Support vector description of clusters for content-based image annotation

Liang Sun, Hongwei Ge, Shinichi Yoshida, Yanchun Liang, Guozhen Tan

Abstract
Continual progress in the fields of computer vision and machine learning has provided opportunities to develop automatic tools for tagging images; this facilitates searching and retrieving. However, due to the complexity of real-world image systems, effective and efficient image annotation is still a challenging problem. In this paper, we present an annotation technique based on the use of image content and word correlations. Clusters of images with manually tagged words are used as training instances. Images within each cluster are modeled using a kernel method, in which the image vectors are mapped to a higher-dimensional space and the vectors identified as support vectors are used to describe the cluster. To measure the extent of the association between an image and a model described by support vectors, the distance from the image to the model is computed. A closer distance indicates a stronger association. Moreover, word-to-word correlations are also considered in the annotation framework. To tag an image, the system predicts the annotation words by using the distances from the image to the models and the word-to-word correlations in a unified probabilistic framework. Simulated experiments were conducted on three benchmark image data sets. The results demonstrate the performance of the proposed technique, and compare it to the performance of other recently reported techniques.

1. Introduction
The number of image archives on the Internet is growing rapidly with the proliferation of user-contributed images. Thus, searching for images that match a user query presents a significant challenge. Popular search engines, such as Google and Yahoo!, rely mainly on textual descriptions contained in the filenames or keywords. Many search engines do not consider the content of images and are able to search only the annotations. On the other hand, in real-world image systems, many images are created without direct annotations of semantic content, which limits the ability to search by text-based engines. This creates the need for content-based image retrieval (CBIR) [1–3]. Research on CBIR has attracted the attention of researchers in various fields, including computer vision and machine learning. In a CBIR system, instead of using text descriptions, searches are based on low-level features such as color, texture, and shape. Some aspects of CBIR have been shown to be successful, and some progress has been made [4–9]. However, it still suffers from the "semantic gap" problem, which arises because the low-level features of an image are not sufficient to encapsulate the high-level semantics [1,10].

One natural way to mitigate the semantic gap problem is to assign tags to images. Appropriate tagging can help to increase the efficiency of retrieval. However, manual tagging is tedious and labor intensive [11–13], so there has been a surge of interest in developing automatic or semi-automatic tagging based on the low-level visual contents of an image [11–27]. These methods are referred to as content-based image annotation (CBA). In greater detail, the methods presented in [13–17] are based on multiple classifiers. They partition the images into different classes, and then they assign to each class a distinct topic of interest and a set of descriptive words. The system treats the annotation of an untagged image as a classification problem, and it selects the relevant annotation based on the classification results. The methods presented in [18–26] are probabilistic modeling methods, which are also referred to as generative modeling methods. They use statistical tools to determine the correlations between images and annotations so that they can compute the joint probability that an untagged image is labeled with a particular word. In the annotating process, the relevant words are selected by a data fusion and aggregation technique [18–22,25,26]. More recently, the development of image platforms on the Internet, e.g., Flickr [28], Alipr [19], and PhotoStuff [29], has enabled users to
annotate images and give feedback on annotations. This provides opportunities to develop automatic annotating methods that use the information provided by users and the existing search results. The methods presented in [11,13], and [27] fall into this category.

Many of the above methods require substantial machine learning techniques to fill the gap between the low-level visual contents of an image and the high-level semantics. Among machine learning techniques, support vector clustering (SVC) [30–32], is a recently developed algorithm that was inspired by the support vector machine (SVM) [33]. The SVC maps data points from the original space to a higher-dimensional space by means of a Gaussian kernel and then seeks the smallest sphere that encloses most of the data points. The sphere is mapped back to the original space as several arbitrarily shaped contours, each of which encloses a subset of the data points. SVC has many advantages over other algorithms, including its ability to determine, without prior knowledge, the topological structure of a system, to delineate the boundaries of irregularly shaped clusters, and to deal with outliers by employing a soft-margin constant [34,35]. In real-world systems, the same descriptions may apply to a wide variety of images. For example, if an image is tagged with “historical building”, then it might be a picture taken in the desert, near the beach, or in the city. There is not a clear way in which these images should be organized. Since SVC is able to delineate irregular cluster boundaries, it can be used to develop unified models to describe unorganized images.

In view of the above, in this paper, we present a novel algorithm for CBIA. In this work, images are represented by colored pattern appearance models (CPAM) [36]. The system has two major components, the training process and the tagging process. In the training process, clusters of images with manually tagged words are used as training instances. For each cluster, the feature vectors of the image are mapped to a higher-dimensional space, and the support vectors are used to describe the cluster. Since the mapping process is the same as that in SVC and its objective is to build models that are described by support vectors for image clusters, we call it the support-vector-based method for image annotation (SVIA). In order to tag an image, the method computes the distances between the image and the models that are described by support vectors. A shorter distance indicates a stronger association between the image and the model, and thus indicates a stronger association between the image and the corresponding words of the model. Moreover, the word-to-word correlations contain rich information about the meanings of the images. For example, if an image is tagged with “France”, then it will have a higher probability of being tagged with “Europe”, and if an image is tagged with “indoor”, then it will have a lower probability of being tagged with “grass”. Therefore, the word-to-word correlations are also considered in the annotation framework. When tagging an image, the system predicts the appropriate words by using the distances from the image to the models and the word-to-word correlations in a probabilistic framework.

The remainder of this paper is organized as follows. Section 2 reviews some of the previous work on CBIA; in particular, the probabilistic modeling methods and the SVM algorithm. Section 3 presents the support-vector-based modeling method. Section 4 presents the proposed probabilistic modeling method for the assignment of annotations. Section 5 presents and compares simulated result, followed by concluding remarks in Section 6.

2. Related works

This work is related to probabilistic modeling methods, and the support-vector-based model is related to SVC. In this section, we begin by reviewing the basic concepts and some prevailing methods of the probabilistic modeling approaches to CBIA. Next, we review the basic concepts of SVC and some of its variants.

