Learning a Context Aware Dictionary for Sparse Representation

Farzad Siyahjani and Gianfranco Doretto

Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV 26506, USA
fsiyahja@mix.wvu.edu, gianfranco.doretto@mail.wvu.edu

Abstract. Recent successes in the use of sparse coding for many computer vision applications have triggered the attention towards the problem of how an over-complete dictionary should be learned from data. This is because the quality of a dictionary greatly affects performance in many respects, including computational. While so far the focus has been on learning compact, reconstructive, and discriminative dictionaries, in this work we propose to retain the previous qualities, and further enhance them by learning a dictionary that is able to predict the contextual information surrounding a sparsely coded signal. The proposed framework leverages the K-SVD for learning, fully inheriting its benefits of simplicity and efficiency. A model of structured prediction is designed around this approach, which leverages contextual information to improve the combined recognition and localization of multiple objects from multiple classes within one image. Results on the PASCAL VOC 2007 dataset are in line with the state-of-the-art, and clearly indicate that this is a viable approach for learning a context aware dictionary for sparse representation.

1 Introduction

Sparse coding has been used successfully in a variety of image processing, and computer vision applications, like image denoising [1], restoration [2], and classification [3, 4]. The main idea is to approximate a signal with an over-complete dictionary. The way in which such dictionary is learned from data greatly affects the task at hand. In object recognition, one of the central computer vision tasks, the dictionary is learned under stringent conditions dictated by the nature of the problem. In particular, the number of training data samples is very large, and high classification performance requires a dictionary with discriminative power.

Sparse coding learns a dictionary while maximizing the reconstructive power given the sparsity constraint. With large training datasets this leads to large dictionaries, which is a problem when the input data is given to the sparse solver in charge of computing the corresponding sparse code. The successful SRC algorithm [3] deals with large datasets by manually selecting the training samples used to construct the dictionary. This is suboptimal in many respects [5], and it does not scale well with the number of object classes to recognize, which

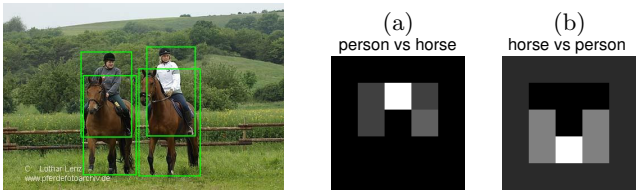

Fig. 1. Context aware sparse code. An over-complete context aware dictionary allows taking the sparse representation of an object of a given class (delimited by a bounding box), and using it for predicting the layout distribution of objects of another class. This means that, given the sparse code of a horse, one would be able to tell where will likely be located a person in close proximity (i.e. on top) (see bright spots in layout distribution (a)). Conversely, given the sparse code of a person, one should be able to tell where will likely be located a horse in close proximity (i.e. below) (see bright spots in layout distribution (b)).

requires a proportional increase of the dataset size. To address this problem, small-size dictionary learning has been approached by [6–9, 5, 10]. These works deploy or extend either the method of optimal directions (MOD) [8] or the K-SVD algorithm [9].

Supporting recognition tasks requires a discriminative dictionary. On the other hand, the K-SVD algorithm, which has established itself for its simplicity, numerical efficiency, and fast convergence, focusses on learning an over-complete dictionary that maximizes the reconstructive power. Rather than decoupling reconstructive and discriminative properties by first learning a dictionary for representation and then training a classifier, like it is done in [6, 11, 12], recent works [5, 10] have achieved state-of-the-art performance by elegantly extending the K-SVD algorithm to simultaneously learn a dictionary that is compact, reconstructive, and discriminative.

In this paper we propose to further extend the K-SVD algorithm to learn an over-complete dictionary from a set of labeled training images, so that spatial contextual information describing the presence of other objects becomes predictable by the dictionary (see Figure 1). It is well known that contextual information boosts object recognition performance [13–20]. (e.g. it is more likely to find a bottle over a table than over an airplane). Modeling contextual information is even more important when an image has a large number of objects. This is because context is better learned and leveraged [13, 21]. Also, context works more effectively when big objects are easily detected first (e.g. a table), while smaller ones (e.g. a bottle) are detected last [21, 22]. Careful modeling also leads to an improved detection of objects embedded in unconventional context [21] (e.g. a sofa on the street). Finally, there is a growing belief that context is very effective for extracting higher-level semantic information. For instance, modeling the interaction between objects leads to the interpretation of actions [13, 19] (e.g. a man playing tennis requires to detect a person and a tennis racket in a suitable relative position).

