Automatic discovery of discriminative parts as a quadratic assignment problem
Ronan Sicre, Julien Rabin, Yannis Avrithis, Teddy Furon, Frédéric Jurie, Ewa Kijak

To cite this version:
Ronan Sicre, Julien Rabin, Yannis Avrithis, Teddy Furon, Frédéric Jurie, et al.. Automatic discovery of discriminative parts as a quadratic assignment problem. ICCV Workshops – CEFRL, Oct 2017, Venise, Italy. pp.1059-1068. hal-01623148

HAL Id: hal-01623148
https://hal.archives-ouvertes.fr/hal-01623148
Submitted on 25 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Automatic discovery of discriminative parts as a quadratic assignment problem

Ronan Sicre
IRISA, France

Julien Rabin
Normandie Univ., France

Yannis Avrithis
Inria, France

Teddy Furon
Inria, France

Frédéric Jurie
Normandie Univ., France

Ewa Kijak
Univ. Rennes 1 / IRISA, France

Abstract

Part-based image classification consists in representing categories by small sets of discriminative parts upon which a representation of the images is built. This paper addresses the question of how to automatically learn such parts from a set of labeled training images. We propose to cast the training of parts as a quadratic assignment problem in which optimal correspondences between image regions and parts are automatically learned. The paper analyses different assignment strategies and thoroughly evaluates them on two public datasets: Willow actions and MIT 67 scenes.

1. Introduction

The representation of images as set of patches has a long history in computer vision, especially for object recognition [2], image classification [9, 36] or object detection [18]. Its biggest advantages are the robustness to spatial transformations (rotation, scale changes, etc.) and the ability to focus on the important information of the image while discarding clutter and background.

Part-based classification raises the questions of i) how to automatically identify what are the parts to be included in the model and ii) how to use them to classify a query image. The work of [38] selects informative patches using an entropy based criterion while the decision relies on a Bayes classifier. Following [38], recent approaches separate the construction of the model (i.e. the learning of the parts) and the decision function [14, 8].

Jointly optimizing modeling and classification is however possible for simple enough part detectors and decision functions [24]. This work aims at defining the parts directly related to the final classification function.

While this argument is understandable, the objective function of this joint optimization is highly non-convex with no guaranty of convergence. Deciding which alternative is better – the joint or separate design – is still an open problem. As an insight, the two stage part-based model of [30] performs better than the joint learning of [24]. We note other differences: [24] models both positive and negative parts while [30] focuses only on the positive ones.

Interestingly, [30, 29] addresses the learning of parts as an assignment problem. Regions are sampled randomly from the training images, and a class is modeled as a set of parts. The assignment region-part is constrained: each part is assigned to one region in each positive image (belonging to the class to be modeled). This yields a bipartite graph. Solving the learning of part-based models via an assignment problem is appealing, yet solution [30] is based on heuristics leaving room for improvements.

Our contribution is an extensive study of this assignment problem: We present a well-founded formulation of the problem and propose different solutions in a rigorous way. We revisit the model of [30] and introduce an alternative constraint of one-to-many assignment, where a part may be assigned to more than one regions in each image. We cast part learning as a quadratic assignment problem, and study a number of convex relaxations and optimization algorithms. These methods are evaluated and compared on two different public datasets and we demonstrate that our methodology remains complementary to the powerful visual representations obtained by state of the art deep learning approaches.

The paper is organized as follows: Section 2 gives the related works, Section 3 presents our new formulation. Then, Section 4 discusses convex relaxations, while Section 5 introduces several optimization algorithms. Finally, Section 6 is devoted to the experimental validation.

2. Previous work

Image classification has received a lot of attention during the last decades. The literature used to focus on models based on aggregated features [6, 25] or the Spatial Pyramid Matching [16]. This was before the Convolutional Network revolution [15] at the heart of the recent methods [31].

Several authors have investigated part-based models in which some parts of the image are combined in order to
3.1. Notation

Column vector \( \text{vec}(X) \) contains all elements of matrix \( X \) in column-wise order. Given matrices \( X, Y \) of the same size, \( \langle X, Y \rangle = \sum_{i,j} X_{ij}Y_{ij} \) is their (Frobenius) inner product, \( \|X\| \) and \( \|X\|_F = \sqrt{\langle X, X \rangle} \) are the spectral and Frobenius norms. Vector \( x_i^\top \) (resp. \( j \)-th column) of matrix \( X \). Vector \( 1_p \) (matrix \( 1_{m \times n} \)) is an \( n \times 1 \) vector (resp. \( m \times n \) matrix) of ones. The dot product between vectors \( x \) and \( y \) is also denoted by \( \langle x, y \rangle \). \( 1_A \) is the indicator function of set \( A \) and \( \text{Proj}_A \) is the Euclidean projector onto \( A \).

