Feature and Region Selection for Visual Learning

Ji Zhao, Liantao Wang
Ricardo Cabral, Fernando De la Torre

CMU-RI-TR-12-14

June 2014

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Visual learning problems such as object classification and action recognition are typically approached using extensions of the popular bag-of-words (BoW) model. Despite its great success, it is unclear what visual features the BoW model is learning: Which regions in the image or video are used to discriminate among classes? Which are the most discriminative visual words? Answering these questions is fundamental for understanding existing BoW models and inspiring better models for visual recognition.

To answer these questions, this paper presents a method for feature selection and region selection in the visual BoW model. This allows for an intermediate visualization of the features and regions that are important for visual learning. The main idea is to assign latent weights to the features or regions, and jointly optimize these latent variables with the parameters of a classifier (e.g., SVM). There are four main benefits of our approach: (1) Our approach accommodates non-linear additive kernels such as the popular χ^2 and intersection kernel; (2) our approach is able to handle both regions in images and spatio-temporal regions in videos in a unified way; (3) the feature selection problem is convex, and both problems can be solved using a scalable reduced gradient method; (4) we point out strong connections with multiple kernel learning and multiple instance learning approaches. Experimental results in the PASCAL VOC 2007, MSR Action Dataset II and YouTube illustrate the benefits of our approach.

Contents

1 Introduction

The last decade has witnessed great advances in machine learning and computer vision that have largely improved the performance and reduced the computational complexity of visual learning algorithms. Although there has been much progress in supervised visual learning, two main limitations still exist: (1) the reliance on human labeling limits the application of supervised methods in problems involving many categories; (2) these discriminative models lack interpretability because they do not produce mid-level representations (e.g., what are most important visual features for discrimination?).

For instance, consider Fig. 1, where there are a set of images that contain a car (Fig. 1 (a)) and a set of images that do not contain a car (Fig. 1 (b)). Given these sets, the goal of a weakly-trained classifier is to discover discriminative regions and use them to train a car detector. Most of the successful approaches for Weakly-Supervised Learning (WSL) [19, 11, 24, 8, 29, 17] rely on bag-of-words (BoW). BoW type of methods build a vocabulary of visual words to encode the visual representation and then use it to learn a binary classifier (e.g., kernel SVM). Although these techniques achieve state-of-the-art performance, the feature spaces induced by kernels obfuscate understanding ow which are the visual features that are most important for discrimination in the image space. The aim of this paper is to develop algorithms that learn in a weakly-supervised manner which are the discriminative features and regions. We aim to answer the following questions: Which visual words are used to discriminate cars versus non-cars (Fig. 1(c)) ? Which are the discriminative regions in the image (e.g., car in Fig. 1(d))? In addition to still images, we also apply our method to find discriminative spatio-temporal regions for activity recognition from video (Fig. 1 (e)-(h)).

WSL algorithms can partially solve the problem of localization of discriminative features, avoiding the time-consuming and error-prone manual localization process. Moreover, the selected regions are more informative to train detectors [19]. Due to its importance, WSL has been a popular topic researched in the last few years. Existing algorithms for WSL rely on multiple instance learning (MIL) and have mostly been applied to linear classifiers. A major challenge is how to extend these methods to cope with kernel representations while allowing for region and feature selection, which is a non-trivial task. For instance, in the case of SVM, an obvious solution would be to kernealize MIL using kernel approximations (e.g., [25]) and apply MIL to a linear SVM. However, this is difficult to implement and generally inefficient (see Section 2).

This paper proposes a feature and a region selection method for visual learning in the kernel space. The feature selection method is general for the family of additive kernels, and the region selection is valid for all kernels. The contributions of our work include: (1) a convex model for feature selection in the kernel space; (2) a method for region selection using non-linear kernels; (3) discovery and visualization of the most discriminative visual words, regions in images and spatio-temporal volumes in videos; (4) connections of our work with existing approaches including multiple kernel learning (MKL) and MIL. Experimental results in the PittCar dataset, PASCAL VOC 2007, MSR Action Dataset II and YouTube dataset illustrate the benefits of our approach.

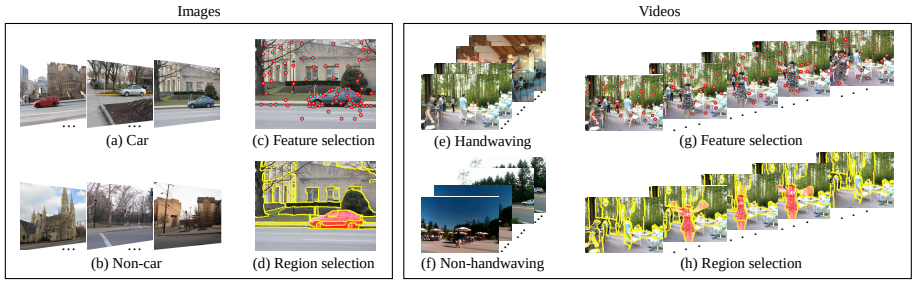


Figure 1: Given a set of images containing a car (a) and images without a car (b), this paper proposes an algorithm to select the visual features (c) and regions (d) that are most discriminative in the kernel space. Similarly, given a set of videos containing hand-waving actions (e) and actions that are not hand-waving (f), we find the most discriminative spatio-temporal features (g) and spatio-temporal regions (h).

