Exploiting Contextual Knowledge for Hybrid Classification of Visual Objects

Thomas Eiter and Tobias Kaminski

Institut für Informationssysteme, Technische Universität Wien
Favoritenstraße 9-11, A-1040 Vienna, Austria
{eiter,kaminski}@kr.tuwien.ac.at

Abstract. We consider the problem of classifying visual objects in a scene by exploiting the semantic context. For this task, we define hybrid classifiers (HC) that combine local classifiers with context constraints, and can be applied to collective classification problems (CCPs) in general. Context constraints are represented by weighted ASP constraints using object relations. To integrate probabilistic information provided by the classifier and the context, we embed our encoding in the formalism LP^{MLN}, and show that an optimal labeling can be efficiently obtained from the corresponding LP^{MLN} program by employing an ordinary ASP solver. Moreover, we describe a methodology for constructing an HC for a CCP, and present experimental results of applying an HC for object classification in indoor and outdoor scenes, which exhibit significant improvements in terms of accuracy compared to using only a local classifier.

1 Introduction

For several decades, AI research has devoted huge efforts to automate logical reasoning in *knowledge representation and reasoning* (*KRR*) and to develop methods for statistical learning and inference in *machine learning* (*ML*). While these areas are rather mature, it became evident that many real-world domains require both logical and statistical reasoning as they comprise complex relational as well as uncertain information. Consequently, *statistical relational learning* (*SRL*) has gained momentum, and many approaches which combine statistical and logical methods have been developed (see [9] for an overview).

One of the basic tasks in SRL is *collective classification*, which is simultaneously finding correct labels for a number of interrelated objects; this has applications in many concrete domains, e.g. classification of interlinked documents, part-of-speech tagging and optical character recognition [20]. A further such application is to predict the labels (i.e., class memberships) of objects in a complex visual scene that contains many objects of different classes. Even if advanced and robust algorithms for object recognition have been developed, e.g. *SIFT* descriptors [11] and the *bag of keypoints* approach [4], they may fail unavoidably and yield ambiguous results due to few training data, noisy inputs, or inherent ambiguity of visual appearance (e.g. a lemon and a tennis ball might be indistinguishable in a low resolution image [15]). It is then still possible to draw on further information from the scene in which an object occurs to disambiguate its label. For an example, consider the street scene in Figure 1, where object 2 is wrongly labeled as 'building' in the center image. This misclassification could be resolved by considering

Fig. 1. Objects in a scene with predicted labels.

all object labels simultaneously, drawing on background knowledge that wheels normally appear at the bottom of a car; thus, the probability for labeling object 3 as 'car' increases.

In KRR, a natural approach to formalize admissible labelings of objects respecting their interrelations would consist in imposing logical constraints on labelings and using constraint programming techniques to compute 'possible worlds' represented by complete labelings. While this approach yields all consistent labelings, it neglects (hidden) features of the concrete classification problem. Hence, it is desirable to combine constraints over label assignments with the output of a probabilistic classifier processing (low-level) object features. A naive such combination is to use a ranking over all labels for each object induced by the probability distributions given by a classifier, and to compute the labeling that maximizes the rank of the assigned labels while satisfying all constraints. However, in real-world domains this approach turns out to be too restrictive. First, real data necessarily has exceptions that cannot all be modeled, which may prevent that a consistent solution is found; second, this approach retains no information about the metric distance between label probabilities, which is essential for deciding whether a label should be changed to a less likely one in order to satisfy some constraint.

In this paper, we bridge the gap between combinatorial and probabilistic object classification by encoding the context of a concrete *collective classification problem* (*CCP*) in a set of *answer set programming* (*ASP*) rules and constraints that we assign a probabilistic semantics. Using ASP to formalize context knowledge, we can combine multiple context relations in even complex constraints and utilize *closed world reasoning* to express e.g. that objects not containing car parts should not be labeled as cars.

Our main contributions are briefly summarized as follows.

(1) We define a general framework for solving CCPs that combines a generic local classifier and context constraints into a *hybrid classifier*, which is given semantics via an embedding into LP^{MLN} [10]. We then show how solutions can be obtained efficiently with a backtranslation from LP^{MLN} into classical answer set programs with weak constraints [2], and by leveraging combinatorial optimization capabilities of state-of-the-art ASP solvers.

(2) We describe a methodology for constructing a hybrid classifier for a specific domain by designing and tuning a context encoding. To the best of our knowledge, this has not been considered before.