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation

3.1. Notation
For a given class, we are looking for the optimal matching matrix \( M^* \in \arg \max_{M \in \mathcal{A}_r} J(M) \) with
\[
J(M) \triangleq \sum_{p \in P, r \in R^+} m_{pr}(w_p(M), x_r). \tag{3}
\]

For a given class, we define \( W(M) \) as the \( d \times P \) matrix whose columns are \( w_p(M) \) for all parts \( p \in P \), and the similarity matrix \( C(M) \triangleq W(M)^T X^+ \). This matrix stores the \( P \times R^+ \) similarities between parts and regions. In the end, we can compactly rewrite the objective function as
\[
J(M) = \langle M, C(M) \rangle.
\]

In this work, we further deviate from the original requirement (ii) of [30] by observing that each part may not only be present once in every positive image, but with more than one instance allowed. It is expected, for instance, to find more than one chair in an office scene. The case of overlapping regions with similar descriptors is also common. With this modification, the first two requirements define the subset of \( \mathbb{R}^{P \times R^+} \)
\[
\mathcal{M}_\kappa \triangleq \left\{ M^T 1_P \leq 1_{R^+} \right\} \cap \left\{ 1_P \leq M 1_{|R^+|} \leq \kappa 1_P, \forall I \in \mathcal{I}^+ \right\}. \tag{4}
\]
Observe that \( \mathcal{M}_1 \) defined in (4) is a special case for \( \kappa = 1 \), representing a one-to-one assignment. On the other hand, for \( \kappa > 1 \), \( \mathcal{M}_\kappa \) represents a one-to-many assignment between parts and regions. This enables assigning a part to up to \( \kappa \) regions, provided that \( \kappa \leq K \triangleq |R^+| / P \). We assume \( |R^+| \) is a multiple of \( P \). The admissible set of \( M \) becomes \( \mathcal{A}_\kappa \triangleq \{0, 1\}^{P \times R^+} \cap \mathcal{M}_\kappa \).

3.3. Recasting as a quadratic assignment problem

Paper [30] solves the problem by alternatively resorting to (2) and (3). Here, we rewrite the objective function \( J \) as a function of \( M \) only by injecting an explicit expression of \( W(M) \). This gives birth to a quadratic assignment problem, which allows a number of alternative algorithms as detailed in the next section. According to LDA (2), \( W(M) = \Sigma^{-1} \left( \frac{1}{P} X^+ M^T - \mu 1^T \right) \), which in turn gives
\[
C(M) = MA - B, \tag{5}
\]
where \( R^+ \times R^+ \) matrix \( A = \frac{1}{P} X^+ \Sigma^{-1} X^+ \) is symmetric and positive definite and \( P \times R^+ \) matrix \( B = 1_P \mu^T \Sigma^{-1} X^+ \) has identical rows (rank 1). Our problem becomes equivalent to finding \( M^* \in \arg \min_{M \in \mathcal{A}_r} J_0(M) \)
\[
J_0(M) \triangleq \langle M, B - MA \rangle - \frac{1}{\beta} H(M) \tag{6}
\]
where \( H(M) = -\langle \log(M), M \rangle \) is the entropy of matrix \( M \), and \( \beta > 0 \) is the regularization parameter.

In the simpler case where \( C(M) \) is a fixed cost matrix, the minimization over \( \mathcal{S}_1 \) becomes tractable and is referred to as soft assignment. This problem has gained a lot attention because it can be solved efficiently at large scales [33]. However, a major limitation is the loss of sparsity of the solution. We describe in the section 5.2 how the authors of [30] have circumvented this problem by iterative soft assignment (ISA), also without assuming \( C \) fixed.

Dedicated methods are also common in the case of fixed \( C \). The Hungarian algorithm examined in section 5.1 gives an exact solution to the linear (hard) assignment problem, without the relaxation of the binary constraint.
4.2. Quadratic regularization

We consider now the quadratic regularization of the problem, see Figure 1 for an illustration:

\[ J_\rho(M) \triangleq \langle M, B - MA \rangle + \rho \|M\|_F^2 \]  
\[ = J_0(M) + \rho P n^+, \]  

where (9) holds provided \( M \in \mathcal{A}_1 \). In this case, \( J_\rho(M) \) and \( J_0(M) \) differ by a constant. Therefore, the minimizers of \( J_\rho \) on \( \mathcal{A}_\kappa \) are the minimizers of \( J_0 \), for any value of \( \rho \). Indeed, if \( \rho \) is sufficiently large (\( \rho > \|A\| \)), \( J_\rho \) becomes convex (see Fig. 1). In general however, \( J_\rho(M) \neq J_0(M) + \rho P n^+ \) when \( M \in \mathcal{S}_\kappa \setminus \mathcal{A}_\kappa \). We may find different solutions as illustrated in Fig. 1.

