels that, in spite of being different, *share* structural properties. E.g., though *"tiger"* and *"lion"* are two


cars, tracks, prototype


grass, ***flowers***, petals


sky, grass, plane, ***lion***


tree, grass, ***tiger***, park

Figure 1: Example images from the Corel-5k dataset [4] and corresponding ground-truth labels. First image is an example of **incomplete-labeling** (tagged with *"car"* but not with *"vehicle"*); second image is an example of **label-ambiguity** (tagged with *"flowers"*, though *"blooms"* would also have been equally correct); & third and fourth images are examples of **structural-overlap** (*"lion"* and *"tiger"* are two different but structurally related labels).

different labels, structurally they are very similar. Figure 1 shows such examples from Corel-5k dataset [4]. All these issues combinedly give rise to the existence of sets of *confusing labels* within a vocabulary. It is important to note that some of such confusing labels might actually be one of the positive labels for a given image, but remain missing in the ground-truth due to these issues. In other words, for a given label l_a, a confusing label l_b is a label that is/could-be used in-place-of/together-with l_a, due to: incompleteness, ambiguity or overlap problems. In this work, our goal is to learn from such data where for a given label, there could be several other labels in the vocabulary that act as its confusing labels. We shall refer this problem as ***"image annotation in presence of confusing labels"***.

Among the image annotation models being proposed in the past, generative or nearest-neighbour (NN)-based models [5, 8, 11, 19, 23] have particularly been shown to be successful for large vocabulary datasets such as Corel-5k [4], ESP Game [20] and IAPRTC-12 [7]. The reason behind this is that in NN-based models, given a sample, the labels that are not present in the ground-truth of its neighbouring samples are simply ignored, rather than being considered as negative. This makes such models somewhat *tolerant* against the issue of confusing labels. In contrary, simple one-vs-rest Support Vector Machine (or *SVM*) [3, 18] has remained almost unexplored in this domain. This might be due to its *strict* discriminative nature: it considers everything other than positive as *equally* negative. E.g., while learning a model for *"flowers"*, samples labeled with *"blooms"* are considered as negative examples. Due to this, for a given label, the samples that are either incompletely labeled, or tagged with another label that is semantically/structurally similar to the given label get confused as negative examples. This, in turn, inhibits learning good decision boundaries, and hence affects the performance of the learned SVM model.

In this work, we demonstrate that despite this strict discriminative behaviour, simple SVM itself can give superior performance than several benchmark image annotation models. Moreover, we claim that if it is made tolerant against confusing labels, then it is possible to achieve significant improvements in its performance. To support this, we propose an extension of the SVM model that (i) is more tolerant against the errors made in the classification of samples tagged with one or more confusing labels; and (ii) penalizes such errors by smaller amount as compared to those in other samples. This is performed by introducing a *tolerance-parameter* in the conventional SVM model that takes care of both these requirements. We call this model as *Suppor Vector Machine with Variable Tolerance* (or *SVM-VT*), and show that it can be as efficiently optimized as the standard SVM. Empirical studies on three popular image annotation datasets demonstrate that it achieves very promising results, thus establishing a baseline for such models in this domain.

2 The SVM-VT Model

Let $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a collection of m samples and $\mathcal{V} = \{l_1, ..., l_n\}$ be a vocabulary of n labels. The dataset $\mathcal{T} = \{(\mathbf{x}_1, L_1), \ldots, (\mathbf{x}_m, L_m)\}$ is a set of tuples of the form $(\mathbf{x}_i, L_i)$ where $\mathbf{x}_i$ is a sample and $L_i \subseteq \mathcal{V}$ is the set of its labels. Let S_i^+ be the set of samples that are annotated with the label l_i. We consider these samples as positive examples of l_i, and denote the remaining samples as $\bar{S}_i^+ = S \setminus S_i^+, \forall i \in \{1, \ldots, n\}$. From now onwards, we shall discuss considering a single label and omit the subscript index for brevity.

For a given label l, the conventional SVM considers the samples in S^+ as its positive examples and those in $\bar{S}^+$ as its negative examples. Using these two sets, a linear classifier $\mathbf{w}$ is learnt (separately for each label) by solving the following optimization problem:

$$\min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{j=1}^{m}[1 - y_j(\mathbf{w}\cdot\mathbf{x}_j)]_+, \tag{1}$$

where $[z]_+ = max(0, z)$ denotes the hinge-loss, $\lambda > 0$ is used to control the trade-off between regularization and loss, and $y_j = 1$ if $\mathbf{x}_j \in S^+$ and -1 otherwise. Solving this leads to finding a hyper-plane $\mathbf{w}$ that best separates the samples in S^+ and $\bar{S}^+$ with maximum margin.