2.1. Probabilistic modeling approaches to content-based image annotation

2.1.1. Probabilistic models

The probabilistic modeling method computes, for each work, the joint probability that an untagged image is tagged with that word. Given an untagged image \( I_q \), the main objective is to find a group of words \( w^* \) in a given vocabulary \( \mathcal{W} \) so that the conditional distributions \( p(w|I_q) \) are maximized as follows:

\[
\arg \max_{w \in \mathcal{W}} p(w|I_q).
\]

(1)

Generally, there are two types of probabilistic models for image annotation, i.e., the two-layer model and the three-layer model. The two-layer model generates words directly from the given image, as shown in Fig. 1(a). By applying Bayes’ rule to Eq. (1), we obtain the following formulation:

\[
\arg \max_{w \in \mathcal{W}} p(w|I_q) = \arg \max_{w \in \mathcal{W}} \sum_{I_i \in \mathcal{T}} p(w|I_i)p(I_i|I_q)p(I_q).
\]

(2)

where \( \mathcal{T} \) is the training image set, \( I_i \) is the \( i \)th training image in \( \mathcal{T} \), \( p(w|I_i) \) is the probability that image \( I_i \) is correlated with word \( w \), \( p(I_i|I_q) \) is the probability that image \( I_q \) is relevant (or similar) to image \( I_i \), and \( p(I_q) \) is the prior probability of the training image \( I_q \). The three-layer model, however, introduces a hidden layer of “topics”, and an image is then represented by a mixture of topics. Words are then generated from these topics, as shown in Fig. 1(b).

By applying Bayes’ rule to Eq. (1), we obtain the following formulation:

\[
\arg \max_{w \in \mathcal{W}} p(w|I_q) = \arg \max_{w \in \mathcal{W}} \sum_{S \in \mathcal{T}} \left\{ \left( \sum_{t_j \in S} p(w|t_j)p(t_j|I_i) \right) \times p(I_i|I_q)p(I_q) \right\},
\]

(3)

where \( S \) is the set of topics, \( t_j \) is the \( j \)th topic in \( S \), \( p(w|t_j) \) is the probability that topic \( t_j \) is correlated with word \( w \), \( p(t_j|I_i) \) is the probability that image \( I_i \) is correlated with \( t_j \), and \( p(I_i|I_q) \) and \( p(I_q) \) have the same meanings as in Eq. (2). In addition, since the word-to-word correlations contain rich information about the meanings of the images, some of the methods integrate the word-to-word correlation \( p(w|I_q) \) into the two-layer model or the three-layer model to maintain semantic consistence.

![Fig. 1. Schematic diagrams of probabilistic models for image annotation. (a) Two-layer model and (b) three-layer model.](image-url)
2.1.2. Prevailing methods

Some of the prevailing methods are based on the formulation described in Eq. (2) or its variants. For instance, Duygulu et al. [18] proposed an object recognition model to translate regions of the image into words. In this method, images are first segmented into regions, which are then classified into region types based on the visual contents. A mapping between the region types and words is then learned by using an algorithm that maximizes expectation. Tang et al. [24] explored an integrated graph-based semi-supervised learning framework that combines multiple and single instances of image features to determine the annotation. This study also explored three strategies for converting from multiple-instance to single-instance representations. Lu and Ip [22] proposed a discriminative stochastic method for image categorization and annotation. In this method, images are divided into blocks, and visual keywords are generated by quantizing the features of the image blocks. The categorization and annotation are then performed by a spatial Markov chain model that uses the visual keywords as input.

Other methods are based on the formulation described in Eq. (3) or its variants. For instance, Li and Wang [19] proposed a statistical modeling approach to CBIA. In this method, categorized images are used to train a set of statistical models, each of which represents a concept visual similarity. Compared with the previous methods, the proposed algorithm helps to mitigate the problem of intraconcept visual diversity.

We propose the use of a support-vector-based algorithm, as used in SVC, to model images that are described by the same words or topics. Unlike Eq. (2), in the proposed algorithm, the probabilities $p(w_i|l)$ and $p(l_i)$ are hidden by the support-vector-based models. Due to the principle of the minimal enclosing sphere, the algorithm exhibits a promising ability to determine the topological structure of the system, without prior knowledge, to delineate the cluster boundaries of irregular shapes, and to deal with outliers by employing a soft-margin constant. With these properties, the proposed algorithm helps to mitigate the problem of intraconcept visual diversity.

2.2. Support vector clustering algorithms

Following the derivations of Ben-Hur et al. [32], the mathematical formulation of the SVC algorithm can be summarized as follows. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a $d$-dimensional data space, and let $\{x_1, x_2, \ldots, x_N\} \subseteq \mathcal{X}$ be a data set. The algorithm uses a nonlinear transformation $\Phi$ to map the data points from $\mathcal{X}$ to a higher-dimensional space and seeks the smallest sphere that encloses most of the data points in the higher-dimensional space; i.e., it uses the minimal enclosing sphere to describe the given data set. The optimization problem can be formulated as follows:

\[
\text{minimize} \quad R^2 + C \sum_{j=1}^{N} \xi_j^2 \\
\text{subject to} \quad \|\Phi(x_i) - a\|^2 \leq R^2 + \xi_j, \quad j = 1, 2, \ldots, N,
\]

where $R$ is the radius of the enclosing sphere, $\xi_1, \xi_2, \ldots, \xi_N$ are slack variables, $C$ is the soft-margin constant, $\|\cdot\|$ is the Euclidean norm, and $a$ is the center of the sphere. The problem can be solved by introducing the Lagrangian function:

\[
L = R^2 - \sum_{j=1}^{N} (R^2 + \xi_j - \langle \Phi(x_j), a \rangle)^2/\beta_j + C \sum_{j=1}^{N} \xi_j - \sum_{j=1}^{N} \xi_j \mu_j,
\]

where $\beta_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers. The solution to the primal problem described in Eq. (5) can be obtained by solving the dual problem [37]:

\[
\text{maximize} \quad W = \sum_j \Phi(x_j)^T \beta_j - \sum_{ij} \beta_j \beta_i \Phi(x_i) \Phi(x_j), \\
\text{subject to} \quad 0 \leq \beta_j \leq C, \sum_j \beta_j = 1.
\]

The products $\Phi(x_i) \Phi(x_j)$ can be replaced by the Gaussian kernel, which is described as

\[
K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2),
\]

annotation, i.e., it computes the probability that an untagged image is tagged with each word, and then finds the group of words for which the conditional distributions are maximized, as in Eq. (1). Since the model of Eq. (2) generates words directly from a given image, it is generally simple yet effective. Thus the algorithm proposed in this paper follows a variant of the formulation of Eq. (2). However, rather than simply following the formulation of Eq. (2), we developed a support-vector-based algorithm to mitigate the problems of intraconcept visual diversity and interconcept visual similarity. Compared with the previous methods, the main differences of this method can be summarized as follows.