2 Previous Work

Feature Selection in Kernel Space Selecting relevant features in kernel spaces has been a challenging problem addressed by several researchers. Cao et al. [5] developed a feature selection method by learning feature weights in the kernel space. This procedure is done as a data processing step, independently of the classifier construction. There also exist methods that perform feature selection and classifier construction jointly by inducing sparsity, such as [10, 1, 4, 13]. We will build on previous work by Nguyen et al. [18] who proposed a convex feature weighting method for linear SVM. Our work, however, extends [18] by adding non-linear additive kernels that are common in computer vision. Note that a trivial solution using kernel approximations (e.g., [25]) will not work for our purposes. For instance, using the kernel expansion for χ^2, each bin in the histogram will transform to several dimensions in the kernel approximation. A linear SVM will weigh each of the components differently; however, the components coming from the same bin should have the same weight. This constraint can be imposed, but it is unclear that the convexity property of [18] holds. Moreover, we also address a different problem, and propose an extension for feature and region selection that is scalable to large amounts of visual data. In addition, we provide connections of this work to MIL and MKL.

Multiple Instance Learning (MIL) In the MIL setting, each image is modeled as a bag of regions, and each region is an instance. With two classes, the negative bag only contains negative instances and the positive at least one positive. The goal of MIL is to label the positive instances in the positive bags. Many MIL algorithms have been successfully used for weakly supervised learning, such as MILboost [9], MI-SVM [2, 19, 8, 29] and SparseMIL [26]. MIL has been applied to object detection for images [8, 19], time series [19] and videos [11, 24, 23]. Among these methods, MI-SVM is arguably the most popular. However, current methods based on MI-SVM have two main limitations: (1) most approaches use bounding boxes for localization (e.g., [19, 22]) instead of arbitrary shapes, and (2) to the best of our knowledge are limited

to linear kernels.

MIL aims to jointly select positive instances while training a classifier, which leads to a NP-hard combinatorial problem that is typically solved using heuristics that heavily depend on the initialization. Our formulation, on the other hand, uses the assumption of a data-driven weighting of instances, and can be formulated as a convex problem. Moreover, this formulation shows the relative importance of part/instances, which are essential for visualization purposes. Our work is most related to Liu and Wang [15], who proposed a region of support to visualize what the BoW model has learned. However, their method uses a linear SVM and it is unclear how to extend it to the kernel domain or be useful for feature selection.

3 Feature Selection for Additive Kernels

This section proposes a convex feature selection method for additive kernels. Let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ (see footnote[1] for an explanation of the notation used in this work) be a training set of n samples, where $\mathbf{x}_i \in \mathbb{R}^D$ is the histogram of BoW for the i^{th} image, D is the number of visual words in the codebook, and $y_i \in \{-1, +1\}$ are the corresponding labels.

Popular choices of kernels for visual learning are additive, such as the χ^2 and the histogram intersection kernels [7]. Formally, a kernel $K(\cdot,\cdot)$ on $\mathbb{R}^D \times \mathbb{R}^D$ is *additive* if it satisfies $K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^D \kappa(x_{ik}, x_{jk})$ for any samples $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$, where x_{ik} is the k^{th} bin of the BoW histogram for the i^{th} image. That is, the kernel function $\kappa(x_{ik}, x_{jk})$ is defined on one bin of the histogram.

Given an additive kernel, our goal is to weigh the features with a weight vector in the kernel space. We parameterize the feature space with a weight vector $\mathbf{p}$. That is, we construct a mapping $\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) = [\sqrt{p_1}\boldsymbol{\psi}^\top(x_{i1}), \cdots, \sqrt{p_D}\boldsymbol{\psi}^\top(x_{iD})]^\top$, that assigns different weights to different feature map bins, where $\boldsymbol{\psi}(x_{ik})$ is the feature map for the k^{th} bin of the i^{th} histogram, $\mathbf{p} = [p_1, \cdots, p_D]^\top$ are the feature weights, and $p_k \geq 0\ \forall k$. In the maximum margin framework, we would like to find the separating hyperplane of a SVM and the feature weighting vector $\mathbf{p}$ that has the largest margin between classes. However, different values of $\mathbf{p}$ correspond to different feature spaces, and since the margins in two different feature spaces cannot be directly compared, it is necessary to normalize the margin. Following [18], we consider the normalized margin, and the SVM becomes:

$$\begin{aligned} \min_{\mathbf{w},b,\mathbf{p},\boldsymbol{\xi}} \quad & \frac{1}{2}\varphi(\mathbf{p})\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 - \xi_i,\ \forall i; \\ & \xi_i > 0,\ \forall i. \end{aligned} \tag{1}$$

where $\varphi(\mathbf{p}) = \sum_{i,j=1}^n \frac{1+y_i y_j}{2}\|\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) - \boldsymbol{\phi}(\mathbf{x}_j, \mathbf{p})\|^2$ is the normalization factor, $\{\xi_i\}_{i=1}^n$ are positive slack variables, and C is the parameter that controls the trade-off between generalization and training error.

[1]Bold lowercase letters, such as $\mathbf{p}$, denote column vectors. p_i represents the i^{th} entry of the column vector $\mathbf{p}$. Non-bold letters represent scalar variables. Calligraphic uppercase letters denote sets (e.g., $\mathcal{S}, \mathcal{B}$).

