A Sparse Support Vector Machine Approach to Region-Based Image Categorization

Jinbo Bi
Computer Aided Diagnosis and Therapy Solutions
Siemens Medical Solutions, Inc.
51 Valley Stream Parkway, Malvern, PA 19355
jinbo.bi@siemens.com

Yixin Chen
Department of Computer Science
University of New Orleans
New Orleans, LA 70148
yixin@cs.uno.edu

James Z. Wang
School of Information Sciences and Technology
The Pennsylvania State University, State College, PA 16802
jwang@ist.psu.edu

Abstract

Automatic image categorization using low-level features is a challenging research topic in computer vision. In this paper, we formulate the image categorization problem as a multiple-instance learning (MIL) problem by viewing an image as a bag of instances, each corresponding to a region obtained from image segmentation. We propose a new solution to the resulting MIL problem. Unlike many existing MIL approaches that rely on the diverse density framework, our approach performs an effective feature mapping through a chosen metric distance function. Thus the MIL problem becomes solvable by a regular classification algorithm. Sparse SVM is adopted to dramatically reduce the regions that are needed to classify images. The selected regions by a sparse SVM approximate to the target concepts in the traditional diverse density framework. The proposed approach is a lot more efficient in computation and less sensitive to the class label uncertainty. Experimental results are included to demonstrate the effectiveness and robustness of the proposed method.

1. Introduction

Designing a computer algorithm to classify images into predefined categories is an important yet challeging research topic. It finds applications in a variety of fields including biomedicine, digital libraries, Web searching, and surveillance systems. There has been an abundance of prior work on automated image analysis. The works reviewed below are most relevant to what we propose in this article, which by no means represent the complete list.

1.1. Related Work

Loosely speaking, current image classification algorithms can be divided into two groups according to the imagery features used in the classification: *global approaches* and *component-based approaches*. The global image classification approaches use features that characterize the global information of an image. For example, k-nearest neighbor classifier on color histograms was proposed to discriminate *indoor* versus *outdoor* images [19]. Bayesian classifiers using edge directions histograms were implemented to organize *city* and *landscape* images [21]. Support Vector Machines (SVMs) built on color histograms were applied to classify images containing a generic set of objects [4].

Although the global features can usually be computed with little cost and are effective for certain classification tasks, some visual contents of images could only be locally defined. A number of component-based approaches have been proposed to exploit local and spatial properties of an image. In the method introduced by Gorkani and Picard [9], an image is first divided into 16 non-overlapping equal-sized blocks with dominant orientations computed for each block. The image is then classified as *city* or *suburb* scenes as determined by the majority orientations of blocks. In the ALIP system [12], a concept corresponding to a particular category of images is captured by a statistic model trained on color and texture features of image blocks. In this model, spatial relations among blocks and across image resolutions are both taken into consideration.

A rigid partition of an image into fixed-size blocks often breaks an object into several blocks or puts different objects into a single block. Thus visual information about objects may be destroyed by a rigid partition. Image segmentation is one way to extract object information [8]. It decomposes an image into a collection of regions, which corresponds to objects if decomposition is ideal. Image segmentation has been successfully applied to image classification. Smith and Li [17] proposed a method for classifying images by spatial orderings of regions where each region is represented by a symbol corresponding to an entry in a pattern library. Barnard et al. [1] proposed a method of relating words to images based on regions. In their method, an image is modeled as a sequence of regions and a sequence of words generated by a hierarchical statistic model, which describes the occurrence and co-occurrence of region features and object names. Fergus et al. [7] developed a mixture-model-based method to recognize object classes, such as *motorbikes*, *airplanes*, *faces*, *cars*, and *spotted cats*. Their model is built upon scale invariant regions generated by an entropy-based feature detector. Image segmentation is not involved in the feature extraction process.

Recently, Chen and Wang proposed a region-based image classification algorithm [5] based on a technique that extends Multiple-Instance Learning [6]. In their method, a collection of region prototypes are learned according to a Diverse Density (DD) function [13]. Each region prototype represents a class of regions that is more likely to appear in images with a specific label than with other labels. An image is then summarized by a collection of features each defined by a region prototype. A standard SVM is trained over the image features. Their method compares favorably with two other SVM-based algorithms on a set of $2,000$ images. However, as pointed by the authors in [5], the performance is sensitive to noise in the negative images because the DD function is a multiplicative model.