Over-relaxing the problem for the sake of convexity is not interesting as it promotes parts described by many regions instead of a few ones. Indeed, when \( \rho > \|A\| \), the minimum of \( J_\rho \) is achieved for the rank-1 matrix \( \frac{1}{2} B (A - \rho I_{R^+})^{-1} \), which may lie inside \( S \). Conversely, when \( \rho \) is negative, the regularization term acts as a force towards the set \( \mathcal{A}_\kappa \), driving the solution to a binary matrix which may be an interesting way to avoid the aforementioned problem of over relaxing the constraints.

5. Optimization

The previous section formalizes the part learning task as an optimization problem. This section now presents two alternatives to numerically solve them: (i) hard assignment optimization directly finding \( M^* \in \mathcal{A}_\kappa \), (ii) soft assignment optimization (Sect. 4.1 and 4.2) finding \( M^* \in \mathcal{S}_\kappa \). This latter strategy is not solving the initial problem. However, as already observed in [30] and [19] for classification, soft-assignment may provide good performance. This observation deserves an experimental investigation in our context.

5.1. Hungarian methods

Hungarian Algorithm (Hun): When the cost matrix \( C(M) \) is fixed and consists of \( n^2 \) square blocks (i.e. \( P = |\mathcal{R}| \)), the minimization of \( J_0(M) \) onto \( \mathcal{A}_1 \) is a linear program which solves a bipartite graph matching. Several dedicated methods give an exact solution, including the well-known Hungarian algorithm with \( O(P^3) \) complexity [3]. Starting from an initial guess \( M_0 \) (see Sect. 6.2), this solution can be seen as computing the orthogonal projection of matrix \( C(M_0) \) onto \( \mathcal{A}_1 \)

\[ M^*_{\text{hun}} \triangleq \text{Proj}_{\mathcal{A}_1}(C(M_0)) = \arg\max_{M \in \mathcal{A}_1} \langle M, C(M) \rangle. \]  

In our setting, \( M \) is not square as we consider partial assignments between \( P \) rows and \( |\mathcal{R}| > P \) columns per image. In the case of one-to-one assignment \( \kappa = 1 \), a simple trick is to add an extra row which sums to \( |\mathcal{R}| - P \) and to define a maximal cost value when affecting columns to it [1].

To achieve one-to-many assignment for \( 1 < \kappa \leq K \), we still add rows but define cost as follows. Since the number of columns \( |\mathcal{R}| \) per image is a multiple of the number of rows \( P \), we define an \( |\mathcal{R}| \times R^+ \) cost matrix consisting of \( \kappa \) blocks equal to \( C(M_0) \) stacked vertically, while the extra row that sums to \( (K - \kappa)P \) has a constant maximal cost value. Observe that this does not solve the problem onto \( \mathcal{A}_\kappa \). Rather, constraint \( 1_P \leq M_1|\mathcal{R}| \leq \kappa 1_P \) in (1) is replaced by \( M_1|\mathcal{R}| = \kappa 1_P \). Hence its solution is sub-optimal. We refer to this approach as Hun\({}^\kappa\).

We use the fast Hungarian algorithm variant of [1]. The experimental section shows that this method gives surprisingly good results in comparison to more sophisticated methods.

Integer Projected Fixed Point (IPFP): The IPFP method [17] can be seen as the iteration of the previous method, alternating between updates of the similarity matrix \( C(M) \) and projections onto the constraints set \( \mathcal{A}_1 \). More precisely, a first order Taylor approximation of the objective function is maximized (e.g. by the Hungarian algorithm) and combined with a linesearch (see Algorithm 1). This approach guarantees the convergence to a local minimizer of \( J(M) \) on the set \( \mathcal{A}_1 \).

Algorithm 1 IPFP algorithm

\begin{verbatim}
Init: M_0, set: k ← 0, M_{-1} ← Ø
while M_{k+1} ≠ M_k do
    k ← k + 1
    G_k ← 2M_kA - B  (gradient \nabla J(M_k))
    P_{k+1} ← Proj_{\mathcal{A}_1}(G_k)
    \Delta_{k+1} ← P_{k+1} - M_k
    c_k ← \langle G_k, \Delta_{k+1} \rangle
    d_k ← \langle \Delta_{k+1}, \Delta_{k+1} \rangle
    t_k ← \min(-\frac{c_k}{2d_k}, 1) if d_k < 0 and t_k = 1 otherwise
    M_{k+1} ← t_kP_{k+1} + (1 - t_k)M_k (linesearch)
end while
Output: P_k
\end{verbatim}

We observed that IPFP converges very fast nevertheless results are not improving. This is explained by the specific structure of our problem where the quadratic matrix \( Q \) of (6) is sparse and negative definite.