In order to transform Eq. (1) into a convex optimization problem, we make use of two properties of additive kernels. First, the normalization factor can be re-written as: $\varphi(\mathbf{p}) = \sum_{k=1}^{D} p_k a_k$, where $a_k = \sum_{i,j=1}^{n} \frac{1+y_i y_j}{2} \|\boldsymbol{\psi}(x_{ik}) - \boldsymbol{\psi}(x_{jk})\|^2$. Note that a_k can be interpreted as the average distance of the k^{th} bin in kernel space, and it can be computed from the training data a priori. Second, the hyperplane $\mathbf{w}$ can be re-written as a vertical concatenation of D column vectors as $\mathbf{w} = [\mathbf{w}_1^\top, \cdots, \mathbf{w}_D^\top]^\top$, where each $\mathbf{w}_k$ weighs the feature map for each bin $\boldsymbol{\psi}(x_{ik})$. Then the following two equations hold: $\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) = \sum_{k=1}^{D} \sqrt{p_k} \mathbf{w}_k^\top \boldsymbol{\psi}(x_{ik})$, and $\|\mathbf{w}\|^2 = \sum_{k=1}^{D} \|\mathbf{w}_k\|^2$.

Since $\varphi(\mathbf{p})$ is homogeneous in $\mathbf{p}$, we can always scale $\mathbf{p}$ appropriately to get $\varphi(\mathbf{p}) = 1$. Using this constraint and the previous re-parameterizations, and making a variable substitution $\mathbf{w}_k \leftarrow \sqrt{p_k}\mathbf{w}_k$, Eq. (1) can be written as

$$\begin{aligned} \min_{\mathbf{w},b,\mathbf{p},\boldsymbol{\xi}} \quad & \frac{1}{2} \sum_{k=1}^{D} \frac{\|\mathbf{w}_k\|^2}{p_k} + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i \left[\sum_{k=1}^{D} \mathbf{w}_k^\top \boldsymbol{\psi}(x_{ik}) + b \right] \geq 1 - \xi_i, \ \forall i; \\ & \sum_{k=1}^{D} a_k p_k = 1; \ \ \mathbf{p} \geq 0; \ \ \boldsymbol{\xi} \geq 0. \end{aligned} \tag{2}$$

Eq. (2) is convex and unlike the work presented in [18] allows for additive kernels. Moreover, while [18] used CVX for optimizing Eq.(1), we use a more scalable optimization strategy, see Section 3.1.

Relation to Multiple Kernel Learning (MKL) We note the remarkable relationship between our feature selection formulation in Eq. (2) and MKL [21], with the main difference being the constraints on $\mathbf{p}$. In MKL, the constraint is that $\mathbf{p}$ lies on a unit L_1-ball, i.e., $\sum_{k=1}^{D} p_k = 1$. The L_1 ball induces negative elements, so we rewrite this constraint to the probability simplex. That is, in our feature selection formulation, the constraint is data-driven and adaptive, i.e., $\sum_{k=1}^{D} a_k p_k = 1$ and $p_k > 0$. Note that weighing each bin differently results in increased accuracy because we can model bins with different variances.

3.1 Optimization with the Reduced Gradient Method (RGM)

The connection between our feature selection method and MKL allows us to exploit the existing algorithms for MKL. For fixed $\mathbf{w}, b, \boldsymbol{\xi}$, Eq. (2) can be reformulated as a non-linear objective function with constraints over the simplex on $\mathbf{p}$. We can derive a scalable algorithm with proven convergence properties by optimizing Eq. (2) with a reduced gradient method [21]. Eq. (2) can be reformulated as

$$\min_{\mathbf{p}} J(\mathbf{p}) \text{ such that } \sum_{k=1}^{D} a_k p_k = 1, \ p_k \geq 0, \tag{3}$$

where

$$J(\mathbf{p}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\sum_{k=1}^{D}\frac{\|\mathbf{w}_k\|^2}{p_k} + C\sum_{i=1}^{n}\xi_i \tag{4}$$
$$\text{s.t.} \quad y_i\left[\sum_{k=1}^{D}\mathbf{w}_k^\top\boldsymbol{\psi}(x_{ik}) + b\right] \geq 1-\xi_i,\ \forall i$$

where $\boldsymbol{\xi} \geq 0$. By setting the derivatives of the Lagrangian in Eq. (4) with respect to the primal variables to zero, we get the associated dual problem

$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \sum_{k=1}^{D} p_k\kappa(x_{ik},x_{jk}) + \sum_{i=1}^{n}\alpha_i \tag{5}$$
$$\text{s.t.} \ \sum_{i=1}^{n}\alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq C,\ \forall i.$$

This dual problem is identified as the standard SVM dual problem using the combined kernel $K(\mathbf{x}_i,\mathbf{x}_j) = \sum_{k=1}^{D} p_k\kappa(x_{ik},x_{jk})$. Because of strong duality, $J(\mathbf{p})$ is also the objective value of this dual problem. By differentiation of the dual function w.r.t. p_k,

$$\frac{\partial J}{\partial p_k} = -\frac{1}{2}\alpha_i^*\alpha_j^* y_i y_j \sum_{k=1}^{D} p_k\kappa(x_{ik},x_{jk}),\ \forall k. \tag{6}$$

