Adaptive spatial partition learning for image classification

Bingyuan Liu, Jing Liu*, Hanqing Lu
National Laboratory of Pattern Recognition, CASIA, Beijing 100190, China

A R T I C L E   I N F O

Article history:
Received 25 November 2013
Received in revised form 21 February 2014
Accepted 24 March 2014
Communicated by Qingshan Liu
Available online 23 May 2014

Keywords:
Image classification
Spatial information
Group sparsity

A B S T R A C T

Spatial Pyramid Matching is a successful extension of bag-of-feature model to embed spatial information
of local features, in which the image is divided into a sequence of increasingly finer grids, and the grids
are taken as uniform spatial partitions in ad-hoc manner without any theoretical motivation. Obviously,
the uniform spatial partition cannot adapt to different spatial distribution across image categories.
To this end, we propose a data-driven approach to adaptively learn the discriminative spatial partitions
corresponding to each class, and explore them for image classification. First, a set of over-complete
spatial partitions covering kinds of spatial distribution of local features are created in a flexible manner,
and we concatenate the feature representations of each partitioned region. Then we adopt a discrimina-
tive learning formulation with the group sparse constraint to find a sparse mapping from the feature
representation to the label space. To further enhance the robustness of the model, we compress the
feature representation by removing the dimensions corresponding to those unimportant partitioned
regions, and explore the compressed representation to generate a multi-region matching kernel
prepared to train a one-versus-others SVM classifier. The experiments on three object datasets
(i.e. Caltech-101, Caltech-256, Pascal VOC 2007), and one scene dataset (i.e. 15-Scenes) demonstrate
the effectiveness of our proposed method.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, the bag-of-feature (BoF) model [5] becomes
very popular in image classification systems. The BoF model starts
from well-engineered local features, such as SIFT [18], HOG [6] and
color invariant descriptors [25], quantizes them into distinct visual
words, and then computes a histogram representation as the
image representation. The BoF-based representation describes an
image as an orderless collection of local features, while the spatial
layout of the features is completely neglected.

To overcome this problem, one popular extension, called as
Spatial Pyramid Matching (SPM) [15], has been shown effective for
image representation. It requires to first partition each image into
a fixed sequence of increasingly finer uniform grids (e.g. 1 × 1,
2 × 2, 4 × 4), and then concatenates the BoF features in each grids
forming a high dimensional image feature, while the grids in the
same level are equally treated for concatenation. However, images
in different classes often have different spatial distribution about
target and background context, contributing different discrimi-
native abilities to recognize a certain category. For instance, as
shown in Fig. 1, the airplanes tend to locate in the middle of the
image and the shown spatial partitioned regions (learned by the

* Corresponding author. Tel.: +86 10 82544507; fax: +86 10 82544594.
E-mail address: jliu@nlpr.ia.ac.cn (J. Liu).

http://dx.doi.org/10.1016/j.neucom.2014.03.057
0925-2312/© 2014 Elsevier B.V. All rights reserved.
remaining the dimensions corresponding to the most important regions and explore the compressed representation to estimate a multi-region matching kernel for a strong SVM classifier to each class. We evaluate our algorithm on three challenging object datasets (i.e. Caltech-101, Caltech-256, and Pascal VOC 2007), and one scene dataset (i.e. 15-Scenes), showing the effectiveness of our method in comparison with some related works.

The rest of the paper is organized as follows. Section 2 reviews the related work of BoF models with spatial information. In Section 3, we elaborate our proposed method of adaptive spatial partition learning for image classification. The experimental evaluation is given in Section 4, and we conclude in Section 5.

2. Related work

The BoF model, although is simple and directly borrowed from text retrieval community, has been proven useful and effective to represent an image for many computer vision tasks, such as object recognition [24], scene classification [2], image annotation and retrieval [17]. The standard BoF extracts a set of local patch descriptors, assigns each descriptor to the closest entry in a visual codebook which is learned offline by clustering a large sampling set of descriptors with K-means and then pool them to an image-level histogram signature. However, the ignorance of spatial information hinders its final performance. To overcome the limitation, many subsequent researches to incorporate spatial information have been done from the following two directions.