1.2. Overview of Our Approach

Following Chen and Wang's work [5], we formulate the image classification problem as a multiple-instance learning (MIL) problem by regarding an image as a bag of regions (instances). We distinguish images with a specific label (the positive class) from images in all other categories (the negative class). Instead of applying a diverse density framework as in [5] to learn a set of region prototypes, a feature mapping based on an appropriate distance metric is adopted to map each image, a bag of regions, to a feature vector whose dimension equals to the total number of regions from all positive bags in the training set. Then the image classification problem becomes solvable by a regular classification algorithm. However, the feature mapping produces a possibly high dimensional space when the number of regions in positive bags is large. It is hence essential and indispensable to select a subset of mapped features that is most relevant to the image classification problem of interest. Support vector machines (SVM) have been proven to be powerful and robust tools for tackling classification tasks. We propose to use the 1-norm SVM which produces sparser solutions (less features will be used) than standard SVMs. Our approach is shown to be as effective as or comparable to DD-SVM in [5] but is a lot more efficient in computation and less sensitive to the noise in the class labeling. Note that the proposed MIL framework is independent of the specific forms of region features. The proposed framework can be applied to regions obtained from image segmentation as well as features generated by various region detectors such as affine region detectors described in [11, 14, 15, 16, 20].

The rest of the paper is organized as follows. We first briefly review the image segmentation approach we use to obtain image regions to represent an image in Section 2. Pertinent notations are also introduced. Section 3 describes the distance feature mapping, which we call region distance feature mapping, and provides a geometric motivation that explains why the proposed mapping makes sense. Section 4 is dedicated to the description of the 1-norm SVM and a concrete sparse SVM formulation is given to construct classifiers and select features. Experimental design and results are presented in Section 5 to show the effectiveness and robustness of our approach. In the last section, we conclude and discuss possible future work.

2. Image Segmentation

Since we will compare, in Section 5, the proposed method with the DD-SVM approach given in [5], we adopt the same image segmentation algorithm as described in [5]. A brief summary is given as follows.

To segment an image, the system first partitions the image into non-overlapping blocks of size 4×4 pixels. A feature vector is then extracted for each block. Each feature vector consists of 6 features. Three of them are the average color components in a block. The LUV color space is used, where L encodes luminance, and U and V encode color information (chrominance). The other three represent square root of energy in the high-frequency bands of the wavelet transforms, i.e., the square root of the second order moment of wavelet coefficients in high frequency bands. The coefficients in different frequency bands show variations in different directions, hence capture the texture properties.

To calculate these moments, a Daubechies-4 wavelet transform is applied to the L component of the image. After one-level wavelet transform, a 4×4 block is decomposed into four frequency bands: the LL (low low), LH (low high), HL, and HH bands, each containing 2×2 coefficients. If the coefficients in the HL band are given as $c_{i,j}$ where $i, j = 1, 2$, then a feature is defined as $f = \left(\frac{1}{4}\sum_{i=1}^{2}\sum_{j=1}^{2} c_{i,j}^2\right)^{\frac{1}{2}}$. The other two features are computed similarly from the LH and HH bands.

A modified k-means algorithm is applied to group the feature vectors into clusters each corresponding to a region in the segmented image. The algorithm does not require the number of clusters be specified. Instead, the number of clusters gradually increases until a stop criterion is met. The number of regions in an image is not a constant. The average number of regions per image changes in accordance with the adjustment of the stop criteria. After segmentation, three extra features are computed for each region to describe shape properties. They are normalized inertia of order 1, 2, and 3. As a result, each region in any image is characterized by a 9 dimensional feature vector $\mathbf{x} = [x_1, \ldots, x_9]^T$ where $[x_1, \ldots, x_6]^T$ is the mean of the set of feature vectors (color and texture features) associated with the region, and $[x_7, x_8, x_9]^T$ contains three shape features.

We denote images in the positive class as $\mathbf{x}_i^+$, and the j^{th} region in the i^{th} image as $\mathbf{x}_{ij}^+$. The vector $\mathbf{x}_{ij}^+$ is in the 9-dimensional feature space as introduced in the above paragraph. We denote this space by $\mathbb{X}$. The image $\mathbf{x}_i^+$ consists of n_i^+ regions $\mathbf{x}_{ij}^+, j = 1, \cdots, n_i^+$. The total number of regions in all positive images is n so $n = \sum_{i=1}^{\ell^+} n_i^+$ where ℓ^+ determines the number of images in the positive class. In a MIL problem, an image is labeled positive if at least one of the regions in it is positive while an image is labeled negative only if all the regions in it are negative. To distinguish the positive regions from the regions that appear in a positive image, we call any region shown in a positive image a p-region or a p-instance. For the sake of convenience, when we line up all p-regions in positive images together, we re-index all these regions as $\mathbf{x}^t, t = 1, \cdots, n$. Likewise, $\mathbf{x}_i^-$ represents an image in the negative class, and $\mathbf{x}_i^-$ contains n_i^- regions $\mathbf{x}_{ij}^-, j = 1, \cdots, n_i^-$. Other notations follow as how we define for positive images.