(3) We examine the usefulness of our methodology with an extensive empirical evaluation in the domain of visual object classification in indoor as well as outdoor scenes. The results provide evidence that hybrid classifiers can significantly improve accuracy, provided that the local classifier works reasonably well, given the outset of few training

data, noisy data or ambigous data. Furthermore, they show that tuning and the use of a validation set are important elements for increasing accuracy gains.

Notably, in our approach knowledge representation and reasoning is a first-class citizen, while SRL approaches often rely on statistical formulations and probabilistic solving methods; this seems less geared towards combinatorial problem solving. Moreover, our encoding can be easily extended by spatial reasoning via rules over extracted facts, as well as by a component for taxonomical reasoning over label categories.

2 Preliminaries

Answer set programs with weak constraints [2] constitute the host language of our hybrid classification encoding. A *normal logic program* P is a finite set of rules of the form

$$H \leftarrow B_1, ..., B_k, \mathbf{not}\, B_{k+1}, ..., \mathbf{not}\, B_m, \tag{1}$$

where H and all B_i are function-free first-order atoms from a classical first-order signature. Given a rule r, H is called the head of r, $B^+(r) = \{B_1, ..., B_k\}$ its *positive body* and $B^-(r) = \{B_{k+1}, ..., B_m\}$ its *negative body*. A rule without a head is called a *constraint*. The grounding P_g of P is obtained by replacing all variables by constants occurring in P in all possible ways, as usual. Stable models (or answer sets) are the minimal models of the GL-reduct (cf. [8]).

A *weak constraint* is written as follows:

$$:\sim B_1, ..., B_k, \mathbf{not}\, B_{k+1}, ..., \mathbf{not}\, B_m\, [w]. \tag{2}$$

The weight w of a weak constraint c, denoted $weight(c)$, is either an integer constant or a variable occurring in the positive body of the constraint. A ground weak constraint has the same weight as the constraint it originates from. For a Herbrand interpretation I and a set C of weak constraints, the violation cost of C wrt. I is $cost_I(C) = \sum_{c' \in C'} weight(c')$, where $C' \subseteq C$ are the weak constraints such that $B^+ \subseteq I$ and $B^- \cap I = \emptyset$. The answer sets of $P \cup C$ are all those answer sets I of P such that no answer set I' of P with $cost_{I'}(C) < cost_I(C)$ exists.

We assign a probabilistic semantics to our encoding by utilizing the formalism LP^{MLN} [10], which employs weighted rules for combining ASP with probabilistic graphical models based on *Markov logic networks* (*MLNs*) [17]. LP^{MLN} programs generalize normal logic programs by assigning a weight w to every rule r of form (1) in the program. The weight w is either a real number or α, representing the infinite weight. When grounding an LP^{MLN} program, every ground weighted rule $w : r_g$ is mapped to the same weight w as its non-ground counterpart $w : r$. A probabilistic semantics is defined for LP^{MLN} programs as follows.

Definition 1 (Unnormalized weight [10]). *For an LP^{MLN} program Π and a Herbrand interpretation I, the unnormalized weight of I under Π is given by*

$$W_\Pi(I) = \begin{cases} exp\Big(\sum\limits_{w:r \in \Pi_I} w\Big) & \text{if } I \in SM[\Pi], \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where Π_I represents all weighted rules $w : r$ in Π s.t. $I \models r$, and $SM[\Pi]$ contains all I s.t. I is a classical answer set of Π_I, omitting the weights.

In order to obtain a probability distribution over all Herbrand interpretations wrt. an LP^{MLN} program, the corresponding weights have to be normalized.

Definition 2 (Normalized weight [10]). *For an LP^{MLN} program Π and a Herbrand interpretation I, the normalized weight of I under Π is given by*

$$P_\Pi(I) = \lim_{\alpha \to \infty} \frac{W_\Pi(I)}{\sum_{J \in SM[\Pi]} W_\Pi(J)}. \tag{4}$$

Lee and Wang [10] define a (probabilistic) stable model of an LP^{MLN} program Π to be a Herbrand interpretation I s.t. $P_\Pi(I) \neq 0$.