The optimization problem in (4) is a non-linear objective function with constraints over the simplex. We use the reduced gradient method to solve this problem. Once the gradient of $J(\mathbf{p})$ is computed, $\mathbf{p}$ is updated using a descent direction ensuring that the equality constraint and the non-negativity constraints on $\mathbf{p}$ are satisfied. Let p_μ be a non-zero entry of $\mathbf{p}$. The reduced gradient of $J(\mathbf{p})$, denoted $\nabla_{red}J$, can be written as

$$[\nabla_{red}J]_k = \frac{\partial J}{\partial p_k} - \frac{a_k}{a_\mu}\frac{\partial J}{\partial p_\mu} = \sum_{k\neq\mu}\left(\frac{a_k^2}{a_\mu^2}\frac{\partial J}{\partial p_\mu} - \frac{a_k}{a_\mu}\frac{\partial J}{\partial p_k}\right)\ \forall k \neq \mu. \tag{7}$$

The positivity constraints also have to be taken into account in the descent direction. Therefore, the descent direction for updating $\mathbf{p}$ is

$$\mathbf{r}_k = \begin{cases} 0; & \text{if } p_k = 0 \text{ and } \dfrac{\partial J}{\partial p_k} - \dfrac{a_k}{a_\mu}\dfrac{\partial J}{\partial p_\mu} > 0 \\ -\dfrac{\partial J}{\partial p_k} + \dfrac{a_k}{a_\mu}\dfrac{\partial J}{\partial p_\mu}; & \text{if } p_k > 0 \text{ and } k \neq \mu \\ \displaystyle\sum_{v\neq\mu, p_v>0}\left(-\frac{a_v^2}{a_\mu^2}\frac{\partial J}{\partial p_\mu} + \frac{a_v}{a_\mu}\frac{\partial J}{\partial p_v}\right) & \text{if } k = \mu. \end{cases} \tag{8}$$

The usual updating scheme is $\mathbf{p} \leftarrow \mathbf{p} + \gamma\mathbf{r}$, where γ is the step size. γ is calculated using a line search method.

4 Region Selection

In the previous section, we have proposed a feature selection method in the kernel space for additive kernels. However, visual features are typically very sparse and it is difficult to assess which regions the classifier uses for learning. In this section, we propose a method for selecting discriminative regions in images and videos. Prior to applying our method, we over-segment the images and videos into regions, i.e. superpixels [3] or spatio-temporal regions [6]. Once the regions are segmented, we encode each region using the BoW codebook learned from all training images/videos. Similar to Section 3, we assume an additive property of the classifier for region selection so that the classifier score of an image is a weighted sum of the score for each of regions. Note, however, that the homogeneity assumption is no longer needed for region selection, allowing our method to be applied to any kernel

4.1 Localization as Region Selection Problem

Given an over-segmentation for each image (or a video) into m_i regions, $\mathbf{h}_{ik}$ and s_{ik} represent the BoW histogram and the importance (weight) for the k^{th} region in the i^{th} image. Our SVM for region selection minimizes

$$\begin{aligned}\min_{\mathbf{w},b,\{\mathbf{s}_i\},\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i\in\mathcal{B}^+} \xi_i^+ + C_2 \sum_{i\in\mathcal{B}^-}\sum_{k=1}^{m_i} \xi_{ik}^- \\ \text{s.t.} \quad & \sum_{k=1}^{m_i} s_{ik}\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{h}_{ik}) + b \geq 1 - \xi_i^+,\ \forall i \in \mathcal{B}^+; \\ & -\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{h}_{ik}) - b \geq 1 - \xi_{ik}^-,\ \forall i \in \mathcal{B}^-, k \in \{1,\cdots,m_i\}; \\ & \|\mathbf{s}_i\|_1 = 1, \mathbf{s}_i \geq 0,\ \forall i \in \mathcal{B}^+;\ \ \boldsymbol{\xi} \geq 0,\end{aligned} \tag{9}$$

where $\boldsymbol{\phi}(\cdot)$ is the kernel feature map, and $\mathcal{B}^+$ and $\mathcal{B}^-$ are index sets of training samples with label $+1$ and -1, respectively. Since $\mathbf{s}_i$ lies on the probability simplex, the solution tends to be sparse and can be used for region selection. C_1 and C_2 trade-off the model complexity and empirical losses on the positive and negative bags, respectively. The first constraint is imposed on the positive bags, and enforces that, for positive images, a combination of its segments' scores is expected to be positive or it will be penalized. The second constraint enforces that all the segments' scores of the negative images should be negative.

Testing Once the SVM parameters are learned, the classification and localization for new test images can be performed simultaneously. Given the i^{th} image and its over-segmented regions (indexed by k), we can provide an initial estimate if a region belongs to a discriminative region or not by computing the decision value $\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{h}_{ik}) + b$. The final score of the image is the weighed average score of its regions, that is, $\sum_k s_{ik}\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{h}_{ik}) + b$. The weights s_{ik} are learned during training.

Relation to MIL The proposed region selection has connections to MIL. MIL makes the assumption that a negative bag has all negative instances, and a positive bag contains at least one positive instance. However, in our region selection method, the bag label is determined by a combination of regions. This is a more reasonable assumption for visual learning because it is difficult to say which region triggers a label for an image considering that the segmentation may not yield perfect results. *Generally speaking, in MIL, the label is determined by the maximum of the instances scores, while in our method, the label is determined by the weighted mean of all the instances' scores*.

Note that our formulation is different from previous key-instance SVM (KI-SVM), where it is assumed that there is only one positive instance in each positive bag [14]. Our formulation is also different from kernel latent SVM (KLSVM) [29], which also relies on a single instance to determine the label for positive bags. In [28], the model scores an image using the combination of regions, but it is limited to the linear kernel case.