3. Region Distance Representation of Images

We introduce the region distance feature mapping in this section. Before talking about the mathematical definition of the mapping, we first give a brief retrospect of the diverse density framework that forms the conceptual basis of the region distance feature mapping. A geometric motivation is provided to further show meaningfulness of the mapping.

3.1. Retrospect of Diverse Density

The diverse density framework is derived in [13] based on the assumption that there exists a single target concept which can be used to label individual instances correctly. An instance is represented as a point in the $\mathbb{X}$ input space. Suppose that all instances in an image can trace out a continuous manifold or a path in the $\mathbb{X}$ space. A reasonable guess for the location of the target concept in the $\mathbb{X}$ space is a point where all positive manifolds intersect without intersecting any negative manifolds. Since an entire manifold cannot be obtained in practice and only some arbitrary sample is drawn from the manifold, a practical calculation of the target concept is to find the area in $\mathbb{X}$ space that attains a high density of positive instances (p-instances) and low density of negative instances (n-instances).

By the maximum likelihood analysis assuming images are conditionally independent given the target concept, a probabilistic measure of diverse density is derived. Then an optimization algorithm, such as the gradient descent approach as used in [13], or EM-DD as used in [22], is used to maximize this probabilistic measure for the target concept. The process of maximization is often intensively time-consuming (requiring hours or even days of running time on datasets of reasonable size such as Musk data [13]). In addition, both gradient descent algorithm and EM-DD cannot guarantee the global optimality and hence they may get stuck at local solutions. Frequently, multiple runs with different starting search points would be needed.

3.2. Region Distance Mapping

In our approach, rather than searching a vector in the entire $\mathbb{X}$ space to achieve the maximal product of high likelihood of p-instances and low likelihood of negative instances in the diverse density framework, our idea is to find a p-instance which is close to instances from positive bags and far from instances in negative bags. We call such a p-instance prototype p-instance. An ideal prototype p-instance is a positive instance which has a small distance to at least one instance from each of the positive bags and has large distance to all instances from negative bags. Similar to the diverse density framework where it is not necessary that all positive bags intersect at a single target concept, there can exist multiple prototype p-instances. This framework amounts to a discretized version of the diverse density function with grid points at the sample p-instances.

To solve the hence-created discretized problem, we first define the region distance mapping $\mathbf{d}$ which maps an image or a bag of regions into a feature space $\mathbb{F}$ based on a metric distance function $m : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$. The space $\mathbb{F}$ is an n dimensional space where each region in the positive image corresponds to a dimension or a feature. Note that the metric distance m can be any suitable metric although Euclidean distance is commonly employed. The distance function should be chosen properly by taking into account prior knowledge or domain insights about the image classification tasks to be solved. The mapping $\mathbf{d}$ is defined as

$$\mathbf{d}(\mathbf{x}_i) = \left[d(\mathbf{x}_i, \mathbf{x}^1) d(\mathbf{x}_i, \mathbf{x}^2), \cdots d(\mathbf{x}_i, \mathbf{x}^n) \right]^T \ .$$

where the function d maps a pair of (a bag, an instance) to a real number, and $d(\mathbf{x}_i, \mathbf{x}^t) = \min_{j=1,\cdots,n_i} m(\mathbf{x}_{ij}, \mathbf{x}^t),\ t = 1, \cdots, n$. Notice that $\mathbf{x}^t$ are those re-indexed p-instances as defined in Section 2, and $\mathbf{x}_i$ here represents a bag (or an image) from either the positive class or the negative class. Clearly, the t^{th} feature in $\mathbb{F}$ is determined by the t^{th} p-instance and it measures the distance between the t^{th} p-instance and the "manifold" corresponding to a given image $\mathbf{x}_i$ where the distance is defined as the distance from $\mathbf{x}^t$ to the instance in $\mathbf{x}_i$ that is the closest to $\mathbf{x}^t$.

For a given training set of ℓ^+ positive images and ℓ^- negative images, applying the above mapping yields the following matrix $\mathbf{D} =$

$$\begin{bmatrix} \mathbf{d}_1^+ \\ \vdots \\ \mathbf{d}_{\ell^+}^+ \\ \mathbf{d}_1^- \\ \vdots \\ \mathbf{d}_{\ell^-}^- \end{bmatrix} = \begin{bmatrix} d(\mathbf{x}_1^+, \mathbf{x}^1) & d(\mathbf{x}_1^+, \mathbf{x}^2) & \cdots & d(\mathbf{x}_1^+, \mathbf{x}^n) \\ \vdots & \vdots & \ddots & \vdots \\ d(\mathbf{x}_{\ell^+}^+, \mathbf{x}^1) & d(\mathbf{x}_{\ell^+}^+, \mathbf{x}^2) & \cdots & d(\mathbf{x}_{\ell^+}^+, \mathbf{x}^n) \\ d(\mathbf{x}_1^-, \mathbf{x}^1) & d(\mathbf{x}_1^-, \mathbf{x}^2) & \cdots & d(\mathbf{x}_1^-, \mathbf{x}^n) \\ \vdots & \vdots & \ddots & \vdots \\ d(\mathbf{x}_{\ell^-}^-, \mathbf{x}^1) & d(\mathbf{x}_{\ell^-}^-, \mathbf{x}^2) & \cdots & d(\mathbf{x}_{\ell^-}^-, \mathbf{x}^n) \end{bmatrix}.$$