Since our goal is to use LP^{MLN} programs for finding the global best labeling for a set of objects, i.e. the answer set encoding the label assignment with the highest probability, we do not discuss conditional probability queries here. Lee and Wang show a close relationship between ASP with weak constraints and LP^{MLN} programs, such that under certain conditions the answer set with the highest normalized probability can be computed directly by an ordinary answer set solver that exhibits optimization capabilities. The authors define a translation $\tau(P) = \Pi$ from an answer set program with weak constraints P to an LP^{MLN} program Π, and show the following correspondence.

Proposition 1 (adapted from [10]). *For an answer set program with weak constraints P that has an answer set, its answer sets are the Herbrand interpretations $\{I \mid \nexists I' : P_\Pi(I') > P_\Pi(I)\}$, where $\Pi = \tau(P)$.*

For translating an LP^{MLN} program into an answer set program with weak constraints, we apply τ^{-1}, which is only applicable to LP^{MLN} programs in which all rules are assigned the infinite weight α and only constraints of the form (2) are assigned arbitrary weights. The translation τ^{-1} works by omitting the weight of rules with non-empty head and by replacing constraints of the form $w : \leftarrow B_1, ..., B_k, \mathbf{not}\ B_{k+1}, ..., \mathbf{not}\ B_m$ by a rule $H \leftarrow B_1, ..., B_k, \mathbf{not}\ B_{k+1}, ..., \mathbf{not}\ B_m$ together with the weak constraint $:\sim \mathbf{not}\ H\ [-w]$, where H is a fresh atom not occurring elsewhere.[1] Under the mentioned restrictions, Proposition 1 still holds.

3 Hybrid Classification

We aim at applying LP^{MLN} programs for simultaneously classifying all objects in a visual scene. In order to obtain a complete labeling that is as close as possible to the ground truth, we exploit two sources of probabilistic information regarding the most likely label for a given object. On the one hand, we use a *classifier* that is trained on vectors of object features and predicts the probability of each local label given the features of a single new object. On the other hand, we exploit the *relational context*

[1] As the logic program rules here are more restricted than in [10], we adapt the translation defined there. Real-valued weights can be approximated by integers in weak constraints.

defined by relations between several objects, by learning the probability of certain label combinations for sets of objects that are related in specific ways, e.g. some objects in an image may be contained in some other objects more or less frequently. In this way, the notion of the best label for some object is probabilistically constrained from two sides, and we strive for an optimal label on the basis of all probabilistic information available. We refer to our combination of local classifier and relational context as *hybrid classifier* (*HC*); notably, their classification results increasingly disagree with context elaboration, and the relational component has a richer structure than in most related approaches on collective classification.

In this section, we define an HC in form of an LP^{MLN} encoding that combines a local classifier with a set of weighted context constraints over label assignments wrt. the relational structure. We start by defining *collective classification problems* (*CCPs*) based on the definition in [20], but we generalize the neighborhood function used there to arbitrary relations between objects. First, we introduce a schema on the basis of which a group of CCPs can be defined.

Definition 3 (CCS). *A* collective classification schema *(*CCS*) is represented by a pair* $S = \langle L, R \rangle$ *consisting of*

- *a set* $L = \{l_1, ..., l_m\}$ *of possible object labels,*
- *a family* $R = \{R_0, ..., R_k\}$ *of sets of* 0*- to* k*-ary* context relation names.

The sets $R_i \in R$ contain names for i-ary relations between objects that can, for instance, be extracted from a scene image, e.g. a binary relation entailing all pairs of objects where the first object is contained in the second object, or a ternary relation stating that an object is located in-between two other objects. The set L contains possible object labels, e.g. 'car' and 'tree' for objects in a street scene.

A CCS is instantiated by a CCP, by fixing the set of objects that need to be classified together with their local object features as well as the concrete relations occurring between them, as follows.

Definition 4 (CCP). *A* collective classification problem *(*CCP*) is represented by a triple* $\mathcal{C} = \langle S, O, e \rangle$ *consisting of*

- *a CCS* $S = \langle L, R \rangle$*,*
- *a set* $O = \{o_1, ..., o_n\}$ *of objects with associated features* $f(o_i)$ *for each* $o_i \in O$*,*
- *a function* $e : \bigcup_{R_i \in R} \rightarrow O^k$ *that maps each* i*-ary relation name to a concrete* i*-ary relation over objects.*

A solution for a CCP is a complete labeling *represented by a mapping* $\lambda : O \rightarrow L$*, assigning a label from* L *to each object in* O*.*

Next, we introduce *local classifiers*, where we abstract from the level of particular object features and assume a classifier that is able to return a probability distribution over all labels for each object by processing their corresponding features. Subsequently, we draw on the information provided by local classifier c for hybrid classification by integrating c with a context encoding into an HC.