One direction is to incorporate the local spatial layout in image, i.e., the relative positions or pairwise positions of the local features. Savarese et al. [26] explore the combination of correlograms and visual words to represent spatially neighboring image regions. In [16], an efficient feature selection method based on boosting is proposed to mine high-order spatial features, while Morioka and Satoh [20] propose to jointly cluster feature space to build a compact local pairwise codebook capturing correlation between local descriptors and in [21] incorporate the spatial orders of local features. In [23], a data mining method is proposed to automatically find spatial configurations of local features occurring frequently on instances of a given object class.

Since images often have spatial preferences, another direction is to incorporate global spatial layout property, i.e., the absolute positions in image, which is also our focus in this paper. Lazebnik et al. [15] pioneer the direction of exploiting spatial layout property and propose SPM. In SPM, the image is divided into uniform grids at different scales (e.g. $1 \times 1, 2 \times 2, 4 \times 4$), and the features are concatenated over all cells. Yang et al. [31] and Wang et al. [30] show that incorporating sparse coding or locality-constrained coding into the SPM model improves the performance. More recently, the combinations of SPM with super vector [33] and fisher vector [22] models are demonstrated effective to obtain a good representation. Different from SPM, Krapac et al. [14] propose to encode the spatial layout by Gaussian mixture model using the Fisher kernel framework. In [3], local features of an image are first projected to different directions creating a series of ordered bag-of-features.

Several parameters in SPM model, such as the number of pyramid levels and the weights of the grid at each level, are chosen in an ad-hoc manner without any optimization [15]. Thus the model is not adaptable to different situations and the performance is highly dependent on the experiences and datasets. To address this issue, Harada et al. [11] propose to form the image feature as a weighted sum of semi-local features over all pyramid levels and the weights are automatically selected to maximize a discriminative power. To design better spatial partition, Sharma and Jurie [28] define a space of grids where each grid is obtained by a series of recursive axis aligned splits of cells and propose to learn the spatial partition in a maximum margin formulation. In this paper, we both consider the weights of different regions and the spatial partition style. The most related work to ours is [13], which adopts the idea of over-complete and formulate the problem in a multi-class fashion with sparse regularization for feature selection. However different categories actually have different spatial distribution, thus we adopt the idea of group sparsity to adaptively learn a class-specific spatial partition style and then explore a semantically compression and multi-region matching kernel to classify each category.

3. Adaptive spatial partition learning

In this section, we describe the details of our proposed framework for image categorization, which is shown in Fig. 2.

3.1. Over-complete spatial partition

As shown in Fig. 2, starting with an input image $I_n$, we densely extract local features (e.g. SIFT or HOG) and encode the features by learned codebook. Instead of spatial pyramid partition, we create an over-complete spatial partition set to compile the spatial information, and use the spatial pooling function to concatenate representations of all the regions.

Spatial pyramid partition plays an important role in many state-of-the-art image classification systems. They adopt an increasingly...
finer uniform grids (e.g. \(1 \times 1.2 \times 2, 4 \times 4\)). Obviously, the uniform spatial partition cannot adapt to different spatial distribution across image categories. In order to overcome this constraint, we propose to construct the spatial partitions in a more flexible scheme, involving as many geometric properties of the local features as possible. We think an ideal partition should be able to divide the image into some semantic regions, i.e. object region and background context region.

First, we employ randomly distributed horizontal and vertical grids to divide the image into rectangular grids (the dotted grids as shown in Fig. 2). These grids are considered as the candidate grids to generate a certain kind of spatial partition. Then a type of spatial partition is created by randomly choosing a subset of the candidate grids (the solid grids as shown in Fig. 2). By covering all the possible combination of the candidate grids, an over-complete partition is established. In this way, we are able to the possible combination of the candidate grids, an over-complete

[Fig. 2. Our image classification pipeline. See Section 3 for details.]