We present a supervised algorithm that for training starts from a dataset of images where objects and their positions (bounding boxes) are known, and incorporates these labels directly into the dictionary learning stage. The resulting objective function is optimized with the K-SVD algorithm. Therefore, the learned dictionary is still compact, reconstructive, and discriminative, and not just representational like traditional approaches [2, 9, 8, 3]. In particular, like [5, 10] our approach learns a simple multiclass linear classifier, in contrast to approaches that learn multiple binary classifiers [4, 6, 23, 24]. More importantly, a collection of linear regressors is also learned, such that the dictionary is able to map sparse codes to the space of layout distributions of objects. The dictionary, the classifier, and such layout maps are learned simultaneously, as opposed to approaches that require iterations between subsets of parameters, or separate the learning of the dictionary from the rest [23, 24, 6, 12]. This is highly desirable, and given the efficiency of the K-SVD, it leads to solutions that scale well with the number of object categories, and tend to avoid local minima.

To demonstrate the effectiveness of the method, we design a framework of structured prediction that addresses the problem of correctly detecting and localizing in an image, multiple objects from multiple classes. As input, such framework takes an image with multiple bounding box hypotheses[1], and assigns an object class label to all of them simultaneously. Even with a simplified model we show that it is possible to estimate the object layout at a very high speed, and obtain performance in line with the state-of-the-art on the PASCAL VOC 2007 dataset [25].

The paper is organized as follows. Section 2 explains how to learn a reconstructive and discriminative over-complete dictionary for sparse coding. Section 3 shows how to learn a context aware dictionary. Section 4 introduces the structured prediction framework, and Section 5 validates the proposed approach.

2 Discriminative Dictionary Learning

This section briefly reviews how to construct a dictionary for sparse coding, and later explains how to add discriminative power to it. Given a set $X \doteq [x_1, \cdots, x_N] \in \mathbb{R}^{n \times N}$ of N n-dimensional signals, a reconstructive over-complete dictionary $D \doteq [d_1, \cdots, d_k] \in \mathbb{R}^{n \times k}$ with $k \gg n$ atom elements for the sparse representation of X is learned by solving the problem

$$< D, A >= \arg\min_{D,A} \|X - DA\|_2^2 \quad \text{s.t.} \ \ \|A\|_0 < T \ , \tag{1}$$

where $A \doteq [a_1, \cdots, a_N] \in \mathbb{R}^{k \times N}$ is the sparse representation of the signals in X. The cost function that is being minimized is the reconstruction error, while the sparsity constraint imposes that the ℓ^0-norm of A should be less than T. This means that every sparse code a_i, representing x_i, should have no more that T elements different than zero. Problem (1) can be solved efficiently by the K-SVD

[1] Bounding box hypotheses can be obtained with a sliding window detector.

algorithm [9]. Once the dictionary D is available, the sparse representation A of the signals X can be computed by applying a matching pursuit method, for instance the orthonormal matching pursuit (OMP) [26], to solve the following problem

$$< A >= \arg\min_{A} \|X - DA\|_2^2 \quad \text{s.t.} \;\; \|A\|_0 < T \; . \tag{2}$$

2.1 Adding Discriminative Power

The dictionary learned through (1) has good reconstructive properties. When used for a recognition task it should also have discriminative properties. One way to "endow" the dictionary with those is to use the sparse codes A as features for classification. This requires training a classifier $f(a, G)$, where its parameters G satisfy

$$< G >= \arg\min_{G} \sum_{i} \mathcal{L}\{y_i, f(a_i, G)\} + \lambda \|G\|_2^2 \; , \tag{3}$$

and where y_i is the label associated with the feature a_i, $\mathcal{L}$ is the classification loss function (for which there are several choices [23, 24, 7, 5]), and λ is a parameter that sets the strength of the regularizing term.

The disconnection between dictionary and classifier learning means that the dictionary may originate only suboptimal features for classification. This issue can be significantly mitigated if the learning of the classifier and of the dictionary happen simultaneously, by combining the cost functions in (1) and (3), leading to

$$\begin{aligned} < D, A, G >= \arg\min_{D,G,A} & \|X - DA\|_2^2 \\ & + \sum_{i} \mathcal{L}\{y_i, f(a_i, G)\} + \lambda \|G\|_2^2 \quad \text{s.t.} \;\; \forall i, \|a_i\|_0 \leq T \; . \end{aligned} \tag{4}$$

Several approaches use a similar architecture [23, 24, 7, 5, 10]. Most of these algorithms design fairly elaborate procedures to solve (4), where subproblems compute temporary estimations of a subset of parameters, increasing the possibility for the optimization to not converge to the right solution.

A substantial difference in approaching problem (4) has been introduced by [5]. In particular, they advocate the use of a simple quadratic cost function, and a simple linear classifier. The outcome is the possibility to solve (4) very elegantly with the K-SVD algorithm [9], inheriting all the computational benefits it comes with.