5.2. Iterative Soft-assignment (ISA)

The strategy of [30] referred to as Iterative Soft-Assign (ISA) solves a sequence of approximated linear assignment problems. It is based on the rationale: if we better detect regions matching a part, we will better learn that part; if we better learn a part, we will better detect region matching that part. Hence, the approach iteratively assigns regions to parts.
The optimization problem can then be written as
\begin{align*}
\min & \quad J\left(\mathbf{M}\right) \\
\text{subject to} & \quad \mathbf{C}(\mathbf{M}) \text{ and } \mathbf{M} \geq 0
\end{align*}
where functions \( G_1, G_2, G_3 \) respectively encode constraints on parts, regions and non-negativity:
\[
\begin{align*}
G_1(\mathbf{M}) &= \sum_{\mathbf{p} \in \mathcal{P}} \mathbb{1}\{m_{\mathbf{p},\mathbf{r}} \in \mathbb{R}_+\} \\
G_2(\mathbf{M}) &= \sum_{\mathbf{r} \in \mathcal{R}} \mathbb{1}\{m_{\mathbf{r},\mathbf{r}} \in \mathbb{R}_+\} \\
G_3(\mathbf{M}) &= \sum_{\mathbf{p} \in \mathcal{P}, \mathbf{r} \in \mathcal{R}} \mathbb{1}\{m_{\mathbf{p},\mathbf{r}} \geq 0\}
\end{align*}

The Generalized Forward Backward (GFB) algorithm [27], described in Alg. 3, alternates between explicit gradient descent on the primal problem and implicit gradient ascent on the dual problem. It offers theoretical convergence guarantees in the convex case and can benefit from parallelization.

\begin{algorithm}
\begin{algorithmic}
\STATE Init: \( M = M_0 \)
\WHILE {not converge}
\STATE \( G \leftarrow \nabla J_p(M) = 2M \mathbf{A}_p + B \) (gradient)
\STATE update \( M^1 \): \( \forall \mathbf{p} \in \mathcal{P} \)
\STATE \( m^1_{\mathbf{p},\mathbf{r}} \leftarrow m^1_{\mathbf{p},\mathbf{r}} + \tau \left( \text{Proj}_{\mathbb{P}_+} \left( 2m_{\mathbf{p},\mathbf{r}} - m^1_{\mathbf{p},\mathbf{r}} - \frac{1}{\lambda} G_{\mathbf{p},\mathbf{r}} \right) - m_{\mathbf{p},\mathbf{r}} \right) \)
\STATE update \( M^2 \): \( \forall \mathbf{r} \in \mathbb{R}_+ \)
\STATE \( m^2_{\mathbf{r},\mathbf{r}} \leftarrow m^2_{\mathbf{r},\mathbf{r}} + \tau \left( \text{Proj}_{\mathbb{P} \leq 1} \left( 2m_{\mathbf{r},\mathbf{r}} - m^2_{\mathbf{r},\mathbf{r}} - \frac{1}{\lambda} G_{\mathbf{r},\mathbf{r}} \right) - m_{\mathbf{r},\mathbf{r}} \right) \)
\STATE update \( M^3 \): \( \forall \mathbf{p} \in \mathcal{P}, \mathbf{r} \in \mathbb{R}_+ \)
\STATE \( m^3_{\mathbf{p},\mathbf{r}} \leftarrow m^3_{\mathbf{p},\mathbf{r}} + \tau \left( \text{Proj}_{\mathbb{P}_+} \left( 2m_{\mathbf{p},\mathbf{r}} - m^3_{\mathbf{p},\mathbf{r}} - \frac{1}{\lambda} G_{\mathbf{p},\mathbf{r}} \right) - m_{\mathbf{p},\mathbf{r}} \right) \)
\STATE update \( M \leftarrow \frac{1}{\lambda}(M^1 + M^2 + M^3) \)
\ENDWHILE
\end{algorithmic}
\end{algorithm}

The positive parameters \( \tau \) and \( L \) control the gradient descent step. Experimentally, we set \( \tau = \frac{1}{\lambda} \) for \( \kappa = 1 \), \( \tau = \frac{1}{2} \) for \( \kappa = 10 \), and \( L = \frac{1}{\lambda} \| \mathbf{A}_p \| \), estimating \( \| \mathbf{A}_p \| \) using power-iteration. Note that other splitting schemes are possible and have been tested but this combination was particularly efficient (faster convergence) due to the simplicity of the projectors onto \( \mathbb{P}_{\leq 1} \) and \( \mathbb{P}_\kappa \) that can be computed in linear time [5] using the projection onto \( \kappa \)-simplex.

