Abstract

This paper presents a renewed image annotation baseline method under the nearest neighbor tag transfer framework. Two key problems are considered in this paper: (1) which images are determined as the neighbors; (2) how their keywords are transferred. Firstly, a soft neighbor selection scheme is designed by image embedding technique, with which we can provide more power to the crucial neighbors in decision making. Next, diffused tag propagation is introduced to allow one tag be propagated to other relevant tags. Besides this, the above two measures are formulated into an optimization framework to further improve the prediction performance. Experimental results on standard database show that the proposed approaches outperform the current state-of-the-art methods.

1. Introduction

Image retrieval is an active research topic in the last two decades, and three research stages have been experienced since then, i.e., traditional text based manual annotation, content based image retrieval (CBIR), and automatic image annotation[1]. Image annotation, whose goal is to automatically assign relevant text keywords to any given image reflecting its content, is significant to the management of large scale image data. Once images are annotated with semantic labels, they can be managed by keywords, where many mature tools from text processing area can be directly applied. The key characteristic of automatic image annotation is that it offers keyword searching based on image content and it employs the advantages of both the text based annotation and CBIR.

Image annotation is a difficult task for two main reasons[2]: First is the well known semantic gap problem, i.e., it is difficult to extract semantically meaningful entities using low level image features. The second difficult arises due to the lack of correspondence between keywords and image regions. Although difficult, much research progress has been made in the research community in order to bridge the gap between image features and high level semantic concepts, such as generative models[3][4][5], discriminative models[6], and nearest neighbor type models[2][7][8]. Compared to a potentially unlimited vocabulary in daily life, these methods can only effectively model a very limited number of concepts. Moreover, the images are usually very diverse in their appearance given the same concept, which makes the learned models unreliable and hardly generalizable[7].

Non-parametric nearest neighbor keyword transfer methods have been found to be quite successful for tag prediction[2][8]. These methods treat the problem of annotation as that of image retrieval: Nearest neighbors are determined from the training set, and their labels are then transferred to the target image. Although simple and effective, the keyword transfer schemes are dominated by two key factors: (1) which images are selected as the ‘neighbors’; (2) how their keywords are transferred. Following the two directions, many machine learning algorithms have been developed to maximize the predictive performance. In [9] and [10], metric learning methods are used in order to select semantically more coherent neighbors. In other research, weighting schemes are designed to make the neighbors have different contributions[2][7][8].

Motivated by the work in [2], we present a renewed baseline under the nearest neighbor transfer framework. Our target is not to develop complex annotation models, but use some simple and existing techniques to alleviate the problems related to the above two key factors. Our contributions are the following. First, the weights of the neighbors are determined based on their relation to the target image through linear embedding technique. With this scheme, soft neighbor selection is realized, which provides more right to the crucial neighbors in decision making. Second, diffused propagation is allowed by considering the tag correlation. As we know, tags are often conceptually relevant, but not isolated, so it is reasonable to extend the neighbor’s tag to other relevant ones during transferring. For example, a neighbor image with tag ‘sea’ may vote to ‘sky’ due to the fre-
quent co-occurrence of ’sea’ and ’sky’, even ’sky’ is not
the tag of this image. At last, we integrate the above two
measures into an optimization framework to further im-
prove the prediction performance.

1.1 Related Work

The existing image annotation methods can be
roughly divided into three groups: generative mod-
els, discriminative models, and nearest neighbor based
methods[2][10].

The generative models treat annotation as trans-
lation from images to tags by modeling the proba-
ibilistic distribution over image features and annotation
tags. CRM[4] and MBRM[3] are two representative ap-
proaches in this domain, and they are among the ear-
liest researches of image annotation. They build non-
parametric density estimations over the co-occurrence
of images and tags, and annotate a new image accord-
ing to the conditional probability over tags given the image
features. Topic models can be seen as an indirect ex-
tension to the co-occurrence models, where images and
tags are interconnected by topics, images are annotated
as samples from a specific mixture of topics[5].