One concern, not explicitly addressed in [9], is the fact that in object recognition, given the large intra-class and inter-class variation of the observations, the size of the training dataset may significantly increase with the number of classes. The result is that maintaining the same discriminative power as the number of classes grows, forces a proportional increase of the dictionary size, which is undesirable. The work in [10] addresses this problem by building on the approach in [9]. They are able to learn a dictionary, simultaneously with a multi-class

linear classifier, that has a size that scales well with the number of classes, while maintaining excellent performance. This is achieved simply by adding a clever regularizing term to the cost function to optimize, which is linear in the sparse codes A. In addition to that, similarly to [9], [10] solves (4) with the K-SVD algorithm, inheriting again all the benefits that come with it.

3 Context Aware Dictionary Learning

In this section we are interested in extending the framework introduced in Section 2, and learn a dictionary that is "endowed" with the ability to predict the context of a given signal x_i, where x_i might represent an object or a region in an image.

So far we made the implicit assumption that a signal x_i can be represented by a sparse code a_i, according to an over-complete dictionary D. We also assumed that the same signal is a sample from a class identified by the label $y_i \in \{1, \cdots, c\}$. If x_i represents an object i of class y_i in an image, predicting its context means being able to provide the likelihood information of whether another object, of a given class j, is located in a certain position p, with respect to object i. We indicate this likelihood with $l_{pi}(j)$. Also, with $l_i(j) \doteq [l_{1i}(j), \cdots, l_{mi}(j)]^\top$ we indicate the set of likelihoods describing the full spatial layout distribution of where an object of class j might be located around object i.

Following the reasoning in Section 2, one way to exploit the dictionary to predict the spatial layout distribution of a class j around an object, is to consider the sparse codes A as features for a regression model $g(a, W(j))$, where its parameters $W(j)$ satisfy the following

$$< W(j) >= \arg\min_{W(j)} \sum_i \mathcal{M}\{l_i(j), g(a_i, W(j))\} + \nu \|W(j)\|_2^2 \,. \tag{5}$$

Here ν is a scalar that sets the strengths of the regularizing term, whereas $\mathcal{M}$ is a regression loss function of our choice.

If we were to proceed along the path just highlighted, we would create a disconnection between the dictionary and the regression learning in that the dictionary would originate only suboptimal features for regression. Again, this issue can be mitigated if the learning of the regression and of the dictionary happen simultaneously by combining the cost functions in (1) and in (5). However, since we are interested in learning also a discriminative dictionary, we combine the cost functions with the one in (3) as well. This leads to the following training problem

$$\begin{aligned} < D, A, G, W(:) >= \arg \min_{D,G,A,W(:)} & \|X - DA\|_2^2 \\ & + \sum_i \mathcal{L}\{y_i, f(a_i, G)\} + \sum_j \sum_i \mathcal{M}\{l_i(j), g(a_i, W(j))\} \\ & + \lambda \|G\|_2^2 + \nu \sum_j \|W(j)\|_2^2 \quad \text{s.t. } \forall i, \|a_i\|_0 \leq T \,. \end{aligned} \tag{6}$$

For estimating the parameters, we are interested in leveraging the lesson learned from [5, 10] and convert (6) into a problem approachable with the K-SVD. We proceed by picking quadratic loss functions for $\mathcal{L}$ and $\mathcal{M}$, and we chose a linear multi-class classifier parameterized by $G \in \mathbb{R}^{c\times k}$, i.e. $f(a, G) \doteq Ga$. Also, we select a linear regression model parameterized by $W(j) \in \mathbb{R}^{m\times k}$, i.e. $g(a, W(j)) \doteq W(j)a$. In addition, we define the following variables: $Y \doteq [y_1, \cdots, y_N] \in \mathbb{R}^{c\times N}$, and $L(j) \doteq [l_1(j), \cdots, l_N(j)] \in \mathbb{R}^{m\times N}$, where a label $y_i = b$ now is represented as a vector with $c-1$ zeros and a 1 in its b-th component, i.e. $y_i = [0, \cdots, 0, 1, 0, \cdots 0]^\top$. Finally, we define the following matrices

$$L \doteq \left[L(1)^\top, \cdots, L(c)^\top\right]^\top \ , \quad W \doteq \left[W(1)^\top, \cdots, W(c)^\top\right]^\top \ . \tag{7}$$

Given the choices outlined above, problem (6) can be expressed as follows

$$\begin{aligned} < D, A, G, W > = \arg \min_{D,G,A,W} & \|X - DA\|_2^2 + \alpha\|Y - GA\|_2^2 + \lambda\|G\|_2^2 \\ & + \beta\|L - WA\|_2^2 + \nu\|W\|_2^2 \quad \text{s.t. } \|A\|_0 \leq T \ , \end{aligned} \tag{8}$$

where α and β are tuning parameters setting the strength of the classification error versus the regression error versus the reconstruction error.