- We propose the use of a support-vector-based algorithm, as used in SVC, to model images that are described by the same words or topics. Unlike Eq. (2), in the proposed algorithm, the probabilities $p(w_i|l)$ and $p(l_i)$ are hidden by the support-vector-based models. Due to the principle of the minimal enclosing sphere, the algorithm exhibits a promising ability to determine the topological structure of the system, without prior knowledge, to delineate the cluster boundaries of irregular shapes, and to deal with outliers by employing a soft-margin constant. With these properties, the proposed algorithm helps to mitigate the problem of intraconcept visual diversity.

- We propose expanding the strategy that uses a single enclosing sphere to one that uses multiple enclosing spheres; i.e., we adopt a strategy of one concept to one sphere. Images described by different words or topics are modeled using different spheres. Since the spheres possess different properties, such as center and radius, they can characterize these images separately. Thus they separate the interrelated images more effectively.

...
where $h$ is the bandwidth of the Gaussian kernel. The trained kernel radius function, which defines the squared radial distance of a given data point $x$ from the sphere center $a$, can be formulated as

$$f(x) = R^2(x) = \|\Phi(x) - a\|^2 = K(x, x) - 2\sum_i \beta_i K(x, x_i) + \sum_{ij} \beta_i \beta_j K(x_i, x_j).$$

(8)

If $\beta = C$, then $x_i$ is identified as a bounded support vector (BSV). If $0 < \beta < C$, then $x_i$ is identified as a support vector (SV). Otherwise, $x_i$ is identified as an inner point.

SVC algorithms have attracted the attention of researchers, and several variants have been proposed. For instance, Chiang and Hao [38] extended the SVC to an adaptive cell-growing model that uses a kernel function to map data points to a higher-dimensional space. Camasta and Verri [39] proposed a kernel method in which each cluster is iteratively refined using a one-class support vector machine. J. Lee and D. Lee [35] used support vector clustering to develop a topological and dynamical characterization of the cluster structures.

3. Support vector description of clusters

In this section, we describe the training process of the proposed SVIA system. First, we review the colored pattern appearance model (CPAM) [36], which represents the color and texture of the images. Following this, we present the details of the support vector modeling process. Finally, we describe the way in which the proposed model estimates the probability of a given image.

3.1. CPAM representation of images

In this work, the visual contents of images are represented by the CPAM. The CPAM shares some similarities with the “bag of visual words” idea that is popular in the computer-vision literature. It captures statistically representative chromatic and achromatic spatial image patterns, and the distributions of these patterns are used to characterize the color and texture information. We used our system to tag images taken from daily life, rather than from special fields such as medicine or geography. Since the CPAM can capture both the features around salient points and those included in the entire image [13], it is suitable for this application.

The CPAM comes with a codebook of prototypes, which has been built and trained based on tens of thousands of image patches and uses vector quantization (VQ). To represent an image using the CPAM, a sliding window is used to partition the image into a set of nonoverlapping $4 \times 4$ blocks. The visual appearance of each block is modeled by an achromatic spatial pattern (ASP) and a chromatic spatial pattern (CSP). The block is then approximated by the ASP and CSP prototypes that are most similar to its corresponding ASP and CSP, respectively. The feature vector $x$ is built as a histogram for the entire image, and it tabulates the frequencies of the appearance prototypes being used to approximate the $4 \times 4$ blocks. Because it partitions the entire image into a set of nonoverlapping $4 \times 4$ blocks and builds the feature vector as a histogram of the ASP and CSP prototypes, a CPAM-based method can deal with images of different sizes. In the experiment that we conducted in this study, we selected a codebook with 64 ASP and 64 CSP prototypes. Therefore, the entire image can be represented by a 128-dimensional vector, i.e., 64 achromatic spatial-pattern histograms (ASPHs) and 64 chromatic spatial-pattern histograms (CSPHs). To determine the similarities of two images, $I_1$ and $I_2$, we can calculate the distance between their ASPHs and CSPHs. As reported in [13] and [36], the $L_1$ distance measure is more robust than the other distance metrics, and so we adopted the $L_1$ distance measure in our experiments. If we suppose that the CPAM-based feature vectors of $I_1$ and $I_2$ are represented by $x_1$ and $x_2$, respectively, then the distance between $x_1$ and $x_2$ can be defined as follows:

$$d(x_1, x_2) = \sum_{I} |\text{ASPH}_{I}(i) - \text{ASPH}_{j}(i)| / \sum_{I} |\text{ASPH}_{I}(i) + \text{ASPH}_{j}(i)| + \sum_{I} |\text{CSPH}_{I}(j) - \text{CSPH}_{j}(j)| / \sum_{I} |\text{CSPH}_{I}(j) + \text{CSPH}_{j}(j)|.$$

(9)

where $|\cdot|$ is the absolute value, ASPH($i$) is the histogram of the $i$th ASP prototype, and CSPH($j$) is the histogram of the $j$th CSP prototype. As can be seen from Eq. (9), for a pair of images, a smaller value of $d(x_1, x_2)$ indicates that $x_1$ and $x_2$ are more similar.