6. Experiments

6.1. Datasets

The Willow actions dataset [7] is a dataset for action classification, which contains 911 images split into 7 classes of common human actions, namely interacting with a computer, photographing, playing music, riding cycle, riding horse, running, walking. There are at least 108 images per action, with around 60 images used as training and the rest as testing images. The dataset also offers bounding boxes, but we do not use them as we want to detect the relevant parts of images automatically.

The MIT 67 scenes dataset [26] is an indoor scene classification dataset, composed of 67 categories. These include stores (e.g. bakery, toy store), home (e.g. kitchen, bedroom), public spaces (e.g. library, subway), leisure (e.g. restaurant, concert hall), and work (e.g. hospital, TV studio). Scenes may be characterized by their global layout (corridor), or by the objects they contain (bookshop). Each category has around 80 images for training and 20 for testing.
6.2. Description and classification pipeline

We essentially follow the learning and classification setup of [30]. During part learning, \(|\mathcal{R}| = 1,000\) regions are extracted from each training image and used to learn the parts. During encoding, \(|\mathcal{R}|\) regions are extracted from both training and test images, and all images are encoded based on the learned parts. Finally, a linear SVM classifies the test images. For each stage, we briefly describe the choices made in [30] and discuss our improvements.

Extraction of image regions Two strategies are investigated:

- **Random regions** (‘R’). As in [30], \(|\mathcal{R}|\) regions are randomly sampled over the entire image. The position and scale of these regions are chosen uniformly at random, but regions are constrained to be square and have a size of at least 5% of the image size.

- **Region proposals** (‘P’). Following [22], up to \(|\mathcal{R}|\) regions are obtained based on selective search [39]. If less than \(|\mathcal{R}|\) regions are found, random regions complete the set.

Region descriptors Again two strategies are investigated:

- **Fully connected** (‘FC’). As in [30], we use the output of the 7th layer of the CNN on the rescaled regions, resulting in 4,096-dimensional vectors. For the Willow dataset, we use the standard Caffe CNN architecture [13] trained on ImageNet. For MIT67, we use the hybrid network [43] trained on ImageNet and on the Places dataset. The descriptors are square-rooted and \(\ell_2\)-normalized. We note that each region was cropped and fed to the network in [30].

- **Convolutional** (‘C’). As an improvement, we use the last convolutional layer, after ReLU and max pooling, of the very deep VGG-VD19 CNN [31] trained on ImageNet. To obtain a region descriptor, we employ average pooling over the region followed by \(\ell_2\)-normalization, resulting in a 512-dimensional vector. Contrary to ‘FC’, we do not need to rescale every region. The entire image is fed to the network only once, as in [11, 10]. Further, following [37], pooling is carried out by an integral histogram. These two options enable orders of magnitude faster description extraction compared to ‘FC’. To ensure the feature map is large enough to sample \(|\mathcal{R}|\) regions despite loss of resolution (by a factor of 32 in the case of VD-19), images are initially resized such that their maximum dimension is 768 pixels. This was shown to be beneficial in [42].

Initialization The initialization step follows [30]. All training positive regions are clustered and for each cluster an LDA classifier is computed over all regions of the cluster. The maximum responses to the classifiers are then selected per image. Two scores are then computed: the average of the maximum responses over positive and negative sets. The ratio of these scores is used to select the top \(P\) clusters to build the initial part classifiers. Finally, an initial matching matrix \(M\) is built by softmax on classifier responses.

Encoding Given an image, belonging to training or testing set, each learned part classifier is applied to every region descriptor to generate a global image descriptor. We use several alternatives:

1. **Bag-of-Parts** (‘BoP’) [30]: For each part, the maximum and average classifier scores are computed over all regions. These data are then concatenated for all parts.

2. **Spatial Bag-of-Parts** (‘SBoP’): In this paper, we also introduce SBoP, which adds weak spatial information to BoP by using Spatial Pyramids as [8]: Maximum scores are computed over the four cells of a \(2 \times 2\) grid over the image and appended to the original BoP.

3. **CNN-on-Parts** (‘CoP’) [30]: The CNN descriptors corresponding to the maximum scoring region per part are concatenated to form the global image descriptor.

4. **PCA on CNN-on-Parts** (‘PCoP’): This paper also investigates PCoP, where centering and PCA are applied to CoP.

The global image descriptors will be the input of the final SVM classifier.