Definition 5 (Local classifier). *Given a CCS* $S = \langle L, R \rangle$*, a* local classifier c *is a function that maps the feature vector* $f(o)$ *of an object* o *to a discrete probability distribution* P_o^c *over all labels in* L *(on the basis of their associated feature vectors).*

Due to the generality of the approach, different kinds of classifiers, e.g. *Logistic Regression* or *Neural Networks*, can be utilized to instantiate the local classifier c.

Example 1. For the scene in Figure 1, we construct a corresponding CCP $\mathcal{C} = \langle S, O, e\rangle$ with $S = \langle\{car, building, wheel\}, \{\emptyset, \emptyset, \{contains\}\}\rangle$, $O = \{o_1, o_2\}$ (omitting 'object 3') and $e(contains) = \{\langle o1, o2\rangle\}$. We assume that the classifier c for $\mathcal{C}$ yields, based on the object features $f(o_i)$ extracted from the image, $P^c_{o_1}(car) = 0.4$, $P^c_{o_1}(building) = 0.5$, $P^c_{o_1}(wheel) = 0.1$, $P^c_{o_2}(car) = 0.1$, $P^c_{o_2}(building) = 0.1$ and $P^c_{o_2}(wheel) = 0.8$.

In other approaches [6, 15, 1], relations between objects are often used to conditionalize the probability distribution of label combinations of the involved objects. As we use ASP constraints to describe the relational context, we use the relations in the sets R_i differently, i.e. to state restrictions over expected label assignments via relations that may be derived from further relations together with other supposed label assignments.

Following Richardson and Domingos [17], the weight of a context constraint in LP^{MLN} can be interpreted as the logarithm of the odds between a possible world where it is satisfied and one where it is not (called the *log odds*), other things being equal. In general, context constraints are not independent from each other, thus changing their truth value also changes the value of other constraints. However, as we consider cases with only few training data (such that the classifier output can still be improved by considering the context), it is unfeasible to learn all interactions between constraints from it. Thus, we assume *bona fide* independence of context constraints and straight use the *log odds* for the constraints calculated from the training instances as weights.

The restrictions over label assignments in terms of the relational context are formalized by a *context encoding* as follows:

Definition 6 (Context encoding). *Given a CCS $S = \langle L, R\rangle$, we use the following designated predicates:* context relation predicates $\mathcal{R}$=$\bigcup_{R_i \in R} R_i$ *and* helper predicates $\mathcal{H}$ *ranging over tuples of objects, and the* label assignment predicate a_label *ranging over pairs of objects and labels. A* context encoding E *for S is an LP^{MLN} program that consists of rules of the form*

$$\alpha : h(\overline{X}) \leftarrow b_1(\overline{X}), \ldots, b_k(\overline{X}), \mathbf{not}\, b_{k+1}(\overline{X}), \ldots, \mathbf{not}\, b_m(\overline{X}), \tag{5}$$

where $h \in \mathcal{H}$ and $b_1, \ldots, b_m \in \mathcal{R} \cup \mathcal{H} \cup \{l\_assign\}$; and constraints of the form

$$w : \leftarrow b_1(\overline{X}), \ldots, b_k(\overline{X}), \mathbf{not}\, b_{k+1}(\overline{X}), \ldots, \mathbf{not}\, b_m(\overline{X}), \tag{6}$$

where $b_1, \ldots, b_m \in \mathcal{R} \cup \mathcal{H} \cup \{l\_assign\}$, and w is the log odds for the constraint being satisfied given the extensions of the predicates in $\mathcal{R} \cup \mathcal{H}$ (as learned from training data).

The helper predicates in $\mathcal{H}$ are used to recursively aggregate relations and label assignments into new relations, which can be utilized to restrict permissible assignments.

Example 2 (cont'd). We define a simple context encoding E for S from Example 1, using the context relation predicate $contains$ and the helper predicate has_car_part:

$$\alpha : has\_car\_part(X) \leftarrow contains(X, Y), a\_label(Y, wheel) \tag{7}$$

$$1.95 : \leftarrow \mathbf{not}\, a\_label(X, car), has\_car\_part(X) \tag{8}$$

The particular weight is chosen for the context constraint because we assume here that we have observed 28 cases in our training data where an object that has a car part is actually a car, and four cases where it is not, i.e. the log odds for the constraint being true given the extension of the predicate has_car_part are $ln(28/4) \approx 1.95$.