In order to make SVM tolerant against the confusing samples, we define a new loss function based on the hinge-loss. It introduces a tolerance-parameter "t" that adjusts both the margin as well the gradient update-rule for each sample separately. Specifically, we formalize the SVM-VT model as that of solving the following optimization problem:

$$\min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{j=1}^{m}[1 - y_j t_j(\mathbf{w}\cdot\mathbf{x}_j)]_+, \tag{2}$$

where the additional parameter $t_j \in [0, 1]$ controls the tolerance against the errors made in the classification of sample $\mathbf{x}_j$. The hyperplane $\mathbf{w}$ is now learnt such that it is more strict towards correctly classifying samples with high value of t_j and any such error leads to a large shift in hyperplane. In other words, the hyperplane is more tolerant against errors made in classification of samples with low value of t_j and such errors lead to a small shift in hyperplane. If $t_j = 1\ \forall j$, it becomes exactly the same as that of the standard SVM as shown in equation 1. In this way, SVM-VT can be viewed as a (strict) generalization of SVM.

In Figure 2 (left), we show how the hinge-loss function varies with different values of t for some sample $\mathbf{x}$. On X-axis and Y-axis, we represent the value of $y(\mathbf{w}\cdot\mathbf{x})$ and that of the hinge-loss respectively. It can be seen that for small value of t, the hinge-loss remains small even for large misclassification errors. As we increase t, the hinge-loss becomes more and more sensitive towards misclassification errors and hence they get more penalized. Another interesting thing to notice is that as we reduce t, the hinge-loss fires even for the samples which are correctly classified with high confidence, though the loss value remains very small. This is desirable when we are confused about the exact label of a sample and want to penalize its highly confident correct classification. Also, if we set $t = 0$ for some particular sample, then the classifier becomes infinitely tolerant against the error made in its classification.

2.1 Determining the tolerance parameter

As discussed in section 1, for a given label l, there could exist several other labels in the vocabulary that act as its confusing labels. Due to this, there could be possibly many samples in the set $\bar{S}^+$ that are tagged with some confusing label of l (which we call as confusing

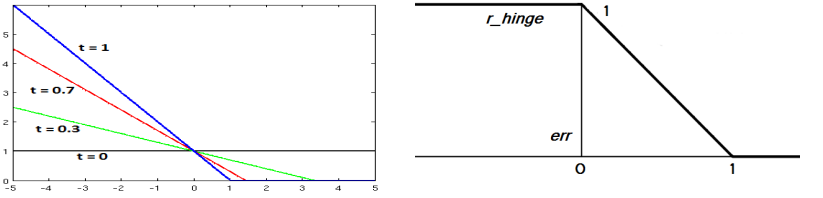

Figure 2: Loss function with variation in the tolerance-parameter. The horizontals axis represents the value of $y(\mathbf{w}\cdot\mathbf{x})$ and vertical axis represents the loss. On the left, the conventional hinge-loss corresponds to $t = 1$, and the proposed *t_hinge* varies with different values of $0 \leq t \leq 1$. On the right, the step-function corresponds to misclassification error (*err*) and the one in bold represents *r_hinge*.

samples). Though in practice it is possible to automatically learn the tolerance parameter in equation 2 using either non-convex optimization or convex-relaxation, it would not solve our purpose of identifying such samples. This is because doing so will look only at the features of the samples without considering other semantic properties. Here we propose a heuristic approach for determining the t-value for each sample given a label that tries to address the three issues discussed in section 1.

For a given label l, we consider three factors to determine the semantic relatedness of each sample $\mathbf{x}_j \in \bar{S}^+$ with that label:
(a) *Reverse nearest-neighbours based score*: For a fixed value of K $(= 5)$, let p_k be the number of samples in S^+ that have $\mathbf{x}_j$ as their k^{th} nearest neighbour. Then we define

$$score_1(\mathbf{x}_j|l) = \frac{\sum_{k=1}^{K}(\frac{p_k}{k})}{\sum_{k=1}^{K} p_k + \varepsilon} \tag{3}$$