Note that this formulation of the learning problem does not include the label-consistent term presented in [10]. This is done to keep the exposition of this work more focused on the newly proposed contribution. However, the reader should be warned that the term proposed in [10] could be added to the cost function in (8), producing the beneficial effects discussed in Section 2. The following section explains how to address (8) with the K-SVD.

3.1 Learning with K-SVD

As a first step, we notice that the cost function in (8) can be rewritten as

$$\begin{aligned} < D, A, G, W > = \arg \min_{D,G,W,A} & \left\| \begin{pmatrix} X \\ \sqrt{\alpha}Y \\ \sqrt{\beta}L \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\alpha}G \\ \sqrt{\beta}W \end{pmatrix} A \right\|_2^2 \\ & + \lambda\|G\|_2^2 + \nu\|W\|_2^2 \qquad \text{s.t. } \|A\|_0 \leq T \ , \end{aligned} \tag{9}$$

and a more compact expression is obtained by setting $\mathcal{X} = (X^\top \ \sqrt{\alpha}Y^\top \ \sqrt{\beta}L^\top)^\top$ and $\mathcal{D} = (D^\top \ \sqrt{\alpha}\, G^\top \ \sqrt{\beta}W^\top)^\top$, leading to

$$< \mathcal{D}, A > = \arg \min_{\mathcal{D},A} \|\mathcal{X} - \mathcal{D}A\|_2^2 + \lambda\|G\|_2^2 + \nu\|W\|_2^2 \qquad \text{s.t. } \|A\|_0 \leq T \ . \tag{10}$$

Finally, since the K-SVD proceeds by iteratively updating each column of $\mathcal{D}$, while maintaining them normalized, the last two regularizing terms are dropped, meaning that λ and ν can be set to 0, and the only free parameters in Equation (10) are two: α and β. To summarize the optimization algorithm we indicate with b_k and a^k the k-th column and row of $\mathcal{D}$ and A, respectively, and define

$E_k \doteq (\mathcal{X} - \sum_{j \neq k} b_j a^j)$. Also, $\tilde{a}^j$ indicates the row vector a^j after removing the zero entries, and $\tilde{E}_k$ is the error E_k after removing the columns in corresponding positions. Following the K-SVD protocol, b_k and $\tilde{a}^k$ are updated by solving

$$< b_k, \tilde{a}^k >= \arg\min_{b_k, \tilde{a}^k} \|\tilde{E}_k - b_k \tilde{a}^k\|_2^2 \,, \tag{11}$$

which can be computed in closed form by decomposing the error $\tilde{E}_k \overset{SVD}{=} U \Sigma V^\top$, and setting (using Matlab notation) $b_k = U(:,1)$, and $\tilde{a}^k = \Sigma(1,1)V(:,1)^\top$. $\tilde{a}^k$ is then used to replace the non-zero values in a^k. After a complete update of $\mathcal{D}$, the sparse code matrix A is updated with a matching pursuit algorithm, and the iteration repeats until convergence. This process maintains the columns of $\mathcal{D}$ normalized. Therefore, similarly to what is done in [5], the columns of D have to be re-normalized, and the columns of G and W have to be re-scaled accordingly.

The iteration requires to initialize the values of D, G, and W. The dictionary can be obtained by simply applying the traditional K-SVD algorithm to the input data X, which means solving for problem (1). The classifier G can then be obtained by solving problem (3) with a quadratic loss function, and the regression maps W are computed through (5), again with a quadratic loss function. The solution to (3) and (5) is given in closed form by [27], and more precisely by

$$G = (AA^\top + \lambda I)^{-1} A Y^\top \,, \quad W = (AA^\top + \nu I)^{-1} A L^\top \,. \tag{12}$$

This learning approach allows to simultaneously estimate all the parameters with a procedure that tends to avoid local minima and is numerically efficient. Through the use of robust and sequential algorithms for computing the SVD [28] it is possible to scale well with the size of the training dataset and with the number of classes to handle, even if the amount of parameters to estimate is large. At the same time, it is possible to impose the desired size of the dictionary while maintaining it reconstructive, discriminative, and context aware. As mentioned before, by adding the contribution proposed in [10], it is possible to obtain even more compact dictionaries without the expense of loosing discriminative power, which further increases scalability with respect to the number of classes.

4 Simultaneous Object Localization and Recognition

In this section we are interested in using the context aware dictionary introduced in Section 3 to design a structured prediction framework able to simultaneously localize and recognize objects in images.