Taxonomic reasoning can easily be added by introducing, e.g. a rule that derives all labels representing car parts. Likewise, spatial reasoning can be implemented by inferring further relations from the given relations (e.g., an object overlaps which another object if one contains the other).

We combine a local classifier for a CCS $S = \langle L, R\rangle$ and a context encoding for S into an HC that yields a solution for a CCP $\mathcal{C} = \langle S, O, e\rangle$ as follows:

Definition 7 (Hybrid classifier encoding). *Given a CCP $\mathcal{C} = \langle S, O, e\rangle$ for a CCS $S = \langle L, R\rangle$, a local classifier c, and a context encoding E for S, the* hybrid classifier (HC) *for $\mathcal{C}$ is represented by an LP^{MLN} program $\Pi_{\mathcal{C}}(c, E) = E \cup A(c, O) \cup I(\mathcal{C})$ where the classifier assignment encoding $A(c, O)$ contains*

(1) the weighted facts

$\alpha : label(l_i)$ *for each label* $l_i \in L$,

$\alpha : clf(o_i, l_j, p)$ *for each* $o_i \in O$ *and* $l_j \in L$, *where* $p = ln\left(\frac{P^c_{o_i}(l_j)}{1-P^c_{o_i}(l_j)}\right)$,

(2) the two guessing rules

$\alpha : a\_label(O, L) \leftarrow object(O), label(L), not\ n\_a\_label(O, L)$,

$\alpha : n\_a\_label(O, L) \leftarrow object(O), label(L), not\ a\_label(O, L)$,

(3) the unique assignment constraint

$\alpha : \leftarrow \#count\{L : a\_label\_prob(X, L, P)\} \neq 1, object(X)$,

(4) the weighted classifier constraint

$P : \leftarrow not\ a\_label\_prob(O, L, P), clf(O, L, P)$

and the rule

$\alpha : a\_label\_prob(O, L, P) \leftarrow a\_label(O, L), clf(O, L, P)$,

and the CCP instance encoding $I(\mathcal{C})$ contains

(5) the weighted facts

$\alpha : object(o_i)$ *for each object* $o_i \in O$, *and*

$\alpha : r_i(o_1, \ldots, o_i)$ *for each* $r_i \in R_i$ *and each* $\langle o_1, \ldots, o_i\rangle \in e(r_i)$.

Here, (5) represents the input part of the HCP, while (2)–(4) are fixed; (2) and (3) ensure that each object gets exactly one label, and (4) assigns the weights by the local classifier to the separate label assignments. Again, we use the log odds between a complete label assignment where a label is assigned vs. not assigned from the local classifier as weight.

Intuitively, a solution of an HCP should minimize the violation costs of context constraints, but at the same time maximize the joint classifier probability of the label assignment. As the two optimization criteria are opposite in general, the goal is a good compromise that yields a better label assignment than the one of the local classifier. As it is not clear a priori how much influence the classifier and the context constraints should have on a solution, the probabilities returned by the local classifier in (4) could be scaled by an influence factor such that its impact on a solution can be varied for tuning an HC.

A solution for a CCP wrt. an HC is defined as follows.

Definition 8 (HC solution). *A solution for an CCP $\mathcal{C}$ provided by an HC $\Pi_{\mathcal{C}}(c, E)$ is a solution λ for $\mathcal{C}$ s.t. for some Herbrand interpretation I, no Herbrand interpretation I' with $P_{\Pi_{\mathcal{C}}(c,E)}(I') > P_{\Pi_{\mathcal{C}}(c,E)}(I)$ exists and $a\_label(o_i, l_i) \in I$ iff $\lambda(o_i) = l_i$.*

Definition 7 encodes the optimization problem by an LP^{MLN} program that can be translated into an ordinary answer set program with weak constraints, such that a solution according to Definition 8 can be extracted from any answer set (cf. Proposition 1).