Parameters of the learning algorithms For the Iterative Soft-Assign (ISA) method, we use the same parameters as [30]. Concerning the \(GFB_{\rho}^K\) method solving (11), we tested different configurations: w/ or w/o regularization (as controlled by \(\rho\)) and one-to-one or one-to-many assignment (as controlled by \(\kappa\)).

When considering 1-to-1 matching with \(GFB_{\rho}^1\), we perform 2k iterations of the projection, except for the MIT67 dataset with convolutional descriptor, where iterations are limited to 1k. In all experiments performance remains stable after 1k iterations. We denote by \(GFB_{\rho}^1\) the case without regularization (\(\rho = 0\)). For the \(GFB_{\rho}^K\), we choose \(\rho = 10^{-3}\|A\|\) after experimental evaluation on the Willow dataset.

When considering the multiple assignment model \(GFB_{\rho}^K\) with \(\kappa = K = 10\), we used \(\rho = -\|A\|\). The motivation
Table 1. Baseline performance, without part learning.

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>Willow</th>
<th>MIT67</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc</td>
<td>mAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full-image</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>70.8</td>
<td>75.7</td>
</tr>
<tr>
<td></td>
<td>mAP</td>
<td>76.3</td>
<td>88.5</td>
</tr>
</tbody>
</table>

Figure 2. Top scoring parts for various images of bowling, florists, gym and wine cellar.

behind this choice is that, because the simplex $S_\kappa$ is larger as $\kappa$ increases, the optimal matrix is more likely to lie inside the simplex, with values between 0 and 1 (soft assignment). This effect can be compensated by using a negative value for $\rho$, which yields a hard assignment solution in practice in our experiments.

6.3. Results

In the following, we are showing results for (i) fully connected layer descriptor on random regions (R+FC), which follows [30], and (ii) convolutional layer descriptor on region proposals (P+C) that often yields the best performance. We evaluate different learning algorithms on BoP and CoP encoding, and then investigate the new encoding strategies SBoP and PCoP as well as combinations for ISA, Hun, and GFB$^\kappa$ algorithms. Methods are evaluated in the context of action and scene classification in still images. On Willow we always measure mean Average Precision (mAP) while on MIT67 we calculate both mAP and classification accuracy (Acc).

We start by providing, in Table 1, a baseline corresponding to the description methods ‘FC’ and ‘C’ applied on the full image without any part learning. The comparison to subsequent results with part learning reveals that part-based methods always provides improvement.

We now focus on the part learning methods. Figure 2 shows some qualitative results of learned parts on MIT67.

Table 2. Performance of ISA and all methods satisfying $M_1$ on Willow and MIT67.

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>ISA</th>
<th>IPFP</th>
<th>Hun</th>
<th>GFB$^0$</th>
<th>GFB$^\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>R+FC BoP</td>
<td>mAP</td>
<td>76.6</td>
<td>79.0</td>
<td>78.9</td>
<td>79.7</td>
<td>80.6</td>
</tr>
<tr>
<td>P+C BoP</td>
<td>mAP</td>
<td>89.2</td>
<td>86.3</td>
<td>88.3</td>
<td>88.2</td>
<td>87.5</td>
</tr>
<tr>
<td>P+C CoP</td>
<td>mAP</td>
<td>91.6</td>
<td>91.3</td>
<td>91.1</td>
<td>91.8</td>
<td>91.8</td>
</tr>
</tbody>
</table>

MIT67

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>ISA</th>
<th>IPFP</th>
<th>Hun</th>
<th>GFB$^0$</th>
<th>GFB$^\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>R+FC BoP</td>
<td>mAP</td>
<td>76.6</td>
<td>75.4</td>
<td>75.7</td>
<td>74.7</td>
<td>76.3</td>
</tr>
<tr>
<td>P+C BoP</td>
<td>mAP</td>
<td>75.1</td>
<td>70.7</td>
<td>72.8</td>
<td>70.9</td>
<td>70.9</td>
</tr>
<tr>
<td>P+C CoP</td>
<td>mAP</td>
<td>80.0</td>
<td>79.2</td>
<td>79.8</td>
<td>79.2</td>
<td>79.3</td>
</tr>
</tbody>
</table>

Then, Table 2 shows the performance of ISA against several methods satisfying the constraint $M_1$, i.e. a part is composed of a single region in every positive image. These methods include IPFP, Hungarian, GFB$^0$, and GFB$^\rho$.