All the notation developed so far is still valid, what changes here is that we consider an image I depicting M objects, each of which is represented by a feature vector x, so $X = [x_1, \cdots, x_M]$ is now the set of features, or signals describing the objects in the image I. Assuming that a dictionary D has already been learned, the sparse codes $A = [a_1, \cdots, a_M]$ representing the objects can be computed by (2). The feature x could be a simple histogram of oriented gradients (HOG) descriptor [29], which is indeed what was used in our experiments.

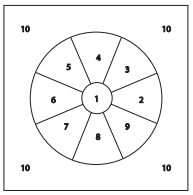

Fig. 2. Spatial context subdivision. The surrounding space of an object is divided into 11 regions. Regions 1 and 10 indicate very-near and far respectively. Region 10 is delimited by the size of the image, and region 11, not indicated, is the space outside the image. Other regions identify also the angular relation between objects.

The descriptors are computed over the M bounding boxes surrounding the hypothetical M objects. The boxes could be provided by a bank of sliding window detectors, or a selective search procedure [30].

For a given object i in image I, $l_i(j)$ describes the layout distribution of objects of class $j \in \{1, \cdots, c\}$, around object i. Figure 2 shows how the space around object i is divided in $m = 11$ regions. The inner circle has a diameter corresponding to half of the diagonal of the bounding box in the image with average size, and the outer circle has a diameter that is three times the same diagonal. Region 10 is delimited by the size of the image, while region 11 is anywhere outside of the image. When the center location of an object of class j falls into a given region, its presence is recorded into the corresponding bin of $l_i(j)$.

Given the framework above, the dictionary D, the classifier G, and the layout maps W, we want to design a score function $S(Y, A)$ that, given the sparse codes A, when optimized with respect to $Y = [y_1, \cdots, y_M]$ produces the correct labels for the set of M boxes in the image. One way to proceed is to make the following two observations about object i, represented by the sparse code a_i: (a) the label y_i should be predictable by the classifier Ga_i; (b) the layout distribution around object i of objects of class j, $l_i(j)$, should be predictable by $W(j)a_i$. This should hold for every object in the image. Therefore, the score function that we are looking for could be expressed as

$$S(Y, A) = \sum_i \|y_i - Ga_i\|_2^2 + \gamma \sum_i \sum_j \|l_i(j) - W(j)a_i\|_2^2 \,, \tag{13}$$

where the first term minimizes the classifier error, and the second term the layout map error. The coefficient γ is a parameter to be learned from data and establishes a balance between the terms.

While (13) is certainly a good option, we design a simplified model that will provide considerable computational advantages. In particular, we change the way we treat contextual information around object i, and we say that an object j should be able to predict the layout $l_j(y_i)$ of the objects of class y_i around object j, which is given by $W(y_i)a_j$. This should hold for every object. Therefore, the score function associated with this model is given by

$$\begin{aligned} S(Y, A) &= \sum_i \|y_i - Ga_i\|_2^2 + \frac{\gamma}{M} \sum_i \sum_j \|l_j(y_i) - W(y_i)a_j\|_2^2 \,, \\ &= \sum_i \|y_i - Ga_i\|_2^2 + \frac{\gamma}{M} \sum_i \|L(y_i) - W(y_i)A\|_2^2 \,, \end{aligned} \tag{14}$$

where, again, γ is a parameter to be learned from training data, and M here is necessary for normalization purposes.

A fundamental advantage of model (14) versus model (13) is the fact that the score function can be written as $S(Y, A) = \sum_i S(y_i, A)$, where

$$S(y_i, A) = \|y_i - Ga_i\|_2^2 + \frac{\gamma}{M}\|L(y_i) - W(y_i)A\|_2^2 \,, \tag{15}$$

and every label y_i can be determined individually, and very efficiently with a simple linear search, by optimizing

$$< y_i >= \arg\min_{y_i} S(y_i, A) \quad \text{for } i = 1, \cdots, M \,. \tag{16}$$

Estimating one label at a time is a considerable advantage with respect to model (13). In fact, optimizing (13) would require a full search over the joint space of the labels of all objects. In alternative, one could start from an initial guess provided by the classifier, and then iteratively update each label one at a time, with all the risks of converging only to a local minimum.