3.2. The model described by support vectors

Let the set of distinct annotation words be $W = \{w_1, w_2, ..., w_k\}$. Given a word $w_i$, let the set of images tagged with $w_i$ be $I(w_i)$. Fig. 2 illustrates the support vector modeling process. First, the CPAM-based feature vectors are extracted from the images. Following this, a model described by support vectors is trained for each image set $I(w_i)$. The training method uses a nonlinear transformation to map the CPAM-based feature vectors to a higher-dimensional space. It then seeks the smallest enclosing sphere in the mapped space. Each sphere represents a support vector described model. This process closely follows the derivations of Eqs. (4)–(6). In Fig. 2, for the convenience of illustration, we assume that the CPAM-based feature vectors are mapped to a three-dimensional space. The spheres are referred to as the support vector described models, which are stored in the form of the parameters of the trained kernel radius functions.

The Gaussian kernel described in Eq. (7) uses the Euclidean norm to calculate the distance between the data points, while in the CPAM-based image system, we use Eq. (9) to calculate the distance between the image features. Thus, the Gaussian kernel described in Eq. (7) cannot be directly applied to the CPAM-based image features. In this paper, we define the kernel function as follows:

$$K(x_i, x_j) = e^{-d(x_i, x_j)/h},$$

(10)

where $d(x_i, x_j)$ is the distance between the CPAM-based features $x_i$ and $x_j$. The bandwidth parameter $h$ plays a crucial role in the support vector described model. It controls the shape of the enclosing contour in the data space and affects the width of the kernel function. The problem of selecting the parameter $h$ can be referred to as balancing the empirical risk and the confidence risk [40]. In this paper, the parameter $h$ is fixed by trial and error, which iteratively adjusts the $h$ value and calculates the system score. The $h$ value that yields the highest system score is adopted. For a set of training images and a set of validating images, the system score can be calculated as follows. First, the system score is set to 1, and then a support vector described model is trained for each image set $I(w_i)$ using the current $h$ value. Following this, for each validating image, the distance from it to the support vector described models is calculated using the trained kernel radius function described in Eq. (8). Suppose that $w_i$ is the model that yields the shortest distance and that $M_i$ is the set of models that actually generate the validating image. If $m_i \in M_i$, then the system score is increased by 1; otherwise, it remains unchanged. The system score is obtained once all the images in the validating set have been evaluated. For the experiment, we randomly selected 10% of the images for training and 5% for validating.
The computational cost of this method is primarily due to solving the quadratic programming (QP) problem described in Eq. (6). The sequential minimal optimization (SMO) algorithm [41], which was proposed for the training of the support vector machine (SVM), is by far the most widely used method to solve this problem. It decomposes the QP problem into subproblems, and then analytically optimizes the subproblems. This process avoids using time-consuming numerical QP optimization. The amount of memory required for SMO increases linearly as the number of training samples increases, and can reduct by the proposed weighted kernel density estimator, the clustering accuracy is improved by the proposed weighted kernel density estimator, the clustering accuracy is improved.

Given test data \( x \), if \( x \) lies outside of the sphere, then we have

\[
\hat{p}(x) = \frac{1}{R} \sum_{j=1}^{n} \beta_j K(x, x_j).
\]

The above derivation indicates that the trained kernel radius function of the support vector described model delineates a weighted density estimator for the underlying distribution of the training data. Thus, in this paper, we use this density estimator to compute the probability that a given image is generated by the support vector described models. Given a CPAM-based feature vector \( x \) of image \( I \) and a trained kernel radius function of model \( m \), the probability that \( x \) is generated by \( m \) can be represented by \( p(x|m) \). Since \( x \) is the CPAM-based feature vector of image \( I \) and model \( m \) is associated with the set of training images tagged with word \( w \), the following formula can be obtained:

\[
p(I|w) = p(x|m) = \frac{1}{R} \sum_{j=1}^{n} \beta_j K(x, x_j).
\]
3.4. The training process

In this subsection, we describe the framework of the training process of SVIA. Let the set of distinct annotation words be $W = \{w_1, w_2, \ldots, w_N\}$. Let the set of training images tagged with $w_i$ be $I(w_i)$. Let the set of trained support vector described models be $M$. The flowchart of the training process of SVIA is shown in Fig. 3. Initially, $W$ is the set of all the annotation words and $M$ is empty. The system trains support vector described models for each of the annotation words until all the words in $W$ have been used. Finally, the system outputs the set of trained support vector described models as outcomes.

4. The annotation method

In this section, we describe the tagging process of the SVIA. We propose a unified probabilistic framework that generates words from the support vector described models and the word-to-word correlations.

4.1. The word-to-word correlations $p(w_i | w_j)$

The word-to-word correlations contain information about the semantic meanings of the images. For instance, if an image is tagged with “France”, then there is higher probability that it will be tagged with “Europe”. If an image is tagged with “indoor”, then there is a lower probability that it will be tagged with “grass”. If the system uses only the visual contents of images to generate tagging words, the information contained in the word-to-word correlations will be lost. Therefore, the word-to-word correlations are considered in the probabilistic framework. Let $T = \{I_1, I_2, \ldots, I_N\}$ be a set of images in the training set, and let $W = \{w_1, w_2, \ldots, w_M\}$ be a given vocabulary. Each image $I_i \in T$ is manually tagged with a set of $M_i$ words $\{w_{i1}, w_{i2}, \ldots, w_{iM_i}\}$. Denote the set of images that contain word $w_i$ in their annotations by $I(w_i)$. Given a pair of words $w_i$ and $w_j$, the following formulations are used to estimate the word-to word correlations:

$$p(w_i | w_j) = \frac{|I(w_i) \cap I(w_j)|}{|I(w_j)|},$$

$$p(w_j | w_i) = \frac{|I(w_i) \cap I(w_j)|}{|I(w_i)|},$$

where $|\cdot|$ is the number of images in the image set. Generally, the relationship between $I(w_i)$ and $I(w_j)$ falls into one of four categories, which correspond to the figures shown in Fig. 4 and can be summarized as follows:

1. If $I(w_i) \cap I(w_j) \neq \emptyset$, then $p(w_i|w_j) > 0$ and $p(w_j|w_i) > 0$;
2. If $I(w_i) \subseteq I(w_j)$, then $0 < p(w_i|w_j) \leq 1$, and $p(w_j|w_i) = 1$;
3. If $I(w_i) \supseteq I(w_j)$, then $p(w_i|w_j) = 1$, and $0 < p(w_j|w_i) < 1$;
4. If $I(w_i) \cap I(w_j) = \emptyset$, then $p(w_i|w_j) = 0$ and $p(w_j|w_i) = 0$.