<image>

classifier leads to the following optimization problem:

\[
\min_{w,b} \frac{1}{N} \sum_{n=1}^{N} l(w^T x_n + b, y_n) + \lambda R(w)
\]

where vector \(w\) and scalar \(b\) are the parameters to be estimated, \(x_n\) is the feature vector of the \(n\)-th sample, \(y_n \in \{-1, 1\}\) is the label of the \(n\)-th sample, \(l(w^T x_n + b, y_n)\) is a certain non-negative convex loss, \(N\) is the number of training images, \(R(w)\) is a regularizer and \(\lambda \in R\) is the regularization coefficient. We choose the binomial negative log likelihood as the loss function:

\[
l(w^T x_n + b, y_n) = \ln(1 + \exp(-y_n(w^T x_n + b)))
\]

which leads to the logistic regression classifier.

In addition, we expect the classifier to select the most discriminative spatial partitions because of the high redundancy. Noted that only a few partitioned regions among all are discriminative and helpful for image classification, which inspires us to perform semantical compression with the group sparsity prior. Recent analysis and application of the mixed norm regularization [1,27,4] show that under certain conditions the sparse coefficient vector \(w\) enjoys the group sparsity property, encouraging the content-based structured feature selection in high-dimensional feature space. Thus we adopt the idea of structured sparsity [32], and train the binary linear classifier via the following optimization problem:

\[
\min_{w,b} \frac{1}{N} \sum_{n=1}^{N} \ln(1 + \exp(-y_n(w^T x_n + b))) + \lambda \|w\|_{2,1}
\]

where \(\|w\|_{2,1}\) denotes \(l_2/l_1\) norm regularizer, incorporating group sparsity property. We set the feature groups here as the dimensions corresponding to a certain partitioned region \(x_{R_i}\), encouraging the model to mine and select discriminative spatial partitions. The number of the groups is \(G\) and the dimension of features in each group is \(D\).

The formulation of the \(l_2/l_1\) mixed-norm regularizer is

\[
\|w\|_{2,1} = \sum_{i=1}^{G} \|w_{R_i}\|_2
\]

where \(w_{R_i}\) is the \(i\)-th group of parameters corresponding to the \(i\)-th spatial region \(R_i\). This motivates dimensions in the same group to be jointly zero. Thus the optimization procedure tends to select a much smaller but more discriminative subset within the over-complete representations. Beyond the regular \(l_1\) norm regularizer, the sparsity is now imposed on spatial region level rather than merely on feature level.

Although the optimization problem for \(l_2/l_1\) regularized logistic regression is convex, the non-smooth penalty function makes the
optimization highly nontrivial. We adopt the efficient algorithm proposed in [19,12]. The dual of the proximal problem associated with the norm can be reformulated as a quadratic min-cost flow problem, which is able to be efficiently computed in polynomial time. The algorithm is very suitable for our problem because of its efficiency and ability to scale up to millions of variables. Also a well implemented open toolbox called SPAMS (Sparse Modeling Software)

3.3. Feature compression

The linear classification trained by Eq. (3) can be directly utilized to classify a given image. However, this classifier is weak because of the high redundancy of our representation. Thus we propose to compress feature dimensions according to the \( w \) learned by Eq. (3), leading to a semantically more compact and discriminative representation.