4.1 Balancing the context

The parameter γ in Equation (15) can be estimated in closed form from training data to effectively balancing the contribution of the context term in the score function. Indeed, the derivative of a score, computed in correspondence of the correct label assignment, where it is minimum, should be zero or, in presence of noise, as close to zero as possible. Therefore, given the set of training images $\{I^q\}$ and corresponding label annotations $\{y_i^q\}$, γ is estimated by solving the following least squares problem

$$< \gamma >= \arg\min_{\gamma} \sum_q \sum_{i=1}^{M^q} \left\| \frac{\partial S(y_i^q, A^q)}{\partial y_i^q} \right\|_2^2 \,. \tag{17}$$

The solution, which we do not report because of lack of space, is computed in closed form, and entails evaluating the derivatives of the score function. The most important step in the derivation is to model $l_j(y_i)$ and $W(y_i)$ as $W(y_i) = ({y_i}^\top \otimes I_m)W$, and $l_j(y_i) = (y_i^\top \otimes I_m)L_j$, where $L_j \doteq [l_j(1)^\top, \cdots, l_j(c)^\top]^\top \in \mathbb{R}^{mc \times 1}$ represents the complete layout of object j. At this point one can relax the binary components of y_i and continue the derivation.

5 Experiments

In order to test the new dictionary learning procedure, and to be comparative with related work, we evaluate the proposed approach on the PASCAL Visual Object Challenge (VOC) 2007 dataset [31]. The dataset consists of about 10000 images of 20 different object classes. About half of them are used for training

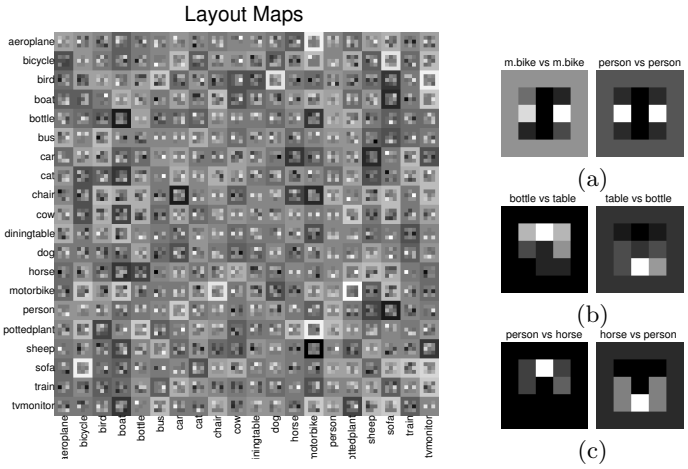

Fig. 3. Layout distributions. A given patch element of the left image shows, for a given class on the horizontal axis, the spatial layout probability to find a given class of the vertical axis. Each patch shows 10 color coded bins corresponding to the 10 regions highlighted in Figure 2, depicting the layout distribution $E[W(j)a]$, where j is the vertical axis, and a is the sparse code of a class in the horizontal axis. Lighter means higher probability. Note that the 11-th bin is not visualized because the probability to find an object of another class very far away is always very high and this biases the color coded visualization of the "within the image" layout distribution. (a), (b), and (c) are close-ups of the layout distributions. See Section 5 for more information.

while the other half are used for testing. The training set consists of approximately 16000 objects in about 5000 images. In addition, since our model uses a background class, background boxes are randomly sampled from the training set, and are selected in a way that they cannot constitute a detection according to the PASCAL evaluation protocol, which says that a detection is considered correct if its bounding box overlaps 50% or more with the groundtruth bounding box. Other general settings, including the type of features extracted from the bounding boxes (i.e., HOG [29]), and how the boxes hypothesis are generated, follow the evaluation in [13].

The training procedure illustrated in Section 3 is affected by the parameters α and β. In our experiments we conducted a grid search in order to find good values for them. The sparsity factor is set to $T = 50$, and the dictionary size is set to $k = 2100$. For the VOC07 training dataset our non-optimized Matlab implementation spent 38% of the time running matching pursuit, and was taking 2.5 minutes to complete one full K-SVD update iteration in a high-end PC. About 15 iterations were enough to converge. During testing, the elapsed time on assigning labels to the boxes depends on the number of objects in an image. Our implementation takes from 0.1 to 0.5 seconds.

Figure 3 is meant to illustrate the results from learning the layout maps $W(j)$. In particular, given an object represented by the sparse code a_i, the quantity $W(j)a_i$ represents the likelihood to find an object of class j around the object

Class	Baseline [32]	[13]	[21]	No-context	Context
plane	0.278	0.288	0.320	**0.411**	0.396
bike	0.559	**0.562**	0.505	0.401	0.420
bird	0.014	0.032	0.089	0.121	**0.156**
boat	0.146	0.142	0.149	0.240	**0.259**
bottle	0.257	0.294	0.252	0.288	**0.320**
bus	0.381	**0.387**	0.369	0.340	0.357
car	0.470	0.487	0.467	0.522	**0.553**
cat	0.151	0.124	**0.189**	0.118	0.116
chair	0.163	0.160	0.181	0.198	**0.212**
cow	0.167	0.177	**0.182**	0.124	0.135
table	0.228	**0.240**	0.229	0.132	0.129
dog	0.111	0.117	0.124	0.110	**0.135**
horse	0.438	0.450	**0.472**	0.331	0.357
motbike	0.373	0.394	**0.418**	0.281	0.317
person	0.352	0.355	0.354	0.364	**0.399**
plant	0.140	0.152	**0.156**	0.144	0.125
sheep	0.169	0.161	**0.218**	0.138	0.137
sofa	0.193	0.201	**0.204**	0.175	0.183
train	0.319	0.342	**0.388**	0.279	0.354
TV	0.373	0.354	0.357	0.374	**0.394**
Avg	.264	.272	0.278	.255	.273