4.2. The probabilistic framework

Given an untagged image $I_q$ when the system has not selected any word as annotation, the conditional probability that $I_q$ is tagged with $w_i$ can be represented by $p(w_i|I_q)$, which can be computed by applying Bayes’ rule:

$$p(w_i|I_q) = \frac{p(I_q|w_i)p(w_i)}{p(I_q)},$$

where $p(I_q|w_i)$ is the probability of $I_q$ being generated by $w_i$, $p(I_q)$ is the probability of the image $I_q$ and $p(w_i)$ is the probability of the word $w_i$. If the system has selected $w_{q1}, w_{q2}, \ldots, w_{qM_q}$ as annotations, taking into account the word-to-word correlations between the candidate word $w_i$ and the selected words $w_{q1}, w_{q2}, \ldots, w_{qM_q}$, then the conditional probability that $I_q$ is tagged with $w_i$ can be represented by $p(w_i|I_q, w_{q1}, w_{q2}, \ldots, w_{qM_q})$, which can be computed by applying Bayes’ rule:

$$p(w_i|I_q, w_{q1}, \ldots, w_{qM_q}) = \frac{p(w_i|I_q, w_{q1}, \ldots, w_{qM_q})p(I_q|w_i)}{p(w_{q1}, \ldots, w_{qM_q}|I_q)p(I_q)}.$$

For each candidate word $w_i$, we can assume that $I_q, w_{q1}, w_{q2}, \ldots, w_{qM_q}$ are independent of each other, and thus

$$p(I_q, w_{q1}, \ldots, w_{qM_q} | w_i) = p(I_q | w_i) \prod_{t=1}^{M_q} p(w_{qt} | w_i).$$

Then Eq. (22) can be written as

$$p(w_i|I_q, w_{q1}, \ldots, w_{qM_q}) = \frac{p(w_i|I_q)p(I_q)}{p(w_{q1}, \ldots, w_{qM_q}|I_q)p(I_q)},$$

where $|\cdot|$ is the number of images in the image set.
and \( p(w_i) \) can be estimated using the training set. Suppose that the total number of annotations on the training images is \( n \) and that the number of annotations using word \( w_i \) is \( n_i \); then \( p(w_i) \) can be estimated by

\[
p(w_i) = \frac{n_i}{n}.
\]  

(25)

Suppose that the CPAM-based feature vector of \( I_q \) is \( x_q \) and that the support vector described model that corresponds to \( w_i \) is \( m_i \); then the probability \( p(I_q|w_i) \) can be computed by the formulation described in Eq. (18), i.e., \( p(I_q|w_i) = p(x_q|m_i) \). For \( w_{q1}, w_{q2}, \ldots, w_{qM_q} \), the word-to-word correlations can be computed by Eqs. (19) and (20). In our implementation, we assume that \( p(w_{q1}, w_{q2}, \ldots, w_{qM_q} | I_q) \) and \( p(I_q) \) are kept constant across different candidate words.

The derivations of Eqs. (22)–(24) are based on the following anticipations and assumptions. Given a candidate word \( w_i \), for instance, \( w_i \) is “bridge”, we anticipate that if the image \( I_q \) contains bridge-like visual features, then the probability of \( I_q \) being tagged with \( w_i \) will be higher. And we anticipate that if the selected words often co-occurs with “bridge”, then the probability of \( I_q \) being tagged with “bridge” will be higher. We further assume that \( I_q, w_{q1}, w_{q2}, \ldots, w_{qM_q} \) are independent of each other. This assumption makes the problem more tractable from the computational perspective and yet it is sufficient for the problem at hand, as supported by our experiments. And it is also supported by the empirical results reported in [13].

Eqs. (21) and (24) follow a variant of the two-layer model described in Eq. (2), i.e., they operate by generating words directly from the given image. However, they are not identical to Eq. (2).

4.3. Annotation of untagged images

Let the set of distinct annotation words be \( \mathcal{W} = \{w_1, w_2, \ldots, w_K\} \). Given an untagged image \( I_q \), let the set of words that have been selected as annotations be the system be \( \mathcal{V} = \{w_{q1}, w_{q2}, \ldots, w_{qM_q}\} \), where \( M_q \) is the number of words in \( \mathcal{V} \). Fig. 5 shows the flowchart of the tagging process of SVIA. To annotate \( I_q \), its CPAM-based feature vector is extracted first. For each \( w_i \in \mathcal{W} - \mathcal{V} \), the conditional probability that \( I_q \) is tagged with \( w_i \) is then computed. If \( \mathcal{V} = \emptyset \), i.e., the system has not selected any words to use as annotation, then the conditional probability is computed by Eq. (21); otherwise, if \( \mathcal{V} \neq \emptyset \), i.e., the system has selected a set of words as annotations, then the conditional probability is computed by Eq. (24). By computing the conditional probability for the candidate words in the set \( \mathcal{W} - \mathcal{V} \), the words can be sorted in
descending order, and the top-ranked word is selected as the next annotation. Suppose that the top-ranked word is $w^*$; then we set $V = V \cup \{w^*\}$ and $M_q = M_q + 1$. This process is repeated until the number of selected words reaches a predefined number $k$.

5. Experimental studies

In this section, we present the results of experiments to evaluate the performance of the proposed SVIA. The experiments were performed on a PC with a Pentium IV 3.0 GHz processor and 4 GB of memory.

5.1. Image data sets

In the experiment, we selected as benchmarks the Corel5k data set [18], the Corel30k data set [20], and the Corel60k data set [25]. They are all from the Corel stock photograph collection and are widely used in evaluating image annotation methods.

The Corel5k data set contains 5000 images that are $192 \times 128$ pixels or $128 \times 192$ pixels. There are 371 distinct words in the vocabulary. Each image is tagged with one to five words, and the average number of words per image is 3.5. In the experiments of [18], the data set is divided into two parts. Out of the 5000 images, 4500 images are used for training and 500 images are used for testing. The testing set vocabulary contains 260 distinct words that are taken from the given vocabulary.