The learned parameters \( w \) in Eq. (3) can be regarded as importance weights of dimensions corresponding to each spatial region. Based on the idea, we adopt a leave-one-out scheme to measure the importance of each partitioned region. The importance value of a particular partitioned region is measured as the increase of the training error with the dimensions corresponding to the region removed, while the training error of our model over all training data given \( w \) is defined as

\[
Error_0 = \frac{1}{N} \sum_{n=1}^{N} \ln(1 + \exp(-y_n(w^T x_n + b)))
\]

where \( w \) and \( b \) are the solutions of Eq. (3). The error of neglecting the dimensions corresponding to the \( j \)-th partitioned region, i.e., setting \( w_{R_j} = 0 \), is denoted as \( Error_j \). Then, the importance of the \( j \)-th region is calculated by measuring the training error increase after neglecting the region dimensions as

\[
S_j = \frac{Error_j - Error_0}{Error_0}
\]

The larger \( S_j \) indicates the neglected dimensions of the \( j \)-th region are more important and discriminative for the category task. We then perform the feature compression on \( x \) by only removing the dimensions corresponding to the regions with positive spatial importance \( S_j \) which is common for the particular class. By this way, the feature compression is performed semantically to remove redundant dimensions adapting to the particular class.

3.4. Multi-region matching kernel for SVM classification

In traditional SPM, the grids in the same level are equally treated to get the final kernel. However, the weights of different spatial regions should be different (usually the object region should be paid more attention than background context region). To this end, the values of \( S_j (j = 1, 2, ..., G) \) are employed to roughly indicate the importance of different regions and define a multi-region matching kernel with the weighted sum of the separate region kernels, prepared to train a SVM classifier:

\[
x(\hat{x}_1, \hat{x}_2) = \sum_{m=1}^{M} S_m \cdot K(\hat{x}_{R_m}, \hat{x}_{R_m})
\]

where \( \hat{x}_1 \) and \( \hat{x}_2 \) are feature vectors of two images after compression, \( \hat{x}_{R_m} \) and \( \hat{x}_{R_m} \) represent the corresponding dimensions of the \( m \)-th region, \( M \) is the remaining number of regions and \( S_m \) is the importance weight of the region. The kernel \( K \) may be any kernel function\(^2\) and a normalization operation is usually needed. With the multi-region matching kernel, we turn to train a one-versus-others classifier for each class to further enhance the classification performance.

4. Experiments and results

In the experiments, we implement and compare our method mainly with KSPM [15] (the popular kernel SPM), ScSPM [31] (the method that uses sparse coding and linear SVM) and some other works considering discriminative spatial information [28,13]. Although our method can be combined with diverse coding and pooling algorithms, we apply the sparse coding and max pooling strategy for fair comparison. We use a single local descriptor type, the popular SIFT descriptor, by densely extracting local patches of \( 16 \times 16 \) pixels computed over a grid with spacing of 8 pixels and then the local features are encoded with sparse coding. For all the experiments, we fix the codebook size as 1024. Considering simplicity and efficiency, we construct our over-complete spatial regions using three horizontal lines and three vertical lines. We also randomly generate some additional spatial grids to incorporate more flexible spatial information. Totally, a set of 100 different spatial partitioned regions is employed as our over-complete spatial partition set. The adaptive spatial partition learning model is then trained to obtain the parameters \( w \) for each class. Finally, we train the SVM classifier by the compressed features with the multi-region matching kernel. The trade-off parameters to the group sparsity regularization term and the SVM regularization term are chosen via 5-fold cross validation on the training data.

We demonstrate the effectiveness of our method on three diverse object databases, Caltech-101, Caltech-256, Pascal VOC 2007 and one scene database, 15-Scenes. Following the common benchmarking procedures, we repeat the experimental process by 5 times with different randomly selected training and testing images to obtain reliable results. The final results are reported by the mean and standard deviation of the classification rates. The extensive comparisons and analysis are presented in the following subsections.

4.1. Results on Caltech-101 dataset

We start our experiments with an in-depth analysis of our method on the dataset of Caltech-101 [8]. The Caltech-101 dataset contains 9144 images totally from 102 different categories, including 101 object categories and 1 additional background category, with high shape variability. The number of images per category varies from 31 to 800. We follow the common experiment setup for Caltech-101, training on 15 and 30 images per category and testing on the rest.