4.2 Optimization with the Reduced Gradient Method

Eq. (9) is a non-linear objective function with constraints over the simplex. We used the reduced gradient method (RGM) to solve it with a coordinate descent strategy. First, we fix the weights $\mathbf{s}$, and optimize the object function w.r.t. $\mathbf{w}$, b and $\boldsymbol{\xi}$. Second, we use the RGM to update $\mathbf{s}$.

In order to simplify the notation, we take each region in a negative image as a negative bag that contains only one instance. We set C_2 equal to C_1, and reformulate the problem as:

$$\min_{\{\mathbf{s}_i\}} J(\{\mathbf{s}_i\}) \text{ such that } \|\mathbf{s}_i\|_1 = 1,\ \mathbf{s}_i \geq 0,\ \forall i, \tag{10}$$

where

$$\begin{aligned} J(\{\mathbf{s}_i\}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i\left[\mathbf{w}^\top \sum_{k=1}^{m_i} s_{ik}\boldsymbol{\phi}(\mathbf{h}_{ik}) + b\right] \geq 1-\xi_i,\ \forall i \end{aligned} \tag{11}$$

where $\boldsymbol{\xi} \geq 0$. By setting the derivatives of the Lagrangian (11) to zero, we get the associated dual problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \left(\sum_{k=1}^{m_i}\sum_{l=1}^{m_j} s_{ik}s_{jl}K(\mathbf{h}_{ik},\mathbf{h}_{jl})\right) + \sum_{i=1}^{n}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0;\ \ 0 \leq \alpha_i \leq C;\ \forall i. \end{aligned} \tag{12}$$

This is the standard dual formulation for SVM with the combined kernel $K(\mathbf{h}_i,\mathbf{h}_j) = \sum_{k=1}^{m_i}\sum_{l=1}^{m_j} s_{ik}s_{jl}K(\mathbf{h}_{ik},\mathbf{h}_{jl})$. Because of strong duality, $J(\{\mathbf{s}_i\})$ is also the objective value of this dual problem. By differentiation of the dual function with respect to

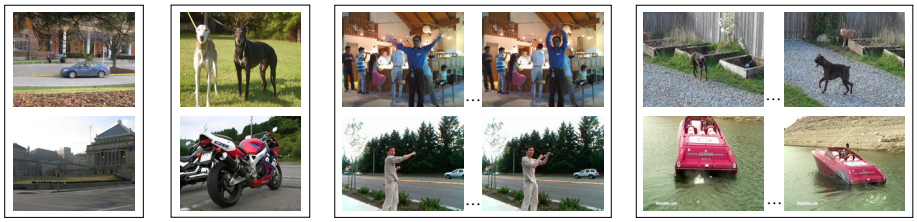


Figure 2: Some examples of the datasets. (a) PittCar; (b) PASCAL VOC; (c) MSR Action II; (d) YouTube Objects.

s_{ik}, we have

$$\frac{\partial J}{\partial s_{ik}} = -\frac{1}{2}\sum_{j=1}^{n}\alpha_i^*\alpha_j^* y_i y_j \sum_{l=1}^{m_j} s_{jl} K(\mathbf{h}_{ik}, \mathbf{h}_{jl}). \tag{13}$$

At first glance, computing the gradient in Eq. (13) seems to be computationally expensive. However, this calculation is efficient for the following reasons. First, we can reformulate it as a compact matrix formulation when calculating $\frac{\partial J}{\partial \mathbf{s}_i}$. Second, since α is sparse, the complexity of calculating gradient is largely reduced.

5 Experimental Results

This section validates the performance of our feature selection and region selection algorithm by comparing them with other state-of-the-art approaches on the following four datasets:

PittCar Dataset [19] contains 400 images of which 200 are positive and 200 negative, see Fig. 2a. There is only one object in each positive image. Half of the positive and negative images were used as training data, and the rest were used for testing. For each image, we extracted SIFT features [16] densely and selected 10000 of them randomly. All the SIFT descriptors were quantized into 1000 visual words, obtained by applying K-means to 100000 training samples.

PASCAL VOC 2007 consists of 9963 images. For examples see Fig. 2b. There are 20 object categories, with some images containing multiple objects. This dataset has been previously split into training and testing sets, which contain 5011 and 4952 images respectively. We proceeded as in the PittCar Dataset, extracting SIFT features and building a codebook of 1000 dimensions. We only used two of the classes since our main purpose is to validate our model as visualization tool.

MSR Action Dataset II [30] comprises 54 video sequences of crowded environments, see Fig. 2c. There are 3 action categories: hand waving, handclapping, and boxing. Each video sequence contains multiple actions. Following [23], we split each video to contain only one action and randomly selected 135 videos as training data and 46 for test data. During this random division, the videos containing multiple actions that

Table 1: The comparison of classification performance for feature selection methods and MKL in the PittCar, PASCAL VOC 2007 and MSR Action II datasets.