Discriminative models realize annotation by classi-
fication solution. These methods learn a classifier for
each tag, and use them to predict whether the test im-
age belongs to the class defined by this tag. The rep-
resentative classification methods include SVM, multiple
instance learning, decision tree, and so on[1].

Nearest neighbor based methods have been com-
monly recognized as a powerful annotation tool since
Makadia’s work[2]. In this method, image annotation is
solved as a retrieval problem, where nearest neigh-
bors are selected by the average of several distances,
and the keywords are then transferred from neighbors to
the given image. Although the idea is simple, this method
is shown to outperform more complex state-of-the-art
annotation methods. In TagProp[8], neighbor weights
are determined based on neighbor rank or distance, and
metric learning method is integrated. In [10] and [9],
sparse coding method is designed toward effective fea-
ture selection with the guidance of keyword similarity.

2. Weighted nearest neighbor annotation

Our proposed method is in fact a weighted nearest
neighbor approach with two factors being integrated:
(1) Image weight. We use locally linear embedding to
assign weights to the neighbors to enable the neighbors
have different contributions. (2) Tag weight. By inves-
tigating the correlation of the conceptual tags, each tag
is allowed to be propagated to relevant tags.

2.1 Image weights by linear embedding

Let the set of \( N \) training images be denoted as \( \mathbf{X} = \{ x_1, x_2, \ldots, x_N \} \). Given the test image \( \hat{x} \), find its \( K \)
nearest neighbors in the training set, \( x_1, \ldots, x_K \). Sup-
pose the data points are sampled from some smooth un-
derlying manifold, the test image, \( \hat{x} \), can be linearly re-
constructed from its neighbors, and the reconstruction
error is measured by the squared distances between the
data point and its reconstruction[11]:

\[
\varepsilon(\mathbf{W}) = ||\hat{x} - \sum_{i=1}^{K} w_i x_i'||^2 \tag{1}
\]

\[
s.t. \sum_{i=1}^{K} w_i = 1
\]

\[
w_i \geq 0, i = 1, \ldots, K
\]

where, the weights \( w_i \) indicate the contribution of the
\( i \)-th neighbor to \( \hat{x} \)’s reconstruction.

By considering the constraint that all \( w_i \)’s sum to
one, we can rewrite the reconstruction error as:

\[
\varepsilon = ||\hat{x} - \sum_{i} w_i x_i'||^2 = \mathbf{W}^T \mathbf{C} \mathbf{W}
\]

where \( \mathbf{C} \) is a \( K \times K \) local covariance matrix asso-
ciated with \( \hat{x} \), and the element value is computed by:

\[
C_{mn} = (\hat{x} - x_m)' (\hat{x} - x_n'), m, n = 1 \ldots K.
\]

Thus, the minimization of the reconstruction error can be readily obtained by solving a standard quadratic
programming problem.

2.2 Tag correlation

The relationship of the tags is derived from two cues:
(1) Semantic correlation, which defines the tags rep-
resenting the similar object, e.g., ’rock’ and ’stone’. This
type of information can be obtained according to some
knowledge base such as WordNet. (2) Concurrence cor-
relation, which indicates the tags that appear together
frequently in the images, and the tags are associated by
the image content due to the positional relationship of
the objects, such as ’sea’ and ’beach’.

In this paper, we define the similarity of tags based on
their co-occurrence. For example, if two tags appear
together frequently, the distance between them should be
small. Assume there are \( M \) distinct tags in the train-
ing images, and denote them as \( T = \{ t_1, \ldots, t_M \} \). The
distance between two tags \( t_i \) and \( t_j \) can be estimated by
the Google distance measure[12], as follows:

\[
d(t_i, t_j) = \frac{\max(\log f(t_i), \log f(t_j)) - \log f(t_i, t_j)}{\log n - \min(\log f(t_i), \log f(t_j))} \tag{2}
\]

where \( f(t_i) \) and \( f(t_j) \) are the number of images con-
taining tag \( t_i \) and \( t_j \) respectively, \( f(t_i, t_j) \) is the num-err of images containing both \( t_i \) and \( t_j \), and \( n \) is the number of training images.
With the Google distance function, the concurrence similarity between tags \( t_i \) and \( t_j \) is then defined as:

\[
s(t_i, t_j) = \exp(-d(t_i, t_j)) \tag{3}
\]

2.3. Weighted annotation

Our weighted annotation scheme propagates the tags of the nearest neighbors to the test image by considering both image weights and tag correlations.