3.3. Geometric Motivation

An intuitive geometric interpretation often allows readers to easily grasp the fundamentals of an approach. Hence we provide a geometric motivation to show that a prototype p-instance determines a feature in $\mathbb{F}$ that has strong capability of discriminating between positive images and negative images and is expected to be selected by a reasonably good feature selection algorithm.

As discussed in the above section, for a given set of training examples, the t^{th} feature in $\mathbb{F}$ realizes the distance values from the t^{th} positive region $\mathbf{x}^t$ to each of the positive images and also to each of the negative images. It presents as the t^{th} column of the matrix $\mathbf{D}$. If a p-instance is approximate to the target concept in the diverse density framework, it attains small distance values to positive images and large distance values to negative images as illustrated in Figure 1. Then the corresponding column in $\mathbf{D}$ has small values in the top half vector associated with positive images and large values in the bottom half vector associated with negative images. Thus this feature has the power to distinguish positive images from negative images.

4. 1-norm Support Vector Machines

Further investigation of the proposed mapping reveals that the region distance mapping also has possibility to produce irrelevant features or redundant features. For example, in Figure 1, the point $\mathbf{x}^1$ does not represent an intersection of any positive images (paths), it can barely be a prototype p-instance. Thus it is likely that the feature induced by $\mathbf{x}^1$ is irrelevant to the search of a target concept or a prototype p-instance. Moreover, there may exist multiple p-instances close to an individual target concept or an intersection of several positive bags. Some of the features induced by these p-instances may turn out to be redundant because they can be highly correlated. Furthermore, when the total number of p-instances n is large, the dimension of $\mathbb{F}$ can be greater than the amount of available training images, which consequently causes the problem of the curse of dimensionality. To alleviate all these problems, one of the most extensively used methods is *feature selection*.

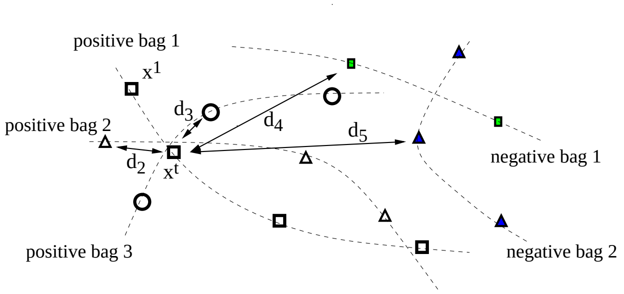

Figure 1. The dotted lines represent manifolds corresponding to 3 positive and 2 negative bags. Various points are samples from the manifolds. The t^{th} feature induced by $\mathbf{x}^t$ realizes the t^{th} column $(d_1, d_2, d_3, d_4, d_5)$ in $\mathbf{D}$ where $d_1 = 0$ since $\mathbf{x}^t$ is an instance in Positive bag 1, and d_2 to d_5 are illustrated in the figure. This feature shows the difference between positive and negative images since d_1, d_2, d_3 are much smaller than d_4 and d_5.

Existing feature selection approaches generally fall in two categories: filter and wrapper. Some filter methods such as ranking through correlation coefficients or through Fisher scores tend to select inter-correlated features and does not guarantee an acquisition of a good classifier. On the contrary, wrappers include the desired classifier as a part of their performance evaluation, and they tend to produce better generalization but may require an expensive computational cost. Our method is basically a wrapper where the 1-norm SVM is used to construct classifiers and select important features simultaneously. The 1-norm SVM can be formulated as a linear program (LP) which, in general, can be solved efficiently from the optimization point of view, so computational cost will not be an issue.

SVMs construct classifiers based on hyperplanes by minimizing a regularized training error, i.e., $\lambda P[\cdot] + error$ where $P[\cdot]$ is a regularization operator, λ is called the regularization parameter, and $error$ is commonly defined through a hinge loss function. Another main characteristic of SVMs is the use of appropriate kernel mappings. Notice that features in $\mathbb{F}$ have a specific geometric meaning as discussed in Section 3. We will not map features in $\mathbb{F}$

to any other space through a kernel in the feature selection process in order to fully explore these region distance features. We consider the classification problem of finding a linear decision boundary $\mathbf{w}'\mathbf{d} + b = 0$ in the feature space $\mathbb{F}$ to classify between positive images and negative images where $\mathbf{w},\ b$ are model parameters and $\mathbf{w}'$ denotes the transpose of $\mathbf{w}$. When an optimal solution $\mathbf{w}$ is obtained, the magnitude of its component w_t indicates the significance of the effect of the t^{th} feature in $\mathbb{F}$ on the classifier. Those features corresponding to a non-zero w_t are selected and used in the classifier.