On the Willow dataset, for the R+FC descriptor with BoP encoding, we observe that GFB$^\rho > GFB^0 >$ Hungarian and IPFP > ISA. However, on MIT67 the results are different and we have ISA > Hungarian and GFB$^0 > GFB^\rho$. Similar trends are observed when using the improved P+C descriptor with the BoP encoding. Nevertheless, note that all methods perform similarly when using the CoP encoding. After these evaluations, IPFP was not evaluated in further experiments since it performs on par with the Hungarian or worst, as explained in Section 5.1.

These results show that overall ISA outperforms other optimization methods, which satisfy $M_1$. The explanation of this difference in performance lies in the fact that ISA is stopped before convergence and does not satisfy $M_1$, as explained in Section 5.2. This result is further confirmed by running an iteration of the Hungarian algorithm on the output of ISA, see Table 3. Such experiment forces the parts resulting from ISA to satisfy $M_1$ and we observe an overall drop of performance.

Table 2 also shows that region proposals combined with convolutional layer descriptions shows a significant performance gain, especially on the Willow dataset. Therefore, the improved region descriptions and encoding are evaluated using ISA, see Table 4. We can see a consistent improvement for the SBoP and PCoP encoding. Also PCA yields more improvement for descriptors based on fully

Table 3. Performance ISA, when forced to satisfy the constraints $M_1$ with hard assignment. ISA+H refers to performing one iteration of the Hungarian algorithm on the solution obtained by ISA.

<table>
<thead>
<tr>
<th>Method</th>
<th>Measure</th>
<th>ISA</th>
<th>ISA+H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Willow R+FC BoP</td>
<td>mAP</td>
<td>76.6</td>
<td>76.9</td>
</tr>
<tr>
<td>Willow P+C BoP</td>
<td>mAP</td>
<td>89.2</td>
<td>88.1</td>
</tr>
<tr>
<td>Willow P+C CoP</td>
<td>mAP</td>
<td>91.6</td>
<td>89.6</td>
</tr>
<tr>
<td>MIT67 R+FC BoP</td>
<td>mAP</td>
<td>78.8</td>
<td>77.9</td>
</tr>
</tbody>
</table>
Table 4. Results on Willow and MIT67 datasets for the ISA method, with improved region descriptions P+C and improved encoding methods SBoP and PCoP.

<table>
<thead>
<tr>
<th>Method</th>
<th>Meas.</th>
<th>BoP mAP</th>
<th>SBoP mAP</th>
<th>CoP mAP</th>
<th>PCoP mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Willow R+FC</td>
<td>mAP</td>
<td>76.6</td>
<td>78.7</td>
<td>81.6</td>
<td>82.4</td>
</tr>
<tr>
<td>Willow P+C</td>
<td>mAP</td>
<td>89.2</td>
<td>90.1</td>
<td>91.6</td>
<td>91.7</td>
</tr>
<tr>
<td>MIT67 R+FC</td>
<td>Acc mAP</td>
<td>76.6</td>
<td>76.1</td>
<td>76.8</td>
<td>77.1</td>
</tr>
<tr>
<td>MIT67 P+C</td>
<td>Acc mAP</td>
<td>77.6</td>
<td>76.7</td>
<td>80.0</td>
<td>80.5</td>
</tr>
</tbody>
</table>

Table 5. Performance of ISA and the proposed methods satisfying the constraint $\mathcal{M}_K. \kappa$ is set to 10 for Hun$^\kappa$ and GFB$^\kappa$.

<table>
<thead>
<tr>
<th>Method</th>
<th>Meas.</th>
<th>ISA mAP</th>
<th>Hun$^\kappa$ mAP</th>
<th>GFB$^\kappa$ mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Willow P+C</td>
<td>mAP</td>
<td>89.2</td>
<td>89.6</td>
<td>89.6</td>
</tr>
<tr>
<td>P+C CoP</td>
<td>mAP</td>
<td>91.6</td>
<td>91.4</td>
<td>91.3</td>
</tr>
<tr>
<td>P+C SBoP+PCoP</td>
<td>mAP</td>
<td>91.9</td>
<td>92.1</td>
<td>92.1</td>
</tr>
<tr>
<td>MIT67</td>
<td>Acc mAP</td>
<td>75.1</td>
<td>77.7</td>
<td>77.3</td>
</tr>
<tr>
<td></td>
<td>mAP</td>
<td>76.7</td>
<td>79.2</td>
<td>79.4</td>
</tr>
<tr>
<td></td>
<td>Acc mAP</td>
<td>80.0</td>
<td>80.4</td>
<td>80.5</td>
</tr>
<tr>
<td></td>
<td>mAP</td>
<td>80.2</td>
<td>80.6</td>
<td>80.5</td>
</tr>
<tr>
<td></td>
<td>Acc mAP</td>
<td>81.4</td>
<td>81.5</td>
<td>81.5</td>
</tr>
<tr>
<td></td>
<td>mAP</td>
<td>81.2</td>
<td>81.7</td>
<td>81.9</td>
</tr>
</tbody>
</table>

Table 6. Evaluation of the influence the parameter $\kappa$ on the performance of the Hungarian algorithm on MIT67.