		linear SVM	χ^2 SVM	MKL-χ^2 [21]	FS-linear [18]	FS-χ^2 (ours)
Car	AP	0.833	0.959	0.961	0.967	**0.988**
	# Features	1000	1000	120	112	56
PASCAL	**cat**: AP	0.290	0.375	0.381	0.315	**0.384**
	# Features	1000	1000	472	665	284
	dog: AP	0.278	0.337	0.342	0.306	**0.347**
	# Features	1000	1000	527	769	423
MSR Action II	**Clap:** AP	0.528	0.563	0.687	**0.717**	**0.717**
	# Features	2000	2000	102	72	79
	HW: AP	0.630	0.699	0.741	0.832	**0.847**
	# Features	2000	2000	96	87	56
	Box: AP	0.716	0.680	0.810	**0.897**	0.852
	# Features	2000	2000	112	83	45

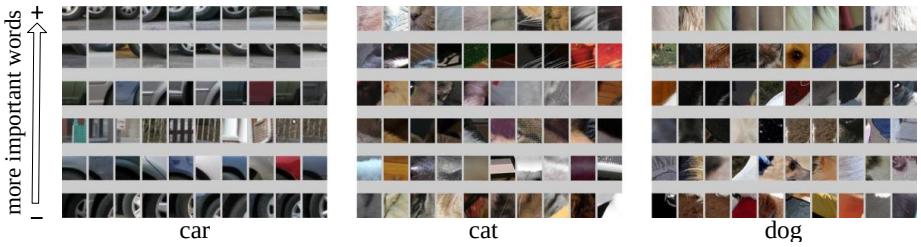


Figure 3: Patch visualization of top 6 visual words with highest weights in the feature selection. From left to right: Car, Cat, Dog. Each row line has 10 randomly selected patches corresponding to the visual word. From top to bottom, the weight changes from high to low.

could not be split temporally were always included in the testing set. We extracted STIP features [12] densely for each video. All the feature points were then quantized into 2000 words, which were obtained by applying K-means to 100000 training descriptors.
YouTube-Objects (YTO) [20] consists of videos collected from YouTube, see Fig. 2d. It contains 10 of the 20 classes in the PASCAL VOC. Tang *et al* [24] generated a ground truth set of 151 shots by manually annotating segments after the segmentation. We used the features in [24] that include histograms of dense-SIFT, histograms of RGB color, histograms of local binary patterns, histograms of dense optical flow, and heat maps.

5.1 Feature Selection

To validate the effectiveness of the proposed feature selection method for additive kernels, we compared our method with the following baselines: (i) Linear SVM; (ii) χ^2 kernel SVM; (iii) the linear feature selection for SVM of [18]; (iv) MKL using χ^2 ker-

nel [21], due to their connection with our method explained in Section 3. For MKL, each kernel is defined on one bin of the histograms.

For each method, parameters (e.g., C) of the SVM was chosen via cross-validation and we measured the classification performance using average precision (AP). To assess the complexity reduction achieved by feature selection, we also measured the number of selected features (i.e., non-zero weight). In this case, the features are the bins (clusters) in the BoW model. The results are presented in Table 1 for all datasets. These results show that the feature selection for χ^2 kernel SVM achieved the best average precision (AP) in all cases except 'Boxing', where it is outperformed only by the linear kernel. In all of our experiments, the number of selected features is significantly smaller than the original feature dimension.

A major goal of the paper is to illustrate that by performing feature and region selection, we can achieve a better interpretability of the BoW model. We visualized the selected visual words in the codebook for image datasets, in Fig. 3. From the feature selection results on the PittCar dataset, we can see that the most discriminative features mainly come from the wheels and doors of the cars. Note that the visual word with the fourth largest weight corresponds to the trunks of trees and fences. This is because trees occur more frequently in negative images than in positive images. As a result, this visual word is selected as a discriminative. For the cat and dog classes in PASCAL VOC dataset, several words latch on to cat and dog faces, while other visual words represent context (e.g., carpets) in which these animals usually appear. Since our method allows us to visualize the patches of visual words with their weights, the irrelevant words can be easily interpret by looking at the images in the dataset. From this example, we can see that feature selection can reveal which context the classifier is using for discriminating among classes.

5.2 Region Selection

As mentioned in Setion 4, region selection requires over-segmenting the images and videos first. For images, we used a hierarchical image segmentation to obtain superpixels [3]. For action localization on the MSR Action II, we followed [6] and used a regular voxel segmentation. For object localization on YTO dataset, we used the streaming hierarchical segmentation method of [27] to get supervoxels.

PittCar: Due to the connection of region selection to MIL approaches, we compared our region selection using linear and χ^2 kernels with two popular MIL methods, MI-SVM [19] and MILboost [9], on the PittCar dataset. We visualized the localization results in Fig. 4, from which we can see that MI-SVM tends to include the entire image. The performance of MI-SVM is better than MILboost. Visually, our region selection performs best among these methods.

To provide a quantitative measure for the localization performance, we compared all methods using precision-recall curves, as shown in Fig. 5. We used the area of overlap (AO) measure to evaluate the correctness of localization. For this criterion, a threshold t should be defined for AO to imply a correct detection. Usually, t is set as 0.5 [7]. However, this is unfair for methods that localize arbitrary shape, because the ground truth is a bounding box and such methods provide a shape mask, which can yield more accurate localization. We thus also set t to 0.4. The PR curves of