where $\varepsilon > 0$ is a small number to avoid division by zero.
(b) *Visual similarity based score*: We compute the visual similarity score $sim(\cdot)$ (scaled into range $[0,1]$) of $\mathbf{x}_j$ with its nearest neighbour $\mathbf{x}_i^* \in S^+$ using JEC [11] method and define

$$score_2(\mathbf{x}_j|l) = sim(\mathbf{x}_j, \mathbf{x}_i^*) \tag{4}$$

(c) *Label co-occurrence based score*: Given a label l, let $\mathbf{y} \in \{0,1\}^m$ be such that its i^{th} entry is 1 if the i^{th} training image is tagged with l, and 0 otherwise. We compute co-occurrence score $co\_occur(l_i, l_j)$ between two labels l_i and l_j by computing cosine similarity between their corresponding vectors $\mathbf{y}_i$ and $\mathbf{y}_j$. Now, let $\mathbf{x}_j$ be tagged with labels L_j. We define

$$score_3(\mathbf{x}_j|l) = \max_{l_j \in L_j} co\_occur(l, l_j) \tag{5}$$

Intuitively, while $score_3$ tries to address incomplete-labeling, $score_1$ and $score_2$ try to address the issues of label-ambiguity and structural-overlap. Based on these three scores, we define the tolerance parameter for sample $\mathbf{x}_j$ given label l as

$$t_j = 1 - \frac{1}{3}(score_1(\mathbf{x}_j|l) + score_2(\mathbf{x}_j|l) + score_3(\mathbf{x}_j|l)) \tag{6}$$

From equation 6, it can be seen that for a given sample in $\bar{S}^+$, smaller tolerance value corresponds to higher chance of it being related to a given label and vice-versa. However,

since it is very difficult to claim if a negative sample is actually positive, we still consider $y_j = -1 \ \forall \mathbf{x}_j \in \bar{S}^+$. This is because our aim is to learn a classifier that is tolerant against confusing labels, rather than getting learnt on them. Also, we take $t_j = 1 \ \forall \mathbf{x}_j \in S^+$ assuming that all the positive samples are correctly annotated.

In Figure 3, we show the negative samples (along with their ground-truth labels) with least t-scores for two labels each from the three datasets. Several interesting observations can be made from these examples. All these negative samples actually look semantically related (*near positive*) with the corresponding labels. In first row, the negative samples for the label *"clouds"* demonstrate examples of incomplete-labeling, as *"clouds"* are clearly visible in these images but missing in their ground-truth. Similar is the case with the negative samples for the labels *"teeth"*, *"toy"*, *"horse"*, and *"bedcover"*. The first negative sample for the label *"man"* is an example of structural-overlap, because though *"man"* is not there in this images, it has *"woman"* that is structurally related with *"man"*. The second negative sample for the same label is an example of label-ambiguity since in this image *"people"* is used to refer the *"man"* climbing the tree. Thus, these examples verify the ability of our method in identifying so-called confusing samples.

2.2 Dual-form

By rewriting equation 2, the dual form of SVM-VT can be easily derived as below:

$$\min_{\mathbf{w},\xi\geq 0} \ \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{j=1}^{m}\xi_j \quad s.t. \ \ \xi_j \geq 1 - y_j t_j(\mathbf{w}\cdot\mathbf{x}_j) \ \ \forall j \in \{1,\ldots,m\} \tag{7}$$

$$= \max_{\alpha} \ \sum_{i=1}^{m}\alpha_i - \frac{1}{2\lambda}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j t_i t_j(\mathbf{x}_i\cdot\mathbf{x}_j) \quad s.t. \ \ 0 \leq \alpha_i \leq \frac{1}{m} \ \ \forall i \in \{1,\ldots,m\} \tag{8}$$

The dual of SVM-VT is very much similar to that of the conventional SVM, thus allowing efficient optimization.

2.3 Properties of SVM-VT

For a sample $\mathbf{x}_j$, let us denote the conventional hinge-loss and misclassification error by

$$hinge(\mathbf{w},\mathbf{x}_j,y_j) = [1 - (y_j(\mathbf{w}\cdot\mathbf{x}_j)]_+$$
$$err(\mathbf{w},\mathbf{x}_j,y_j) = \delta(y_j(\mathbf{w}\cdot\mathbf{x}_j) < 0)$$

where $\delta(\cdot)$ is 1 if the argument holds true and 0 otherwise. Then it can be easily shown that the hinge-loss provides an upper-bound on the misclassification error. Now, let us denote the modified hinge-loss of SVM-VT as

$$t\_hinge(\mathbf{w},\mathbf{x}_j,y_j,t_j) = [1 - y_j t_j(\mathbf{w}\cdot\mathbf{x}_j)]_+$$