Denote the class label variable by y and y takes values of +1 and -1. The hinge loss function $\xi = \max\{1 - y(\mathbf{w}'\mathbf{d} + b), 0\}$ is employed by standard SVMs to define a training error metric. The regularization operator in standard SVMs is the squared 2-norm $||\mathbf{w}||$ of the weight vector $\mathbf{w}$, which formulates SVMs as quadratic programs (QP). Solving QPs is typically computationally more expensive than solving linear programs (LPs). SVMs can be transformed into LPs as in [2, 18, 23]. This is achieved by regularizing with a sparse-favoring norm, e.g. the 1-norm $||\mathbf{w}||_1 = \sum |w_t|$. Thus 1-norm SVM is also referred to as sparse SVM and has been similarly applied to other practical problems such as drug discovery in [3].

Another issue worthy of mention is that in many MIL problems, especially in our image categorization problems, the number of negative examples can be much larger than the number of positive examples since only images in a specific category are labeled positive and all images from other categories are labeled negative. The problem becomes rather imbalanced. To tackle this imbalanced issue and make classifiers biased towards the minor class, a simple strategy we used is to penalize differently on errors produced respectively by positive examples and by negative examples. Hence the 1-norm SVM is formulated as follows:

$$\begin{array}{ll} \min & \lambda \sum_{t=1}^{n} |w_t| + C_1 \frac{1}{\ell^+} \sum_{i=1}^{\ell^+} \xi_i + C_2 \frac{1}{\ell^-} \sum_{j=1}^{\ell^-} \eta_j \\ \text{s.t.} & (\mathbf{w}'\mathbf{d}_i^+ + b) + \xi_i \geq 1, \quad i = 1, \ldots, \ell^+, \\ & -(\mathbf{w}'\mathbf{d}_j^- + b) + \eta_j \geq 1, \ \ j = 1, \ldots, \ell^-, \\ & \xi_i,\ \eta_j \geq 0,\ i = 1, \ldots, \ell^+, j = 1, \ldots, \ell^-. \end{array} \tag{1}$$

Choosing different values for parameters C_1 and C_2 will penalize differently on errors in one class versus errors in the other class. Usually C_1 and C_2 are chosen so that the training error is determined by a convex combination of the training errors occurred on positive examples and on negative examples. In other words, let $C_1 = \mu$ and $C_2 = 1 - \mu$ where $0 < \mu < 1$.

Category ID	Category Name
0	African people and villages
1	Beach
2	Historical building
3	Buses
4	Dinosaurs
5	Elephants
6	Flowers
7	Horses
8	Mountains and glaciers
9	Food
10	Dogs
11	Lizards
12	Fashion models
13	Sunset scenes
14	Cars
15	Waterfalls
16	Antique furniture
17	Battle ships
18	Skiing
19	Desserts

Table 1. The 20 image categories.

To form a LP for the 1-norm SVM, we rewrite $w_t = u_t - v_t$ where $u_t,\ v_t \geq 0$. If either u_t or v_t has to equal to 0, then $|w_t| = u_t + v_t$. Then the LP is formulated in variables $\mathbf{u}$, $\mathbf{v}$, b, $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ as

$$\begin{array}{ll} \min & \lambda \sum_{t=1}^{n} (u_t + v_t) + \frac{\mu}{\ell^+} \sum_{i=1}^{\ell^+} \xi_i + \frac{1-\mu}{\ell^-} \sum_{j=1}^{\ell^-} \eta_j \\ \text{s.t.} & \big((\mathbf{u} - \mathbf{v})'\mathbf{d}_i^+ + b\big) + \xi_i \geq 1, \quad i = 1, \ldots, \ell^+, \\ & -\big((\mathbf{u} - \mathbf{v})'\mathbf{d}_j^- + b\big) + \eta_j \geq 1, \ \ j = 1, \ldots, \ell^-, \\ & u_t, v_t \geq 0, \quad t = 1, \ldots, n, \\ & \xi_i,\ \eta_j \geq 0,\ i = 1, \ldots, \ell^+, j = 1, \ldots, \ell^-. \end{array} \tag{2}$$

Solving LP (2) yields solutions equivalent to those obtained by the 1-norm SVM (1) because any optimal solution to (2) has at least one of the two variables u_t, v_t equal to 0 for all $t = 1, \ldots, n$. Otherwise, assume $u_t > v_t > 0$ without loss of generality, and we can find a better solution by setting $u_t = u_t - v_t$ and $v_t = 0$, which contradicts the optimality of $(\mathbf{u}, \mathbf{v})$.