The Corel30k data set has properties similar to those of the Corel5k data set, except that it is substantially larger. It contains 31,695 images that are $384 \times 256$ pixels or $256 \times 384$ pixels. There are 5587 distinct words in the vocabulary. Each image is tagged with 1–5 words, and the average number of words per image is 3.6. In the experiments of [18] and [20], the data set is divided into training and testing sets with a ratio of 9:1. In the training set, only the words that are used as annotations by at least 10 images are considered. Therefore, the total number of words in the training set vocabulary is 1035. In the testing set, the vocabulary contains 950 words.

The Corel60k data set contains about 60,000 images that are $384 \times 256$ pixels or $256 \times 384$ pixels. There are 417 distinct words in the vocabulary. The images are assigned to 600 categories, and each category has about 100 images and represents a distinct topic of interest. Each category is tagged with 1–7 words that describe the category as a whole but do not accurately describe each individual image. In the experiments of [25], 80 images of each category are used for training, and the remaining 20 images of each category are used for testing.

Fig. 6 shows the thumbnails of some examples of randomly selected images from the Corel5k, Corel30k, and Corel60k data sets. As can be seen from Fig. 6, the data sets cover a wide variety of images, ranging from nature scenes to historical buildings to human activities; these reflect the diversity of the data sets. Since the application of our method is to tag images representative of daily life rather than special fields such as medicine or geography, the selected data sets are suitable for evaluating the performance of our method.

5.2. Experimental settings

Experimental settings are crucial for a performance evaluation. The first setting is for the selection of the training and testing images. For the Corel5k and Corel30k data sets, to make a fair comparison with the results reported in the literature, we used the same training and testing sets as those used in [18] and [20]; i.e., 90% of the images were used for training and 10% of the images were used for testing. For the Corel60k data sets, we randomly selected 80% of the images in each category for training and 20% of the images in each category for testing. Since the data set did not accurately describe each individual image, in our experiment, for each image, the ground truth tagging words are taken as the words that describe its category.

In our experiment, to fix the Gaussian bandwidth parameter $h$, described in Eq. (10), 15% of the entire data set was further randomly selected from the training data set. Among the selected images, 60% of them were used for training and 40% were used for validating. After the $h$ value was fixed, the entire training image set was used to train the support vector described models.

Basically, the performance of the system was evaluated by comparing the words tagged by the system with the words tagged by humans. Moreover, the precision and recall rates were used as measurements for evaluation. For each word $w$, let $n_t$ denote the number of images tagged by the system, let $n_r$ denote the number of ground truth related images, and let $n_c$ denote the number of images correctly annotated by the system; then the precision and recall rates can be computed by

$$
\text{precision}(w) = \frac{n_c}{n_t}, \quad \text{recall}(w) = \frac{n_c}{n_r}.
$$

We then compute the average precision and recall rates over all the words in the testing set vocabulary to evaluate the performance of
the system. Generally, there is a trade-off between precision and recall rate. For the testing images, when the number of words provided by the system increases, the recall rate will usually increase, whereas the precision rate will usually decrease. In our experiment, to compare the precision rates at different levels of recall rates, we changed the parameter $k$ (the number of words provided by the system, which was described in Section 4.3) from 1 to 15. Moreover, we also considered the coverage of the system, which can be calculated as follows. Denote the number of words with nonzero recall rate by $n^+$, and denote the number words in the testing set by $n$. Then the coverage rate can be computed by

$$\text{coverage} = \frac{n^+}{n}. \quad (27)$$

The coverage rate shows the proportion of images that are learned by the system. It provides an indication of the generalizability of the system.

5.3. Comparison to existing systems

For comparison, we selected the following systems, which have been evaluated for all or some of the selected data sets.

1. Supervised multiclass labeling (SML) [20]: SML poses annotation and retrieval as classification problems, where each class is defined as the group of database images labeled with a common semantic label. In this system, the minimum probability of error annotation and retrieval is computed by establishing a one-to-one correspondence between the semantic labels and the classes. The untagged images are then labeled based on the obtained probabilities.

2. Automatic linguistic indexing of pictures-real time (ALIPR) [25]: in ALIPR, a discrete distribution clustering method is used to group objects represented as bags of weighted vectors. A generalized mixture modeling technique for nonvector data is developed using the concept of hypothetical local mapping (HLM).

3. Graph learning model (GLM) [23]: in GLM, image-based graph learning is performed to obtain candidate annotations for each image. Next, word-based graph learning is used to refine the relationships between the images and the words to obtain the final annotations for each image. Moreover, two types of word correlations based on web-search results are applied.

4. Hybrid probabilistic model (HPM) [13]: the HPM integrates low-level image features and high-level user-provided tags to automatically tag images. For images without any tags, HPM predicts new tags based solely on the low-level image features. For images with user-provided tags, HPM jointly exploits both the image features and the tags in a unified probabilistic framework to recommend additional tags to label the images. Moreover, a collaborative filtering method based on the non-negative matrix factorization is developed for tackling the data sparsity issue.

The SML, ALIPR, and HPM are probabilistic modeling methods. They perform labeling tasks by computing the joint probabilities that an untagged image is labeled with the candidate annotation words. The GLM is performed by propagating the keywords from the tagged images to the untagged images by visual similarities. These systems have been shown to be successful and can obtain suitable annotation results. Hence, a comparison with them will demonstrate the performance of the proposed SVIA and will show if it is an improvement over existing systems.

5.4. Comparison results

5.4.1. Results for the Corel5k data set

This subsection presents the results of SVIA for the Corel5k data set. Table 1 presents the computation efforts involved in the experiment. The contents of the table include the description of the experiment, the scale of the experiment (i.e., the number of images used), and the computation time involved. We note that these computation times are not definitive since the computation time varies when the computer configuration is different. The reason that we show them here is to roughly illustrate the time complexity of the proposed algorithm.