<table>
<thead>
<tr>
<th>Method</th>
<th>Meas.</th>
<th>Hun$^\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P+C BoP</td>
<td>Acc mAP</td>
<td>72.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>76.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>77.7</td>
</tr>
</tbody>
</table>

Table 7. Performance in terms of accuracy of existing part-based and non-part-based methods on the MIT67 dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Part-based</th>
<th>MIT67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al [42]</td>
<td>No</td>
<td>70.8</td>
</tr>
<tr>
<td>Peng et al [35]</td>
<td>Yes</td>
<td>74.9</td>
</tr>
<tr>
<td>Wang et al [40]</td>
<td>Yes</td>
<td>75.3</td>
</tr>
<tr>
<td>Mahmood et al [20]</td>
<td>No</td>
<td>75.6</td>
</tr>
<tr>
<td>Zuo et al [44]</td>
<td>Yes</td>
<td>76.2</td>
</tr>
<tr>
<td>Parizi et al [24]</td>
<td>Yes</td>
<td>77.1</td>
</tr>
<tr>
<td>Mettes et al [22]</td>
<td>Yes</td>
<td>77.4</td>
</tr>
<tr>
<td>Sicre et al [30]</td>
<td>Yes</td>
<td>78.1</td>
</tr>
<tr>
<td>Zheng et al [42]</td>
<td>No</td>
<td>78.4</td>
</tr>
<tr>
<td>Wu et al [41]</td>
<td>Yes</td>
<td>78.9</td>
</tr>
<tr>
<td>Cimpoi et al [4]</td>
<td>No</td>
<td>81.0</td>
</tr>
<tr>
<td>Herranz et al [12]</td>
<td>No</td>
<td>86.0</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>Yes</td>
<td><strong>81.5</strong></td>
</tr>
</tbody>
</table>

connected layers than on convolutional ones.

We further study the problem of part learning following the constraint $\mathcal{M}_K, \kappa$, see Eq. (4). Therefore, a part can be composed of several regions of the same image. We remind that each of these regions can be assigned to at most one part. We compare ISA to GFB$^\kappa$ and Hun$^\kappa$, which satisfy $\mathcal{M}_K$. In our setup, we have $1 \leq \kappa \leq K = 10$. Concerning Hun$^\kappa$, $\kappa$ is set to $K$. Results given on Table 5 show that GFB$^\kappa$ and Hun$^\kappa$ offer better results than ISA, especially with the BoP encoding. Moreover, there is a large improvement over the same methods satisfying $\mathcal{M}_1$ and we note that GFB adapts to the various types of constraints.

Since the Hungarian algorithm forces parts to be composed of a fixed number of regions $\kappa$, we evaluated the impact of this parameter on the classification performance on MIT 67 dataset, see Table 6. Interestingly, high values of $\kappa$ offer the best performance. Therefore, a mixture of parts combining all regions of images allows a better description. An explanation for such results can be that parts will be more diverse and therefore more distinct one to another.

Finally, our methods offer good performance competing with the state of the art on both datasets: 92.1% mAP on Willow and 81.5% accuracy on MIT67, see Table 5 and 7.

7. Conclusion

To conclude, we have investigated in this work the problem of discovering parts for part-based image classification. We have shown that this problem can be recast as a quadratic assignment problem with concave objective function to be minimized with non-convex constraints. While being known to be a very difficult problem, several techniques have been proposed in the literature, either trying to find “hard assignment” in a greedy fashion, or based on optimization of the relaxed problem, resulting in “soft assignment”. Several methods have been investigated to address this task and compared to the previous method of [30]. Of the proposed algorithms, GFB is the most adaptable and the Hungarian is the fastest. Both algorithms offer improved performance on two public datasets.

Our reformulation and investigation of different optimization methods explore the limits of the original problem defined in [30]. We introduce a number of new algorithms, which are designed to satisfy the constraint of the problem definition. We show that the explicit relaxation of the constraint on the assignment of regions to parts leads to a better part model. Such adaptation was not possible in the work of [30]. We believe this knowledge will help the community in the search for more appropriate models, potentially end-to-end trainable, using better network architectures.

We additionally proposed improvements on several stages of the classification pipeline, namely region extraction, region description and image encoding, using a very deep CNN architecture. Furthermore, the new region description method is orders of magnitude faster, as this process was previously the bottleneck in [30].
References