5. Experimental Results

We evaluate the proposed MIL learning framework based on the same data set as used in [5]. Section 5.1 describes the experimental setup, including information of the image data set and the implementation details. Section 5.2 compares our MIL framework to DD-SVM [5] in terms of categorization accuracies. Section 5.3 analyzes the effects of training label uncertainty on the algorithm performance. Computational issues are discussed in Section 5.4.

	Cat. 0	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8	Cat. 9
Cat. 0	**63.6**%	1.2%	10.4%	1.6%	1.2%	11.6%	2.4%	3.6%	0.4%	0.4%
Cat. 1	4.0%	**62.8**%	6.4%	4.4%	0.8%	4.4%	0.8%	0.0%	14.8%	1.6%
Cat. 2	8.0%	3.6%	**67.6**%	4.0%	0.8%	7.2%	0.8%	0.0%	0.4%	1.6%
Cat. 3	1.2%	2.8%	2.8%	**86.8**%	0.0%	0.8%	0.4%	0.4%	0.4%	4.4%
Cat. 4	0.4%	0.4%	0.0%	0.0%	**97.6**%	0.8%	0.0%	0.0%	0.0%	0.8%
Cat. 5	6.4%	3.6%	6.8%	0.0%	2.4%	**69.6**%	0.0%	3.2%	7.2%	0.8%
Cat. 6	2.0%	1.2%	0.0%	0.0%	0.0%	0.0%	**94.4**%	0.0%	0.4%	2.0%
Cat. 7	3.2%	2.0%	2.8%	0.0%	0.0%	1.2%	1.2%	**89.6**%	0.0%	0.0%
Cat. 8	5.2%	15.2%	7.2%	1.2%	3.2%	7.2%	0.8%	0.0%	**58.8**%	1.2%
Cat. 9	7.6%	2.4%	0.4%	3.2%	0.4%	2.4%	1.6%	2.4%	0.4%	**79.2**%

Table 2. The confusion matrix of image categorization experiments (over 5 randomly generated test sets). Each row lists the average percentage of images (test images) in one category classified to each of the 10 categories. Numbers on the diagonal show the classification accuracies.

5.1. Experimental Setup

The image data set consists of 2,000 images taken from 20 CD-ROMs published by COREL Corporation. Each COREL CD-ROM contains 100 images representing a distinct concept. Therefore, the data set has 20 thematically diverse image categories, each containing 100 images. All the images are in JPEG format of size 384×256 or 256×384. The category names are listed in Table 1 along with the identifiers (IDs) for the 20 categories. Since the classification problem is multi-class, we use the one-against-the-rest strategy.

In our experiments, images within each category were randomly partitioned in half to form a training set and a test set. We repeated each experiment for 5 random splits, and reported the average of the results obtained over 5 different test sets. The parameter μ and λ in sparse SVM were selected according to a twofold cross-validation on the training set. We chose μ from 0.1 to 2.0 with step size 0.1 and λ from $\{0.1,\ 1,\ 10,\ 100\}$. We found that $\mu = 0.5$ and $\lambda = 1$ give the minimum twofold cross-validation error. Therefore, we fix $\mu = 0.5,\ \lambda = 1$ in all subsequent experiments. The linear program of 1-norm SVM was solved using CPLEX version 6.6 [10].

5.2. Categorization Results

We first report the confusion matrix of the proposed method in Table 2 based on images in Category 0 to Category 9, i.e., 1,000 images. Each row lists the average percentages of images in a specific category classified to each of the 10 categories. The numbers on the diagonal show the classification accuracy for each category and off-diagonal entries indicate classification errors. A detailed examination of the confusion matrix shows that two of the largest errors (the underlined numbers in Table 2) are errors between Category 1 (Beach) and Category 8 (Mountains and glaciers): 15.2% of *Mountains and glaciers* are misclassified as *Beach*; 14.8% of *Beach* images are misclassified as *Mountains and glaciers*. This observation is in line with that presented in [5]. As stated in [5], the high classification errors are due to the fact that many images from these two categories have regions that are semantically related and visually similar, such as regions corresponding to mountains, river, lake, and ocean.

We compare the overall prediction accuracy of 1-norm SVM with that of DD-SVM. The average classification accuracies over 5 random test sets and the corresponding 95% confidence intervals are provided in Table 3. For the 1000-image data set, the performance of DD-SVM is slightly better than 1-norm SVM. As the number of categories in the data set increases to 20, the 1-norm SVM performs comparably well relative to DD-SVM. Although the average accuracy of DD-SVM is slightly higher than that of 1-norm SVM, the difference is not statistically significant as indicated by the 95% confidence intervals.