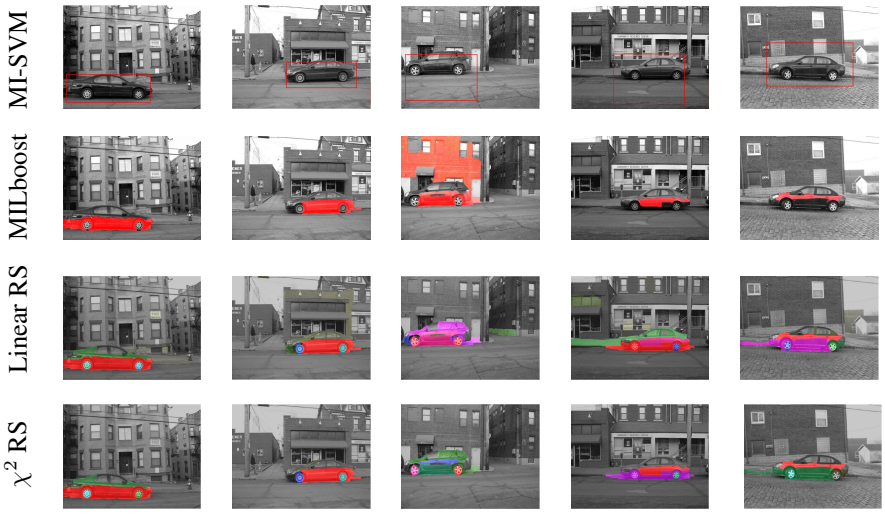
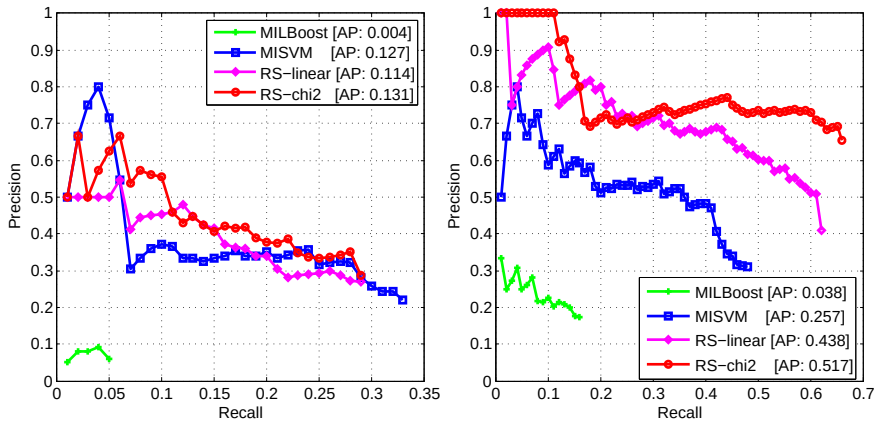


Figure 4: Region selection for PittsCar dataset. For our method (rows three and four) the color encodes the weights of the selected regions (warmer means higher); only regions with positive weights are colored. Images best seen in color.



(a) overlap threshold is 0.5 (b) overlap threshold is 0.4

Figure 5: Localization performance on the PittCar dataset.

different t values are shown in Fig. 5. We can see that our method and MI-SVM perform comparably when $t = 0.5$. For $t = 0.4$, the region selection method performs significantly better than the baselines. Also, our region selection method using χ^2 kernel performs better than with a linear kernel, which reinforces the usefulness of kernels in visual learning.

MSR Action II: Since it is unclear how to apply the MI-SVM proposed in [19] to video, we used the state-of-the-art method of Siva and Xiang [23] as a baseline.

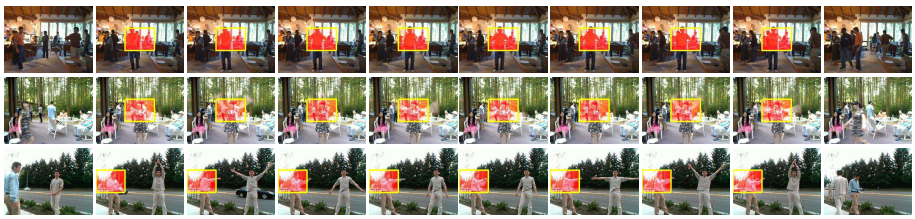


Figure 6: Localization examples on MSR action II dataset. Each row corresponds to randomly selected 10 frames in a video. Yellow bounding boxes are the localized actions in the videos.

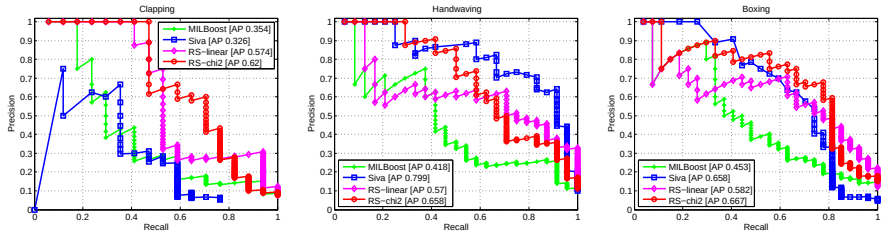


Figure 7: Localization performance on MSR Action II.

As in the previous experiment, we used precision-recall curve to evaluate the localization performance quantitatively. To ensure comparability, we replicate the setup of [23] and set the area of overlap (AO) to $1/8$. Qualitative and quantitative results are shown in Fig. 6 and Fig. 7 respectively. We can see that our region selection method using χ^2 kernel (RS-chi2) performs better than linear kernel (RS-linear). The region selection with a χ^2 kernel outperforms MILboost significantly and yields comparable results to Siva and Xiang [23]. Note, however, that our method is independent of the video-segmentation methods, whilst the method of Siva explicitly assumes the use of human detector.