Since the images in the Corel5k data set is tagged with 1–5 words, we show the results of SVIA when the system provides 5 words for each image. Fig. 7 compares system-tagged words and human-tagged words for some randomly selected testing images. As can be seen from Fig. 7, though the tagging performed by SVIA is not an exact match to human tagging, it is usually more plausible. Take as an example the second image in the top row. The words provided by SVIA are “beach, sand, tree, valley, sky”, and the human-provided words are “beach, palm, tree, people”. From the contents of the image, it can be seen that the words “sand, sky” are more plausible than the human-provided words.

Fig. 8 is a plot of precision–recall curves for annotation of the Corel5k testing data set using SVIA and SML. The curve generated by SML is taken from [20]. In the experiment of [20], the Gaussian mixture model combined with the discrete cosine transform representation (GMM–DCT representation) is adopted. The curve is generated when the dimension of the DCT feature space is 63.

From Fig. 8, it can be seen that the SVIA curve has the best precision around 0.37 when the recall is around 0.13, and its precision is superior to that of SML when the recall rate is below 0.35. There are also some levels of recall rates where SML obtains a better precision, especially when the recall rate is between 0.4 and 0.6. This is due to the fact that the system has to provide more words to obtain a higher recall rate, and the number of correct annotations does not increase as the number of system-provided words increases.

Table 2 presents the results of SVIA when the system provides 5 words for each image, and the results of studies using SML, GLM, and HPM. The contents of the table include the number of images in the testing set ($n_{\text{image}}$), the number of words in the testing set vocabulary ($n_{\text{word}}$), the average precision and recall rates over the entire testing set vocabulary (precision $\%$, recall $\%$), and the coverage rate of the system (coverage $\%$). In HPM, the system uses user-provided tags to enhance tagging accuracy. In order to present a fair comparison, we list the results of HPM under the Given 0 protocol, i.e., the results of HPM that do not use any user-provided tags as hints. From Table 2, it can be seen that SVIA obtains the best results in terms of recall and precision. The coverage rate of SVIA is lower than those of SML and HPM, and higher than that of GLM.

5.4.2. Results for the Corel30k data set

This subsection presents the results of SVIA for the Corel30k data set. The experiments that were conducted on the Corel5k
data set were repeated on the Corel30k data set. Table 3 presents the computation efforts involved in the experiment. The item meanings of Table 3 are the same as those in Table 1.

Fig. 9 is a plot of the precision–recall curves for the annotation of the Corel30k testing data set using SVIA and SML. The curve generated by SML is taken from [20]. In the experiment of [20], the GMM-DCT representation was adopted. The SML classifiers were learned using 3-level hierarchies. Fig. 9(a) and (b) compares the results of SVIA and ALIPR in terms of precision and recall rates, respectively; here, precision and recall rates are shown with the number of words provided by the system increasing from 1 to 15. It can be seen from Fig. 9(a) that the SVIA curve has the best precision around 0.36, and its precision and recall rates are significantly superior to that of SML.

Table 4 presents the results of SVIA when the system provided 5 words for each image, and the results of studies using SML and HPM. The item meanings of Table 4 are the same as those in Table 2. As in the experiments on the Corel5k data set, we list the results of HPM under the Given 0 protocol. From Table 4, it can be seen that SVIA obtains the best results in terms of recall and precision rate. The coverage rate of SVIA is lower than those of SML and HPM.

5.4.3. Results for the Corel60k data set

This subsection presents the results of SVIA for the Corel60k data set. The experiments that were conducted on the Corel5k and Corel30k data sets were repeated on the Corel60k data set. Table 5 presents the computation efforts involved in the experiment. The item meanings of Table 5 are the same as those in Table 1.

Fig. 10 compares the results SVIA, SML, and ALIPR for annotation of the Corel testing data set. Fig. 10(a) is a plot of precision–recall curves using SVIA and SML. In the experiment with SML, the GMM-DCT representation was adopted. The SML classifiers were learned using 3-level hierarchies. Fig. 10(a) and (b) compares the results of SVIA and ALIPR in terms of precision and recall rates, respectively; here, precision and recall rates are shown with the number of words provided by the system increasing from 1 to 15. It can be seen from Fig. 10(a) that the SVIA curve has the best precision around 0.40, and its precision and recall rates are significantly superior to that of SML.

Table 6 presents the results of SVIA when the system provides 5 words for each image, and the results of studies using SML and ALIPR, respectively. The item meanings of Table 6 are the same as those in Table 6. The coverage rate of SVIA is 42.6% when the system provides 5 words for each image. Since the coverage rates of SML and ALIPR are not reported in [20] and [25], they are not included in Table 6. From Table 6, it can be seen that SVIA obtains the best precision rate. The recall rate of SVIA is higher than that of SML and lower than that of ALIPR.

5.5. Discussion

The performance of SVIA on the Corel5k, Corel30k, and Corel60k data sets can be summarized as follows.

1. The tagging process took an average of 0.22 s for the experiments on the Corel5k, Corel30k, and Corel60k data sets.
Table 3
Computation efforts involved in the experiment on the Corel60k data set.

<table>
<thead>
<tr>
<th>Period</th>
<th>Description</th>
<th>Scale</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training process</td>
<td>Fixing parameter $h$</td>
<td>4279</td>
<td>443.1</td>
</tr>
<tr>
<td></td>
<td>(training and validating)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Training support vector described</td>
<td>28,525</td>
<td>2167.6</td>
</tr>
<tr>
<td>Tagging process</td>
<td>Average time for tagging one image</td>
<td>3170</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Fig. 9. Precision–recall curves for annotation of the Corel30k testing data set using SVIA and SML.

Table 4
Results of SVIA compared to other systems for the Corel30k testing data set.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$n_{image}$</th>
<th>$n_{word}$</th>
<th>Precise (%)</th>
<th>Recall (%)</th>
<th>Coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVIA</td>
<td>1035</td>
<td>950</td>
<td>27.5</td>
<td>28.1</td>
<td>41.1</td>
</tr>
<tr>
<td>SML</td>
<td>12.0</td>
<td>21.0</td>
<td>44.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPM</td>
<td>10.0</td>
<td>19.0</td>
<td>46.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Computation efforts involved in the experiment on the Corel60k data set.