Table 1. Per-class AP scores on the PASCAL VOC 2007 dataset [31]. The first column is our baseline [32]. The second and third columns show the results from [13], and from [21], respectively. The last two columns show the results with "No-context", which means that only the multi-linear classifier was used, and the last column shows the results from the full "Context" model introduced in Section 4.

i. In Figure 3 the vertical axis represents the object class j, whereas the horizontal axis represents the class of the object i. Therefore, in the table showing the layout maps, the row corresponding to the class j shows the average likelihood distribution $E[W(j)a_i]$, for a sparse code a_i that comes from each of the 20 classes of the dataset. Each of the entry $W(j)a_i$ in the 20×20 table is represented by an image patch, where each pixel element is color coded with the likelihood of the corresponding layout bin. Lighter areas mean higher likelihood. Figure 3-(a)(b)(c) show a close-up view of a few layout distributions. In particular, Figure 3-(b)(c) highlight that in the dataset there certainly are enough examples showing bottles on tables, and people riding horses, as it appears from the layout distributions. Similar interpretations hold for the other examples. More importantly, this is a clear indication that the proposed approach is able to capture these types of contextual constraints. In addition, Figure 3-(b)(c) show that inverting the order of the two classes considered leads to a flipping of the distribution with respect to the center. This is expected (e.g. if bottles are likely to sit on tables, then tables are likely to hold bottles, and viceversa), and it gives another confirmation that the model is capturing the correct information.

Figure 4 shows the precision recall (PR) curves obtained following the `comp3` protocol of the VOC Challenge. Here we report only the results of four classes,

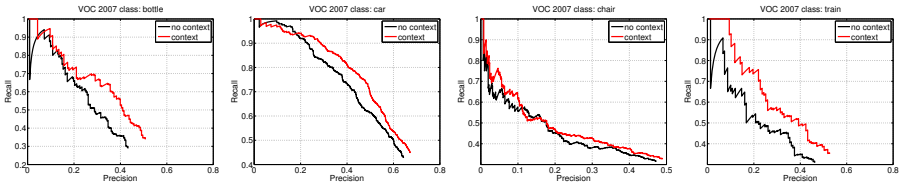

Fig. 4. Precision Recall curves. PR curves are obtained following the `comp3` protocol of the VOC Challenge 2007. Each graph shows results obtained with the full model (context), and with the reduced model (no-context) on a particular class. From left to right, each graph pertains to the following classes: bottle, car, chair, and train, respectively.

namely bottle, car, chair, and train. 15 classes exhibited a notable improvement in the precision recall curves by adding context. Table 1 shows the corresponding per class average precision (AP) for all the 20 classes. The table includes the AP of the baseline algorithm, which is the same one used in [13, 21], and we also report the results of [13] and [21]. We have included AP values for the full model, indicated with "Context", and for the reduced model where only the multi-linear classifier is used, which is indicated with "No-context". The full model leads to a 7% increase of the average AP over the "No-context" model. As it can be seen, the context aware dictionary, applied according to the structured output framework proposed in Section 4, shows the potential to be able to compete with state-of-the-art algorithms, and in 8 classes it achieves the highest AP score. As a multi-class score, we have also computed the per-image AP and obtained 0.391, which is again in line with the state-of-the art. Finally, Figure 5 shows samples from the testing dataset with overlayed the bounding boxes with confidence rank higher than a certain threshold, and with the corresponding labels assigned by the structured prediction framework.