Proposition 1. *For $t_j \in [0,1]$, $t\_hinge(\mathbf{w},\mathbf{x}_j,y_j,t_j) \geq err(\mathbf{w},\mathbf{x}_j,y_j)$, i.e. t_hinge provides an upper-bound on the misclassification error.*

Also, the robust hinge-loss is ususally considered as more stable than the conventional hinge-loss [10, 13]. It is given by:

$$r\_hinge(\mathbf{w},\mathbf{x}_j,y_j) = min(1, hinge(\mathbf{w},\mathbf{x}_j,y_j))$$

However, since this is a non-convex function, it is difficult to optimize. From Figure 2, it can be easily seen that *t_hinge* provides an upper-bound on *r_hinge* for $t \in [0,1]$.

Proposition 2. *For $t_j \in [0,1]$, $t\_hinge(\mathbf{w},\mathbf{x}_j,y_j,t_j) \geq r\_hinge(\mathbf{w},\mathbf{x}_j,y_j) \geq err(\mathbf{w},\mathbf{x}_j,y_j)$.*

Figure 3: For example labels (in blue) from the three datasets, the top "negative" samples that have least t-scores and corresponding ground-truth labels (for a given label, smaller t-score of a negative sample implies higher semantic relevance with that label).

2.4 Comparison with Other Methods

Several extensions of SVM have been proposed in the past that try to modify the loss-function. Here we give a brief overview of some of these methods whose formulation *looks* similar to that of SVM-VT, and discuss how SVM-VT differs from them. In [14], a separate scaling parameter is used for hinge-loss of positive and negative classes. This is generalized in [22] where hinge-loss corresponding to each sample is scaled individually by a parameter in the range $[0, 1]$. In [16], the loss is made sensitive to the distance of a sample from class-centroid. One similarity among all these methods is that they try to learn a classifier that is robust against outliers, by looking *only* at the features of samples. SVM-VT differs from these models in at least two ways. First, while these methods modify either the margin constraint [16] or the (classifier) update-rule [14, 22] of the conventiional SVM, the proposed hinge-loss of SVM-VT modifies both of these simultaneously. Second, all these methods consider only the distribution of samples in feature space, whereas the hinge-loss of SVM-VT has an associated semantic meaning that relates samples and labels based on semantic properties in addition to visual features.

For the image annotation task, Structured SVM (or *SSVM*) [17] seems to be an appropriate model. Intuitively, the idea behind SSVM is to benefit from the structure in the output space. Through SVM-VT we have tried to infuse this idea in the SVM model, though indirectly. This is because while learning the classifier for a given label, the amount of penalty for each non-positive sample differs depending on how much confusion it introduces in the classifier training. A sample that is more confusing for a given label adds a smaller penalty as

Dataset	Labels	No. of Training Img.	No. of Test Img.	Avg. Labels/Img.	Avg. Images/Label
Corel-5k	260	4,500	499	3.4	58.6
ESP Game	268	18,689	2,081	4.7	362.7
IAPRTC-12	291	17,665	1,962	5.7	347.7

Table 1: Statistics of the three image annotation datasets used in our experiments.

compared to others. This way, each negative becomes a negative in its own way as in SSVM. However, the time-complexity and/or memory requirements during training of SSVM-based models such as [1, 9] increase significantly as we move to very large datasets with large vocabularies. This usually makes it difficult to scale such models for the practical scenarios of large-scale learning (an interesting exception being the WSABIE model proposed in [21] that was shown to outperform SVM in terms of performance, time-complexity as well as memory requirements). SVM-VT provides the flexibility of both introducing semantics in classifier-training as well as efficient optimization comparable to binary SVM. Our work also relates with [13] that uses correlation between labels to improve annotation.

3 Experiments

3.1 Datasets and Features

We use three datasets popular in the image annotation task [5, 8, 11, 19]. These are Corel-5k [4], ESP Game [20] and IAPRTC-12 [7]. While Corel-5k has become the de-facto dataset in this domain, the other two datasets are very challenging with significant diversity among their samples. Table 1 shows general statistics of these datasets.