YouTube-Objects: We also compared our region selection with CRANE [24] which is the state-of-the-art for object localization in videos. Here we use the χ^2 kernel in our method. The average precision for each class is shown in Tab. 2. We can see that our method gets better results on most of the vehicle categories and gets worse results on animal categories. The reason lies in the pre-segmentation. Since animals are often small in these videos and perform non-rigid motion, the segmentation method we used can not provide as good segmentation as that used in [24]. In general, however, our result is comparable to CRANE, which can be seen from the averaged PR curve over classes in Fig. 8. However, one should note that our method can be applied for region/feature selection, and results are comparable despite the fact that we have a worse segmentation algorithm.

Table 2: Average precision on YouTube-Objects dataset.

	aero	bird	boat	car	cat	cow	dog	horse	mbike	train	AVG.
CRANE	0.365	**0.363**	**0.271**	0.446	**0.250**	**0.334**	**0.345**	**0.286**	0.158	0.204	**0.292**
Ours	**0.426**	0.279	0.268	**0.612**	0.204	0.203	0.283	0.148	**0.202**	**0.263**	0.289

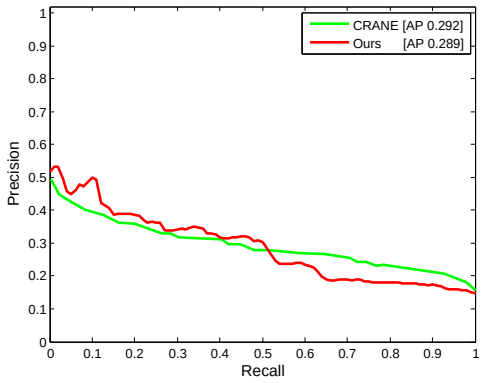


Figure 8: Localization performance on YouTube-Objects dataset.

6 Conclusions

This paper proposes a feature and region selection method for analysis and understanding of the BoW model. These methods can be used for visualizing discriminative features and regions. A major advantage of our feature selection is that we can select features in the kernel space by solving a convex problem. Our feature selection method is restricted to additive kernels, that are the most commonly used in visual classification tasks. In future work, we plan to address this limitation.

Beyond the classification performance, our feature selection method achieves better classification accuracy than state-of-the-art methods using significantly fewer number of features. Our region selection method provides a tool to visualize the regions that the classifier is weighting more aggressively to differentiate between class labels. In future work, we will explore the use of the region selection algorithm to provide weakly-supervised tools for labeling visual data that are faster and more reliable.

References

[1] G. I. Allen. Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics*, 22(2):284–299, 2013.

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[3] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2011.

[4] P. S. Bradley and O. L. Mangasaria. Feature selection via concave minimization and support vector machines. In *ICML*, 1998.

[5] B. Cao, D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Feature selection in a kernel space. In *ICML*, 2007.

[6] C.-Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *CVPR*, 2012.

[7] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88:303–338, 2009.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.

[9] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, 2008.

[10] Y. Grandvalet and S. Canu. Adaptive scaling for feature selection in SVMs. In *NIPS*, 2002.

[11] G. Hartmann, M. Grundmann, J. Hoffman, D. Tsai, V. Kwatra, O. Madani, S. Vijayanarasimhan, I. Essa, J. Rehg, and R. Sukthankar. Weakly supervised learning of object segmentations from web-scale video. In *ECCV Workshop on Web-Scale Vision*, 2012.

[12] I. Laptev. On space-time interest points. *IJCV*, 64(2):107–123, 2005.

[13] F. Li and C. Sminchisescu. The feature selection path in kernel methods. In *AISTATS*, 2010.

[14] Y.-F. Li, J. T. Kwok, I. W. Tsang, and Z.-H. Zhou. A convex method for locating regions of interest with multi-instance learning. In *ECML*, 2009.

[15] L. Liu and L. Wang. What has my classifier learned? Visualizing the classification rules of bag-of-feature model by support region detection. In *CVPR*, 2012.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] R. Mittelman, H. Lee, B. Kuipers, and S. Savarese. Weakly supervised learning of mid-level features with beta-bernoulli process restricted boltzmann machines. In *CVPR*, 2013.

[18] M. H. Nguyen and F. De la Torre. Optimal feature selection for support vector machines. *Pattern Recognition*, 43(3):584–591, 2010.

[19] M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother. Weakly supervised discriminative localization and classification: A joint learning process. In *ICCV*, 2009.

[20] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[21] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *JMLR*, 9:2491–2521, 2008.

[22] O. Russakovsky, Y. Lin, K. Yu, , and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012.

[23] P. Siva and T. Xiang. Weakly supervised action detection. In *BMVC*, 2011.

[24] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei. Discriminative segment annotation in weakly labeled video. In *CVPR*, 2013.

[25] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE TPAMI*, 34(3):480–492, 2012.

[26] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *CVPR*, 2008.
[27] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.
[28] O. Yakhnenko, J. Verbeek, and C. Schmid. Region-based image classification with a latent SVM model. *INRIA Technical Report*, 2011.
[29] W. Yang, Y. Wang, A. Vahdat, and G. Mori. Kernel latent SVM for visual recognition. In *NIPS*, 2012.
[30] J. Yuan, Z. Lin, and Y. Wu. Discriminative video pattern search for efficient action detection. *IEEE TPAMI*, 33(9):1728–1743, 2011.