The performance comparison results are shown in Table 1. The ASPL (adaptive spatial partition learning) in the table denotes the method that directly use the spatial partition learning model trained by Eq. (3) as the final classifier and ASPL-SVM denotes our method described in Section 3.4 of training a stronger SVM classifier using the compressed feature with multi-region matching kernel. It is indicated that the best result is obtained by using the final compressed representations with multi-region matching kernel, which outperforms the traditional method, i.e. ScSPM [31], by a margin of roughly 4% according to our implementation. To evaluate the effect of our spatial partition learning model, we compare our method with the scheme of randomly selecting

\[^2\] In experiments, we use \( \chi^2 \) kernel for 15-Scenes dataset, and linear kernel for Caltech-101, Caltech-256 and Pascal VOC 2007 datasets.
the same number of spatial regions from the over-complete set instead of the spatial partition learning process, which is denoted as Rand + SVM in Table 1. It shows that our method outperforms the randomly selecting scheme, indicating that our method does help to select more important and discriminative spatial partitions for image classification. Our method also performs better than [13], in which they perform feature selection in a multi-class fashion to learn discriminative spatial partition for the whole categories. In our method we adaptively learn category-specific spatial partitions because different categories obviously have different spatial distributions, and the feature compression and multi-region matching procedure also improve the performance. In Fig. 3, we show some examples of the classes that our method increase most and some bad examples that our method decrease the performance. It is shown that our method obviously improve the classification accuracy under the condition that the objects have strong spatial prior nature.

In our method, there is one free parameter $\lambda$ in Eq. (3) needed to determine when we learn the discriminative spatial partitions, which controls the sparsity of the solution. The bigger $\lambda$ is, more sparse the solution will be, thus in the feature compression process described in Section 3.3 more spatial partitioned regions will be removed, leading to more compact final image representations. Fig. 4 shows the effect of the tradeoff coefficient $\lambda$ by investigating the classification accuracy with different values of $\lambda$. The accuracies are bad when the parameter is small or large, since small $\lambda$ may cause redundancy of the features, while large $\lambda$ may bring about loss of useful information. Thus the best performance was achieved with well balanced parameter. Empirically, we found that keeping the tradeoff term $\lambda$ to be around 0.5 yields good results. In the traditional spatial pyramid representation, the final image representation is in the dimension of 21,504 (the size of the codebook is 1024 and the spatial partition setup is 1 $\times$ 1, 2 $\times$ 2, 4 $\times$ 4). In our implementation when $\lambda$ is 1.5 the dimension of the final feature is 16,384, which is much less than spatial pyramid representation and performs better in the recognition task. This indicates that a more compact and discriminative representation can be obtained by our model, as we better explore the sparse nature of the features. While the best recognition performance is achieved with $\lambda$ set as 0.5, the final dimension of the image representation is 27,648.

Fig. 5 is an encouraging demonstration to show what we learned by the spatial partition learning method. We also draw the importance maps of the spatial region using the accumulated importance value of the partitioned regions obtained by our method described in Section 3.3, as shown in Fig. 5(c). It shows that the learned spatial partition properly capture the structural information of the classes. For example, the airplanes in the Caltech-101 dataset tend to locate in the middle of a image and our learned spatial regions shown in Fig. 5 capture this property by paying more attention to the middle regions. In the dolphin category, the objects appear mostly in the middle and top of the image, thus these regions are more significant to recognize the dolphin, which is well modeled by our method. It is demonstrated that the spatial importance we get really provide better spatial information than traditional method. However, for some classes, our method has not capture the spatial distribution of the target well because of the large diversity of the object and the uncon-straint of the location. Overall, our method provides a more discriminative class adaptive spatial information than traditional SPM and some other related works.