Given the test image \( \tilde{x} \), we first find its \( K \) nearest neighbors in the training set, and compute the corresponding weights \( W \) with equation (1). Let \( L_{M \times K} \) be the tag membership matrix for the \( K \) neighbor images, where \( L_{ij} = 1 \) if tag \( i \) belongs to image \( j \), and 0 otherwise. Denote \( S \) as the similarity matrix of the tags obtained by (3). Then, we compute the annotation of the test image by:

\[
T(\tilde{x}) = S \cdot L \cdot W \tag{4}
\]

3. Annotation by optimization framework

Based on the assumption of content consistency that visually similar images often reflect similar themes and thus are typically annotated with similar tags, this section presents an optimization based annotation method.

Let \( X = \{x_1, \ldots, x_n\} \) be the union of the training and test images, and \( F = \{f_1, \ldots, f_n\} \) be the corresponding tag matrix, whose element \( f_{ij} \) indicates the likelihood of tag \( t_i \) with respect to image \( x_j \).

Similar to the idea of LLE[11], the reconstruction weights, which recover the local neighborhood structures of the data points in the visual space, are expected to be equally valid in the tag space. The distortion function is defined as:

\[
\Omega(F) = \sum_{i=1}^{n} \text{dist}(f_i, \sum_{j=1}^{K} w_{ij} f'_j) \tag{5}
\]

where \( f'_j \) is the tag vector of the \( j \)-th neighbor of \( x_i \).

In consideration of tag correlation, the quadratic form distance is adopted. Besides this, a fitting constraint is integrated to enforce that the tags should not change too much from their initial values. Thus, the cost function is formulated as:

\[
\Omega(F) = (1 - \alpha) \sum_{i=1}^{n} |f_i - y_i|^2 + \alpha \sum_{i=1}^{n} (f_i - \sum_{j=1}^{K} w_{ij} f'_j)^T \cdot S \cdot (f_i - \sum_{j=1}^{K} w_{ij} f'_j) \tag{6}
\]

where \( y_i \) are the initial image tags. The initial tags are directly available for the training images. For test images, however, they can be obtained by weighted nearest neighbor annotation process.

The above equation can be written in matrix form as

\[
\Omega(F) = (1 - \alpha) tr((F - Y)^T (F - Y)) + \alpha \cdot tr((I - W)^T F^T S F (I - W)) \tag{7}
\]

Differentiating \( \Omega(F) \) with respect to \( F \), we have

\[
(1 - \alpha)(F - Y) + \alpha \cdot (S + S^T)FDD^T = 0 \tag{8}
\]

where \( D = I - W \).

The above matrix equation is called Sylvester Equation which often occurs in the control domain. The final annotation result, \( F \), is thus obtained by solving this equation through some existing methods.

In weighted nearest neighbor annotation, the tags are transferred unidirectionally from training images to the test one. In contrary to this, the training and test images are equally managed in the optimization framework, and tags are allowed to be propagated in the whole data set.

4. Experiments

This section presents a quantitative evaluation of the proposed methods on the Corel5K data set, and the comparison with some previous methods is also given.

4.1 Data set and features

Corel5K is a basic evaluation benchmark in the image annotation community. It contains 5,000 images collected from the larger Corel CD set. Out of the 5,000 images, 4,500 images are used for training and the other 500 images are used for testing. In this DB, each image is annotated with 1 to 5 keywords (3.5 keywords on average), and the dictionary is formed by 260 words that appear in both the training and the test set.

The 15 features of TagProp are adopted here for a fair evaluation[8]: one Gist descriptor, 6 color histograms and 8 bag-of-words local features. The color histograms are generated with 16 bins quantization in each color channel for RGB, LAB, and HSV space. The local features include SIFT as well as a hue descriptor, both extracted densely on a multi-scale grid or on the interest points. To compute the distances from the descriptors, L2 are used as the base metric for Gist, L1 for global color histograms, and \( \chi^2 \) for the others.