In our experiments, we use the same features as those in [8]. These include global RGB, HSV, LAB and GIST features; and local SIFT and Hue descriptors extracted densely from multi-scale grid as well as from Harris-Laplacian interest points. All features other than GIST are also computed over three equal horizontal partitions. This gives a set of 15 features per image. In our experiments, we also report results using chi-squared kernel. For SVM, SVM-VT and their kernelized versions, we calibrate the scores using [15].

3.2 Evaluation

We use the same evaluation criteria as being used by previous methods [6, 8, 11, 19, 23]. Given a new sample, first we compute the score for each label using the corresponding classifier, and then assign it the five top-scoring labels. To evaluate annotation performance, we use three measures. These are (a) average precision per label P, (b) average recall per label R, and (c) number of labels that are correctly recalled for at least one sample N+. Given a label l_i and its positive sample-set S_i^+, let Q_i be the set of images for which it is predicted. Then the precision for label l_i will be $\frac{|S_i^+ \cap Q_i|}{|Q_i|}$, and recall will be $\frac{|S_i^+ \cap Q_i|}{|S_i^+|}$. These values are computed for each label and averaged to obtain P and R scores. We also compute average F1-score per label (F1 = 2PR/(P+R)) to analyze the trade-off between P and R.

Table 2 shows the annotation performance of different methods. It can be seen that our method consistently improves performance over the conventional one-vs-rest SVM. Also, it performs comparable or better than even the recently proposed annotation methods such as [6, 8, 23] (except for IAPRTC-12 dataset where its performance is inferior only to the best results of [8]). We also compare with two SVM-based models [1, 9]. In [9], we use

Dataset →	Corel-5k				ESP Game				IAPRTC-12			
Method ↓	P	R	F1	N+	P	R	F1	N+	P	R	F1	N+
MBRM[6]	0.24	0.25	0.245	122	0.18	0.19	0.185	209	0.24	0.23	0.235	233
SML[2]	0.23	0.29	0.257	137	-	-	-	-	-	-	-	-
JEC[11]	0.27	0.32	0.293	139	0.22	0.25	0.234	224	0.28	0.29	0.285	250
TagProp-ML[3]	0.31	0.37	0.337	146	**0.49**	0.20	0.284	213	**0.48**	0.25	0.329	227
TagProp-σML[3]	**0.33**	**0.42**	**0.370**	**160**	0.39	**0.27**	**0.319**	**239**	0.46	**0.35**	**0.398**	**266**
GS[23]	0.30	0.33	0.314	146	-	-	-	-	0.32	0.29	0.304	252
RF[5]	0.29	0.40	0.336	157	0.41	0.26	0.318	235	0.44	0.31	0.364	253
M3L[9]	0.27	0.34	0.301	138	0.31	0.25	0.277	234	0.35	0.25	0.291	233
KM3L[9]	0.33	0.37	0.349	146	0.40	0.26	0.315	239	0.44	0.28	0.342	242
MLR-GL[1]	0.15	0.13	0.139	74	0.19	0.15	0.168	181	0.19	0.13	0.154	169
KMLR-GL[1]	0.18	0.16	0.169	85	0.22	0.17	0.192	190	0.23	0.16	0.189	174
SVM	0.24	0.37	0.291	164	0.25	0.26	0.255	254	0.29	0.28	0.285	262
KSVM	0.29	0.43	0.346	174	0.30	0.28	0.290	256	0.43	0.27	0.332	266
SVM-VT (Ours)	0.27	0.39	0.319	171	0.29	0.30	0.295	257	0.33	0.31	0.320	265
KSVM-VT (Ours)	**0.32**	**0.42**	**0.363**	**179**	**0.33**	**0.32**	**0.325**	**259**	**0.47**	**0.29**	**0.359**	**268**

Table 2: Performance comparison among different methods on the three image annotation datasets. The prefix 'K' corresponds to kernelization. Previous and our best results are highlighted in bold.

Dataset →	Corel-5k				ESP Game				IAPRTC-12			
Method ↓	P	R	F1	N+	P	R	F1	N+	P	R	F1	N+
2PKNN+ML[19]	0.44	0.46	0.450	191	0.53	0.27	0.357	252	0.54	0.37	0.439	278
KSVM-VT	0.32	0.42	0.363	179	0.33	0.32	0.325	259	0.47	0.29	0.359	268

Table 3: Performance comparison between the current state-of-the-art [19] and the best results of this work.

label co-occurrence scores to form the prior matrix. Here also, SVM-VT provides superior performance than both of these. This demonstrates its efficiency in capturing semantic correlations even in an independent manner. Table 3 compares our results with [19] that tries to benefit from label-specific local neighbourhoods and achieves current state-of-the-art performance on standard datasets. Though our results are somewhat inferior to that of [19], our method offers reduced run-time and better scalability. Figure 4 shows some qualitative results obtained using our method. These results reflect semantic connectedness among the labels predicted by our method; *e.g.* {*railroad, train, locomotive*}, {*boat, ship, ocean*}, etc.