<table>
<thead>
<tr>
<th>Period</th>
<th>Description</th>
<th>Scale</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training process</td>
<td>Fixing parameter $h$</td>
<td>7200</td>
<td>1439.2</td>
</tr>
<tr>
<td></td>
<td>(training and validating)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Training support vector described</td>
<td>48,000</td>
<td>6972.7</td>
</tr>
<tr>
<td>Tagging process</td>
<td>Average time for tagging one image</td>
<td>12,000</td>
<td>0.22</td>
</tr>
</tbody>
</table>

6. Conclusion

To deal with automatic or semi-automatic image annotation problems, one of the common ways is to build probabilistic models. In this paper, we proposed a probabilistic modeling algorithm. The algorithm has two major components, the training process and the tagging process. In the training process, a support-
A vector-based model was developed to describe clusters of images with manually tagged words. In the tagging process, a unified probabilistic framework was proposed to predict the annotation words. The performance of the proposed algorithm was tested on the Corel5k, Corel30k, and Corel60k data sets. The simulated results showed that the proposed algorithm yields higher precision and recall rates than the existing algorithms. The main contribution of this paper is the support-vector-based approach. The fact that the SVIA can exploit the advantage of support-vector-based approach for its ability to delineate cluster boundaries of arbitrary shape makes it particularly useful when the training images are not organized.

As an area of future work, we hope to find methods that work when there are fewer manually tagged training images and to develop techniques that can improve the coverage rates of the system.

Conflict of interest statement

There is no conflict of interest.

Acknowledgments

The authors are grateful to Kobus Barnard at the University of Arizona, Nuno Vasconcelos at the University of California, San Diego, and James Z. Wang at the Pennsylvania State University for providing the data sets. This work was partially supported by the Japan Society for the Promotion of Science (24500258), the National Natural Science Foundation of China (61103146, 61073075, 61272207), the High-Tech 863 Program (2012AA111902), the National Key Technology R&D Program of China (2011BAK02B02), the Fundamental Research Funds for the Central Universities of Dalian University of Technology, Ph.D. Program Foundation of MOE.
of China (2012006110094), and the Research Promotion Grant-in-Aid of Kochi University of Technology.

References


Liang Sun received the B.E. degree in computer science and technology from Xidian University, Xi’an, China, and the M.S. degree in computer application technology from Jilin University, Changchun, China, in 2003 and 2006, respectively. During 2006–2009, as a D.E. candidate, he was at College of Computer Science and Technology, Jilin University China. Since 2009–2012, as a D.E. candidate, he was at Kochi University of Technology (KUT), Japan, as an international student of cooperation between KUT and Jilin University. He received double D. degree from Kochi University and Jilin University in March, 2012 and June 2012, respectively. He is currently a lecturer in the College of Computer Science and Technology, Dalian university of technology, Dalian, China. His main research interests lie in computational intelligence and machine learning methods.

Hongwei Ge received B.S. and M.S. degrees in mathematics from Jilin University, China, and the Ph.D. degree in computer application technology from Jilin University, in 2006. He is currently a vice professor in the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests are computational intelligence, machine learning, optimization and modeling, system control. He has published more than 50 papers in these areas. His research was featured in the IEEE Transactions on Systems, Man, and Cybernetics, the Computers and Structures, the Nonlinear Analysis: Real World Applications, the Advances in Soft Computing and the Neuro-computing.

Shinichi Yoshida is an associate professor in School of Information, Kochi University of Technology. He received his B.E. from Chuo University in 1996. He received Dr. Eng. from Tokyo Institute of Technology in 2001. From 2001 to 2004, he was an assistant professor of Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology. From 2004 to 2007, he was an assistant professor of Information Systems Engineering, H. Sophia Gakuen University. Since 2007, he has been an assistant professor of Department of Information Systems Engineering. Since 2009, he has been an assistant professor of School of Information, Kochi University of Technology. Since 2013, he has been an associate professor of School of Information, Kochi University of Technology. His interest covers multimedia information retrieval based on softcomputing and computational intelligence. He is a member of IEICE, IEEE, IPSJ, and SOFT.
Yanchun Liang received the Ph.D. degree in applied mathematics in 1997 from Jilin University, Changchun, China, where he is currently a professor in the College of Computer Science and Technology. He was a visiting scholar at Manchester University of United Kingdom from 1990 to 1991, a visiting professor at the National University of Singapore from 2000 to 2001, a guest professor in the Institute of High Performance Computing of Singapore from 2002 to 2004, and a guest professor at Trento University, Italy, from 2006 to 2008, and 2010. He has published more than 300 papers. His research was featured in the Bioinformatics, IEEE Transactions on SMCA, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Geoscience and Remote Sensing, Journal of Micromechanics and Microengineering, Physical Review E, Smart Materials and Structures, Applied Artificial Intelligence, etc. He was the recipient of several grants from NSFC, EU, etc. His research interests include computational intelligence, machine learning methods, text mining, MEMS modeling, and bioinformatics.

Guozhen Tan received the B.S. degree from Shenyang university of Technology, Shenyang, M.S. degree from Harbin Institute of Technology, Harbin, and the Ph.D. degree from Dalian University of Technology, Dalian, PR China. He was a visiting scholar with the Department of Electrical and Computer Engineering of University of Illinois at Urbana-Champaign, IL, USA, from January 2007 to 2008. He has been a Professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, PR China, and Director of Engineering and Technology Research Center for the Internet of things and Collaborative Sensing, Liaoning province, and currently is Dean of the College of Computer Science and Technology, Dalian University of Technology, Dalian. His research interests include the Internet of things (IoT), cyber-physical systems (CPS) VANET, Intelligent transportation systems(TIS), network optimization algorithms, etc. He was the recipient of the 2006 National Science and Technology Progress Award (second class) for his work in vehicle position and navigation, location-based service, traffic signal control, rapid despondences and processing for traffic emergency. Dr. Tan has been an Editor for the Journal of Chinese Computer Systems. He has been the member of China Computer Federation(CCF), committer of Professional Committee of Software Engineering of CCF, and committer of Professional Committee of High Performance Computing of CCF.