5.3. Sensitivity to Label Uncertainty

We also compared the proposed learning framework with DD-SVM in terms of the sensitivity to label uncertainty. In terms of binary classification, we define the label uncertainty as the probability that an image is mislabeled. In this experiment, training sets with different levels of label uncertainty are generated as follows. We first randomly pick $d\%$ of positive images and $d\%$ of negative images from a training set. Then, we modify the labels of the selected images by negating their labels, i.e., positive (negative) images are labeled as negative (positive) images. Finally, we put these images with new labels back to the training set. The new training set has $d\%$ of images with negated labels.

We compare the accuracies between our framework and DD-SVM for $d = 0,\ 2,\ 4,\ 6,\ 8,$ and 10 based on 200 images from Category 2 (Historical buildings) and Category 7

	Average Accuracy : [95% confidence interval]	
	1000 images (10 categories)	2000 images (20 categories)
1-norm SVM	77.0% : $[76.9\%, 78.1\%]$	65.7% : $[64.7\%, 66.7\%]$
DD-SVM	81.5% : $[78.5\%, 84.5\%]$	67.5% : $[66.1\%, 68.9\%]$

Table 3. Image categorization performance of 1-norm SVM and DD-SVM. The numbers listed are the average classification accuracies over 5 random test sets and the corresponding 95% confidence intervals. The 1000-image data set contains images from Category 0 to Category 9. The 2000-image data set contains images from all 20 categories. Training and test sets are of equal size.

(Horses). The training and test sets have equal size. The average classification accuracies (over 5 randomly generated test sets) are presented in Figure 2. From Figure 2, DD-SVM outperforms 1-norm SVM by 2% in terms of classification accuracy on average at the lower level of label uncertainty. As the uncertainty level increases to 10%, the performance difference is reversed: the average classification accuracy of 1-norm SVM is 2% higher than that of DD-SVM. As d changes from 0% to 10%, the average classification accuracy of DD-SVM decreases 8.1%, while the average classification accuracy of 1-norm SVM only decreases 5%. When we extend this experiment to the 1000-image data set, the average classification accuracy of 1-norm SVM for 10 categories over 5 randomly generated test sets is 75.1% with the 95% confidence interval $[73.6\%, 76.6\%]$, showing the robustness of the 1-norm SVM. In contrast, the diverse density function in DD-SVM is very sensitive to instances in negative bags. The diverse density value at a point is exponentially reduced if there is a single instance from a negative bag close to the point. Consequently, the region prototypes in DD-SVM, which are defined as the local maximizers of the DD function, are sensitive to label uncertainties. We abandon the process of maximizing the DD function in our approach, so our approach is not sensitive to label uncertainties due to the appropriate regularization.

5.4. Speed

On average, the learning of each 1-norm SVM classifier using a training set of 500 images (4.31 regions per image) takes less than 5 seconds of CPU time on a Pentium III 700MHz PC running the Linux operating system. The training for a DD-SVM using the same training set on the same computer system takes around 40 minutes of CPU time.

The time required in testing depends on the number of nonzero elements in $\mathbf{w}$ (for 1-norm SVM) or the number of region prototypes (for DD-SVM). In our experiments, the average number of selected variables in 1-norm SVM is around 50, which is slightly smaller than the average number (53.8) of region prototypes in DD-SVM.

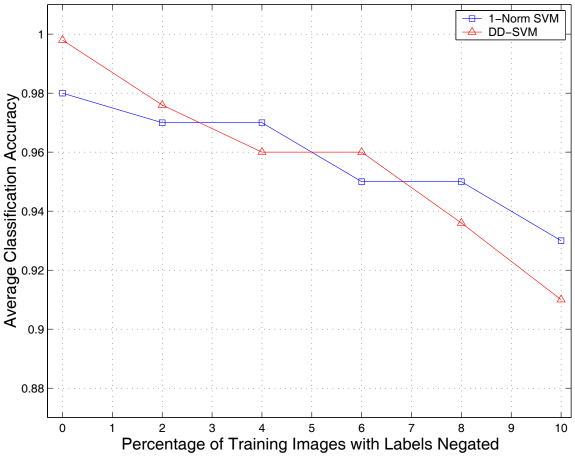

Figure 2. Comparison between DD-SVM and 1-norm SVM as the labeling uncertainty of training images varies.