4.2 Results

As in previous works, three standard measures are used for performance evaluation: mean precision(\( \mathbf{P} \)), mean recall(\( \mathbf{R} \)), and the number of words with nonzero recalls(\( \mathbf{N}^+ \)). The precision of a keyword is defined as the number of correctly annotated images divided by the total number of images annotated with this keyword,
and the recall is similarly defined. In the annotation, each image is assigned with the 5 most relevant keywords, and the mean values of precision and recall are computed over all 260 keywords.

The effectiveness of the two factors, image weights and tag weights, is analyzed under the weighted nearest neighbor annotation scheme, as shown in Table 1. ‘Basic kNN’ is the standard kNN voting method with no image and tag weights, which is same to JEC-15 in [8]. Linear embedding is used in ‘Image weight’ to get suitable weights for the neighbor images, but tag correlation is not considered. On the contrary, the tag similarity is adopted in ‘Tag weight’ while the neighbor images are equally treated. At last, both the two weights are integrated to realize the proposed weighted nearest neighbor annotation.

It can be seen from Table 1 that image weights are helpful toward performance improvement, i.e., the neighbors should be assigned reasonable powers in decision making. This is consistent with other researches, where more complex optimization methods are often used to compute the image weights to get better annotation. No benefits are obtained from tag weights alone. What is more, the performance was degraded. This is mainly due to the unreasonable usage of equal image weights. By combining the tag and image weights, the mean precision is further improved to 0.33 from 0.31.

We compare the proposed methods with some other state-of-the-art methods, as shown in Table 2. In this table, the proposed weighted nearest neighbor method and optimization based annotation method are denoted as Wnn and Wopt respectively. By allowing the tags to be propagated in the whole image set, the precision can be further improved from 0.33(Wnn) to 0.35(Wopt). Both the two methods are superior to other methods in annotation precision, which confirms the effectiveness of our methods. The recall value is better than all methods except TagProp. This is because an additional logistic model is designed in TagProp to boost the probability of rare tags in order to get high recall scores.

5 Summary

A renewed image annotation baseline method is proposed in this paper. To solve the two key problems related to tag propagations, image embedding and tag correlation are used respectively to generate image and tag weights. Certainly, there are some other choices that can be used to further improve the performance, such as feature selection or metric learning to get better consistency between image and semantic space, complex sparse coding methods to get better image weights, etc. However, our target is not to develop complex annotation models, but present a baseline with some simple and existing measures. Although the idea of the proposed methods is simple, these methods still outperform the state-of-the-art methods in the standard data set.

Table 1: Comparison of different weight factors

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>N+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic kNN</td>
<td>0.28</td>
<td>0.33</td>
<td>140</td>
</tr>
<tr>
<td>Image weight</td>
<td>0.31</td>
<td>0.35</td>
<td>145</td>
</tr>
<tr>
<td>Tag weight</td>
<td>0.26</td>
<td>0.34</td>
<td>142</td>
</tr>
<tr>
<td>Both weights</td>
<td>0.33</td>
<td>0.38</td>
<td>152</td>
</tr>
</tbody>
</table>

Table 2: Results of different annotation methods

<table>
<thead>
<tr>
<th>CRM</th>
<th>MMRM</th>
<th>JEC</th>
<th>TagProp</th>
<th>MSC</th>
<th>GF</th>
<th>Wnn</th>
<th>Wopt</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.16</td>
<td>0.24</td>
<td>0.27</td>
<td>0.33</td>
<td>0.25</td>
<td>0.30</td>
<td>0.33</td>
</tr>
<tr>
<td>R</td>
<td>0.19</td>
<td>0.25</td>
<td>0.32</td>
<td>0.42</td>
<td>0.32</td>
<td>0.33</td>
<td>0.38</td>
</tr>
<tr>
<td>N+</td>
<td>107</td>
<td>122</td>
<td>139</td>
<td>160</td>
<td>136</td>
<td>146</td>
<td>152</td>
</tr>
</tbody>
</table>

References