### 4.2. Results on Caltech-256 dataset

The Caltech-256 [10] dataset is an extension of the Caltech-101. It holds 29,780 images in 256 object categories where the number of images in each category varies from 31 to 800. This dataset is much more challenging as it possesses much higher intra-class variability and higher object location variability compared with

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>15 training</th>
<th>30 training</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSPM [15]</td>
<td>56.40</td>
<td>64.6 ± 0.80</td>
</tr>
<tr>
<td>ScSPM [31]</td>
<td>67.0 ± 0.45</td>
<td>73.2 ± 0.54</td>
</tr>
<tr>
<td>LCC + SPM [30]</td>
<td>65.43</td>
<td>73.4</td>
</tr>
<tr>
<td>Jia et al. [13]</td>
<td>—</td>
<td>75.3 ± 0.70</td>
</tr>
<tr>
<td>ScSPM</td>
<td>66.38 ± 0.30</td>
<td>72.32 ± 0.35</td>
</tr>
<tr>
<td>ASPL</td>
<td>65.04 ± 0.21</td>
<td>70.52 ± 0.20</td>
</tr>
<tr>
<td>Rand + SVM</td>
<td>65.93 ± 0.55</td>
<td>71.39 ± 0.84</td>
</tr>
<tr>
<td>ASPL + SVM</td>
<td>69.51 ± 0.30</td>
<td>76.72 ± 0.38</td>
</tr>
</tbody>
</table>

Fig. 4. The classification performance comparisons with varying tradeoff coefficient $\lambda$.
Caltech-101. Following the common experiment setup for Caltech-256, we tried our algorithm on 15, 30, 45, and 60 training images per class respectively and tested on the rest.

The performance comparison results are shown in Table 2. In the more challenging dataset than Caltech-101, our ASPL+SVM also consistently leads the performance and outperforms the baseline ScSPM by more than 4 percent for all the cases. In Fig. 6, we show some examples of the classes that our method increases most and some bad examples that our method decreases the performance.

**Fig. 5.** Demonstrations showing the effectiveness of our method to learn discriminative spatial partition of the Caltech-101 dataset. (a) Image examples for the classes. (b) The top 3 important regions according to the weights learned in our method. (c) Spatial importance maps obtained by our method for the particular class. The lighter, the more important the region is.
4.3. Results on Pascal VOC 2007 dataset

The PASCAL Visual Object Challenge (VOC) datasets are widely used as testbeds for evaluating algorithms for image understanding tasks and provide a common evaluation platform for both object classification and detection. This database is considered to be an extremely challenging one because all the images are daily photos obtained from Flicker where the size, viewing angle, illumination, appearances of objects and their poses vary significantly, with frequent occlusions. The PASCAL VOC 2007 dataset [7] consists of 9963 images from 20 classes, which are divided into “train”, “val” and “test” subsets, i.e. 25% for training, 25% for validation and 50% for testing. The classification performance is evaluated using the Average Precision (AP) measure, a standard metric used by PASCAL challenge. It computes the area under the Precision/Recall curve, and the higher the score, the better the performance. We use the train and val sets for training our model and report the mean average precision for the 20 classes on the test set as the performance measure, following the standard protocol of this database. Very high performance has been reported by using multiple features and combining diverse models [29]. For efficiency, here we just adopt one single feature, SIFT, just to evaluate our method.

In Fig. 7, we show our scores for all 20 classes in comparison with our baselines, ScSPM and ASPL. The performance of our method for most of the classes and on an average (43.02 vs. 41.17) is higher than the traditional SPM method. Fig. 8 shows the improvement of our method for each category. It is shown that for some classes, such as sheep, aeroplane and car, the performance increase is larger due to the well captured spatial nature of the objects. However, for some classes, such as dining table and pottedplant, the improvement of our method is limited. This is mainly on account of the highly diversity of the images in the database.