6. Conclusions and Future Work

In this paper, we proposed a region-based image categorization method using a new formulation of Multiple-Instance Learning. Unlike many existing MIL approaches that rely on the diverse density framework, our approach performs an effective feature mapping by the definition of a distance metric between an instance and a bag. Thus the MIL problem can be solved by a regular classification algorithm, such as sparse SVM. We adopt a 1-norm SVM formulation to dramatically reduce the regions that are needed to classify images. The selected regions in our framework approximate to the target concepts in the traditional diverse density framework.

We tested the proposed 1-norm SVM approach over a set of COREL images. The performance of our framework is comparable with that of a state-of-the-art MIL approach, DD-SVM. Compared with DD-SVM, our framework is less sensitive to label uncertainties. In addition, the training time of our system is less than 0.2% of the training time for DD-

SVM. This makes the proposed framework a strong candidate for tasks that have stringent time limit, such as object recognition tasks that require on the fly training.

Although the experimental results are based on region features extracted from image segmentation, the proposed 1-norm SVM learning framework can be applied to regions generated by other region detectors. As continuation of this work, we intend to test the proposed method over features generated by different region detectors, such as those described in [11, 14, 15, 16, 20]. We will also explore the applications of the proposed 1-norm SVM learning framework to object recognition problems.

Acknowledgments

The research of Yixin Chen was supported in part by the Research Institute for Children at Children's hospital in New Orleans and NASA/LEQSF(2004)-DART-12. The research of James Z. Wang was supported in part by the National Science Foundation under grant No. IIS-0219272 and the PNC Foundation.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan, "Matching Words and Pictures," *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] K. P. Bennett, "Combining Support Vector and Mathematical Programming Methods for Classification," *B. Schölkopf, C. Burges and A. Smola, editors, Advances in Kernel Methods – Support Vector Machines*, pp. 307–326, 1999.

[3] J. Bi, K. P. Bennett, M. Embrechts, C. Breneman and M. Song, "Dimensionality Reduction via Sparse Support Vector Machines," *Journal of Machine Learning Research*, 3:1229–1243, 2003.

[4] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support Vector Machines for Histogram-Based Image Classification," *IEEE Transactions on Neural Networks*, 10(5):1055–1064, 1999.

[5] Y. Chen and J. Z. Wang, "Image Categorization by Learning and Reasoning with Regions," *Journal of Machine Learning Research*, 5:913–939, 2004.

[6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the Multiple Instance Problem with Axis-Parallel Rectangles," *Artificial Intelligence*, 89(1-2):31–71, 1997.

[7] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, 2:264–271, 2003.

[8] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements," *IEEE Int. Conf. on Computer Vision*, pp. 716–723, 2003.

[9] M. M. Gorkani and R. W. Picard, "Texture Orientation for Sorting Photos 'at a glance'," *Proc. 12th Int'l Conf. on Pattern Recognition*, I:459–464, 1994.

[10] ILOG, *ILOG CPLEX 6.5 Reference Manual*, ILOG CPLEX Division, Incline Village, NV, 1999.

[11] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," *Proc. of the 8th European Conference on Computer Vision*, pp. 404-416, 2004

[12] J. Li and J. Z. Wang, "Automatic Linguistic Indexing of Pictures By a Statistical Modeling Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, 2003.

[13] O. Maron and T. Lozano-Pérez, "A Framework for Multiple-Instance Learning," *Advances in Neural Information Processing Systems* 10, pp. 570–576, 1998.

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide baseline Stereo from Maximally Stable Extremal Regions," *Proc. British Machine Vision Conference*, 1:384-393, 2002.

[15] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," *International Journal of Computer Vision*, 60(1):63–86, 2004.

[16] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?' ", *Proc. of the 7th European Conference on Computer Vision*, 1:414–431, 2002.

[17] J. R. Smith and C.-S. Li, "Image Classification and Querying Using Composite Region Templates," *Int'l J. Computer Vision and Image Understanding*, 75:(1/2):165–174, 1999.

[18] A. J. Smola, B. Schölkopf, and G. Gätsch, "Linear Programs for Automatic Accuracy Control in Regression," *Proc. Int'l Conf. Artificial Neural Networks*, Berlin, Springer. 1999.

[19] M. Szummer and R. W. Picard, "Indoor-Outdoor Image Classification," *Proc. IEEE Int'l Workshop on Content-Based Access of Image and Video Databases*, pp. 42–51, 1998.

[20] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *International Journal on Computer Vision*, 59(1):61–85, 2004.

[21] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H.-J. Zhang, "Image Classification for Content-Based Indexing," *IEEE Transactions on Image Processing*, 10:(1):117–130, 2001.

[22] Q. Zhang and S. Goldman, "EM-DD: An Improved Multiple-Instance Learning Technique," *Advances in Neural Information Processing Systems* 14, 2002.

[23] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, "1-norm Support Vector Machines," *Advances in Neural Information Processing Systems*, 16, 2004.