4 Discussion

The SVM-VT model, despite its simplicity, offers several advantages over the existing discriminative and NN-based methods for image annotation which we discuss below:

(a) **Scalability**: Most of the discriminative methods for multi-label problems such as [1, 9] learn model(s) for all the labels in a vocabulary jointly in a single optimization problem. Though this provides the advantage of incorporating inter-label relationships, it is sometimes difficult to scale such methods to very large vocabularies. In contrast, SVM-VT provides

Figure 4: Example images from the three datasets and corresponding top 5 labels predicted using KSVM-VT method.

a framework for learning a model for each label in an independent manner, and at the same time takes care of inter-label relationships as well. Given an efficient way of computing the tolerance parameter, this can be scaled to very large vocabularies similar to SVM.

(b) **Time-complexity**: Once we have computed the tolerance parameter, time-complexity of SVM-VT is almost comparable to that of SVM. Since each classifier can be learnt independent of others, practically it is possible to learn all them simultaneously. Once we have learned all the classifiers, predicting labels for a new image becomes several times faster than the NN-based models [5, 8, 11, 19].

(c) **Performance**: As discussed before, the performance of SVM has remained (almost) unexplored in the task of image annotation on standard datasets. We demonstrated that simple SVM itself achieves superior performance than several existing methods. Moreover, SVM-VT demonstrates that it is possible to achieve significant improvement in performance by relaxing the strict discriminative behaviour of the SVM classifier. To the best of our knowledge, this is the first study where a discriminative one-vs-rest type of model has been shown to give promising results on image annotation task with large vocabularies.

Along with the SVM-VT model, we have also proposed a method for determining semantic relationships of negative samples with a given label based on visual similarity and dataset statistics. As vocabulary size grows, such relationships start getting prominent. However, due to limitations of human annotations, these give rise to the three issues discussed before. We show that using our method, we are able to find such relationships efficiently.

5 Conclusion and Future Work

We propose SVM-VT model for handling the issues of incomplete-labeling, label-ambiguity and structural-overlap that are frequently encountered in large vocabulary image annotation datasets. Our model is generic and can find applications in a wide variety of classification as well as multi-label tasks. We experimentally demonstrate that despite its simplicity, it

performs superior than several existing methods. In future, we would like to extend our work to the scenario where even some of the positive samples act as confusing samples. This, in turn, would help in learning models that are robust against incorrect ground-truth.

References

[1] S. S. Bucak, R. Jin, and A. K. Jain. Multi-label learning with incomplete class assignments. In *CVPR*, 2011.

[2] G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. In *PAMI*, 2007.

[3] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-78019-5.

[4] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[5] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[6] H. Fu, Q. Zhang, and G. Qiu. Random forest for image annotation. In *ECCV*, 2012.

[7] M. Grubinger. *Analysis and Evaluation of Visual Information Systems Performance.* PhD thesis, Victoria University, Melbourne, Australia, 2007.

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

[9] B. Hariharan, L. Zelnik-Manor, S. V. N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *ICML*, 2010.

[10] N. Krause and Y. Singer. Leveraging the margin more carefully. In *ICML*, 2004.

[11] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. In *IJCV*, 2010.

[12] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*, 2004.

[13] T. Mensink, J. Verbeek, and G. Csurka. Tree-structured crf models for interactive image labeling. In *PAMI*, 2012.

[14] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *ICML*, 1999.

[15] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

[16] Q. Song, W. Hu, and W. Xie. Robust support vector machine with bullet hole image classification. In *IEEE Transactions on Systems, Man and Cybernetics*, 2002.

[17] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[18] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[19] Y. Verma and C. V. Jawahar. Image annotation using metric learning in semantic neighbourhoods. In *ECCV*, 2012.

[20] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM SIGCHI*, 2004.

[21] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. In *ECML*, 2010.

[22] L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In *AAAI*, 2006.

[23] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D.N. Metaxas. Automatic image annotation using group sparsity. In *CVPR*, 2010.