6 Conclusions

This work has presented a technique for learning an over-complete dictionary for sparse representation of signals that is compact, reconstructive, discriminative, and that simultaneously learns a set of linear regression maps. Even though such maps are suitable for modeling and predicting contextual information, their use is not restricted to this scenario. Important properties about this framework include numerical stability, and the ability to scale well with the number of parameters to learn, and with the size of the dataset. This is mainly due to the use of the K-SVD, and robust online algorithms for computing the SVD. We tested this idea by designing a structured prediction framework for simultaneously localizing and recognizing multiple objects from multiple classes, and that exploits the sparse domain representation to perform a very fast assignment of the class labels. We have shown that the approach is able to learn meaningful layout distributions. Moreover, even with a non-optimized extraction of the HOG features, we have verified that the method is on par with similar state-of-the-art approaches. In particular, on the VOC07 dataset, exploiting the context

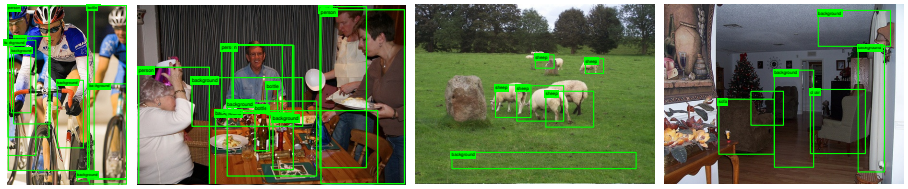

Fig. 5. Object localization and recognition. Samples from the PASCAL VOC 2007 dataset showing the labeling of the bounding boxes hypothesis with top confidence rank, computed by the proposed structured prediction framework.

aware dictionary has led to a 7% increase of the per-class mean average precision, with respect to not using contextual information. These results are a clear indication that the basic idea of the approach is valid and worth exploiting in future developments.

Finally, we would like to point out that by incorporating in our framework the label-consistent contribution proposed in [10], it would be possible to increase the discriminative power and improve the compactness of the dictionary. This would benefit the scalability of the approach with respect to the number of classes. We leave for future work testing this extension with the SUN09 dataset [21], which includes images from many more object classes than the VOC07, and is well suited for testing models of contextual interactions.

References

1. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE TIP **15** (2006) 3736 –3745
2. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. IEEE TIP **17** (2008) 53 –69
3. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE TPAMI **31** (2009) 210 –227
4. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE CVPR. (2009) 1794 –1801
5. Zhang, Q., Li, B.: Discriminative k-svd for dictionary learning in face recognition. In: IEEE CVPR. (2010) 2691 –2698
6. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: IEEE CVPR. (2008) 1 –8
7. Pham, D.S., Venkatesh, S.: Joint learning and dictionary construction for pattern recognition. In: IEEE CVPR. (2008) 1 –8
8. Engan, K., Aase, S., Husoy, J.: Frame based signal compression using method of optimal directions (mod). In: IEEE ISCAS. Volume 4. (1999) 1 –4 vol.4
9. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE TSP **54** (2006) 4311 –4322
10. Jiang, Z., Lin, Z., Davis, L.S.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: IEEE CVPR. (2011) 1697–1704
11. Huang, H., Aviyiente, S.: Sparse representation for signal classification. In: NIPS. (2007)
12. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: IEEE CVPR. (2010) 2559 –2566

13. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. IJCV **95** (2011) 1–12
14. Divvala, S., Hoiem, D., Hays, J., Efros, A., Hebert, M.: An empirical study of context in object detection. In: IEEE CVPR. (2009) 1271 –1278
15. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE TPAMI **32** (2010) 1627 –1645
16. Galleguillos, C., McFee, B., Belongie, S., Lanckriet, G.: Multi-class object localization by combining local contextual interactions. In: IEEE CVPR. (2010) 113 –120
17. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: IEEE ICCV. (2007) 1 –8
18. Torralba, A.: Contextual priming for object detection. IJCV **53** (2003) 169–191 10.1023/A:1023052124951.
19. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: IEEE CVPR. (2010) 17 –24
20. Sadeghi, M., Farhadi, A.: Recognition using visual phrases. In: IEEE CVPR. (2011) 1745 –1752
21. Choi, M.J., Lim, J., Torralba, A., Willsky, A.: Exploiting hierarchical context on a large database of object categories. In: IEEE CVPR. (2010) 129 –136
22. Park, D., Ramanan, D., Fowlkes, C.: Multiresolution models for object detection. In: ECCV. (2010)
23. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. In: NIPS. Volume abs/0809.3083. (2008)
24. Yang, J., Yu, K., Huang, T.: Supervised translation-invariant sparse coding. In: IEEE CVPR. (2010) 3517 –3524
25. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88** (2010) 303–338
26. Chen, S., Cowan, C., Grant, P.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE TNN **2** (1991) 302 –309
27. Golub, G.H., Hansen, P.C., O'Leary, D.P.: Tikhonov regularization and total least squares. SIAM J. Matrix Anal. Appl. **21** (1999) 185–194
28. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve basis extraction and its application to images. IEEE TIP **2** (1998) 456–460
29. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR. Volume 1. (2005) 886–893
30. van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: IEEE CVPR. Volume 0. (2011) 1879–1886
31. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (2007)
32. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: IEEE CVPR. (2008) 1 –8