4.4. Results on 15-Scenes dataset

We finally experiment with a popular scene classification benchmark, 15-Scenes dataset. This dataset includes 15 classes of different scenes (e.g. kitchen, coast, highway), containing totally 4485 gray-scale images with the number of each category ranging from 200 to 400. Fig. 9 shows some example images of the 15-Scenes dataset. Following the common experiment setup of the dataset [15,31], we take 100 images per class for training and the rest for testing. The other parameters are transposed from the former experiments. Table 3 shows the detailed comparison results. It is shown that the ASPL + SVM method achieves better performance than traditional KSPM and ScSPM. The evaluation on the 15-Scenes dataset demonstrates that our method also adapt to the scene categories to improve the classification accuracy beyond the object recognition. Fig. 10 shows the confusion table between the 15 scene categories. Confusion occurs
Fig. 9. Example images for the 15-Scenes dataset.

Table 3
Classification rate (%) comparison on 15-Scenes.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Classification rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSPM [15]</td>
<td>81.40 ± 0.50</td>
</tr>
<tr>
<td>ScSPM [21]</td>
<td>80.29 ± 0.93</td>
</tr>
<tr>
<td>Sharma and Jurie [28]</td>
<td>80.10 ± 0.60</td>
</tr>
<tr>
<td>ScSPM</td>
<td>79.29 ± 0.53</td>
</tr>
<tr>
<td>ASPL</td>
<td>77.85 ± 0.54</td>
</tr>
<tr>
<td>Rand + SVM</td>
<td>77.90 ± 0.82</td>
</tr>
<tr>
<td>ASPL + SVM</td>
<td>82.19 ± 0.10</td>
</tr>
</tbody>
</table>

Fig. 10. Confusion table for the 15-Scenes dataset. Average classification rates for individual classes are listed along the diagonal. The entry in the $i$-th row and $j$-th column is the percentage of images from class $i$ that are misidentified as class $j$. 

<table>
<thead>
<tr>
<th>Class</th>
<th>Suburb</th>
<th>Coast</th>
<th>Forest</th>
<th>Highway</th>
<th>Insidecity</th>
<th>Mountain</th>
<th>OpenCountry</th>
<th>Street</th>
<th>TallBuilding</th>
<th>Office</th>
<th>Bedroom</th>
<th>Industrial</th>
<th>Kitchen</th>
<th>Living Room</th>
<th>Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>suburb</td>
<td>0.90</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>coast</td>
<td>0.00</td>
<td>0.97</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>forest</td>
<td>0.00</td>
<td>0.00</td>
<td>0.98</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>highway</td>
<td>0.00</td>
<td>0.03</td>
<td>0.00</td>
<td>0.89</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>insidecity</td>
<td>0.01</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
<td>0.79</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>mountain</td>
<td>0.00</td>
<td>0.04</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.89</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>opencountry</td>
<td>0.01</td>
<td>0.11</td>
<td>0.05</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.72</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>street</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
<td>0.00</td>
<td>0.92</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>tallbuilding</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>office</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.92</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>bedroom</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>industrial</td>
<td>0.01</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.05</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.11</td>
<td>0.00</td>
<td>0.03</td>
</tr>
<tr>
<td>kitchen</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>living room</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>store</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td>0.00</td>
<td>0.04</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
between the indoor classes (e.g., bedroom, kitchen, and living room) and also between some natural classes (e.g., coast and open country).

5. Conclusion

In this paper, we address the issue of the discriminative spatial partition learning and propose a data-driven approach to discover the most discriminative partitioned regions to the particular class. Different from traditional SPM applying manually defined spatial pyramid partitions, the proposed approach constructs a more flexible spatial partition set and adopt the idea of structured sparsity to learn discriminative spatial partitions. Then we propose to measure the importance of the spatial partitioned regions by the learned sparse parameters and train SVM classifier with a multi-region matching kernel. Our method outperforms traditional SPM method on three object datasets (i.e. Caltech-101, Caltech-256, Pascal VOC 2007), and one scene dataset (i.e. 15-Scenes), and the experiments have shown its effect to adaptively capture the spatial information of the images belonging to different categories.

Acknowledgements

This work was supported by 973 Program (2010CB327905) and National Natural Science Foundation of China (61272329, 61273034, 61202325).

References