Example 3 (cont'd). The LP^{MLN} program $\Pi_{\mathcal{C}}(c, E)$ representing the HC for $\mathcal{C}$, c and E as in the previous examples consists of E, with the weighted facts $\alpha : obj(o1)$, $\alpha : obj(o2)$, $\alpha : label(c)$, $\alpha : label(b)$, $\alpha : label(w)$, $\alpha : clf(o1, c, -0.41)$, $\alpha : clf(o1, b, 0)$, $\alpha : clf(o1, w, -2.2)$, $\alpha : clf(o2, c, -2.2)$, $\alpha : clf(o2, b, -2.2)$, $\alpha : clf(o2, w, 1.39)$ and $\alpha : contains(o1, o2)$ (abbreviating the labels), and (2) to (4) from Definition 7.

Without the context encoding E, the single stable model of the program with the highest normalized weight would contain $a\_label(o1, b)$ and $a\_label(o1, w)$; this does not correspond to the correct labeling of the scene shown rightmost in Figure 1. The previous assignment would not satisfy the constraint in E. Hence, when considering E, there are three ways to satisfy it by changing the assigned labels: changing (1) the label of $o1$ to car; or (2) the label of $o2$ to either (2) $building$ or (3) car. As the constraint has weight 1.95 and the label adaptations result in a weight difference of -0.41 for (1) and -3.59 for (2) and (3) for the classifier constraints, only (1) would yield an overall weight improvement. Thus, labeling o_1 as car and o_2 as $wheel$ is the only solution for $\mathcal{C}$ via $\Pi_{\mathcal{C}}(c, E)$ according to Definition 8; this is the correct labeling of the scene.

Note that if the difference between the probability that $o2$ is a $wheel$ and e.g. a $building$ would be small enough, satisfying the constraint (7) in E by changing the label of $o2$ could actually result in a higher overall weight. Thus, context constraints can also decrease the accuracy of the resulting labeling, depending on the quality of the probabilities provided by the classifier.

4 Hybrid Classifier Construction

After having defined HCs abstractly above, we now describe a methodology for constructing a concrete HC for a given CCP, which we also employ in our empirical evaluation. We suggest the following strategy for obtaining a good HC, where the objective is high accuracy of the corresponding solution.

(1) *Data and local classifier preparation*. We assume that we are given a set of CCPs that are all defined over the same CCS for training the HC, together with a solution λ for each CCP representing the ground truth. Obviously, the concrete relations e are usually different in each CCP and first must be extracted from the raw data. For testing the influence of different context constraints, it is crucial to use part of the data for validation to avoid overfitting of the designed constraint encoding. Hence, we split the initial data set into a training set, a validation set and a test set. The local classifier is trained on the associated features of all objects in the CCPs in the training set separately.

(2) *Designing the context encoding*. Although context constraints theoretically could be learned, e.g. by ILP techniques, the current approach assumes that a domain expert with background knowledge on the particular task for the HC has designed the context

encoding. However, failure patterns in the output of the local classifier can be used to guide the design process. For this purpose, the local classifier is first used to classify all objects in the validation set and a *confusion matrix* is compiled, which reveals objects that are difficult to classify for the local classifier and the pairs of labels confused most often. This way, the constraint encoding can be tailored to counter the shortcomings of the local classifier that might result from few, noisy or ambiguous data.
For a constraint c of the form (6) (see Definition 6), its weight w is computed as follows. Determine in the training set the number of ground instances where the label assignment specified by atoms in $\mathcal{L}$ is false (resp., true), denoted by f_c (resp., t_c), provided the context dscribed by the atoms in $\mathcal{R} \cup \mathcal{H}$ of (6) is satisfied. If we would not fix these atoms for counting, e.g. in Example 2 for (8) each object not containing a car part would count as positive instance. However, in this case we are interested in the odds for an object being a car if it has a car part. The weight w of the constraint c is then $\ln(f_c/t_c)$.
(3) *Constraint selection and influence tuning*. After having designed the constraint encoding, the resulting HC could be evaluated already on the test set and used on new CCP instances. However, as discussed in Example 3, context constraints may also decrease the overall accuracy of the results. Hence, the constraint encoding E should be evaluated on the validation set first. As constraints may interact, in general each subset C of constraints must be tested to single out the optimal one wrt. the validation set. As there are exponentially many C, a heuristics is to assess the influence of each constraint c separately and keep it if the accuracy does not decrease if c is applied alone resp. increase if c is dropped from the set of all constraints. In addition, the validation set can be used to tune the influence of the local classifier and the context encoding on the solution, by testing different influence factors.

Example 4 (HC in visual scenes). In the context of visual object classification in scene images, a CCS $S = \langle L, R \rangle$ is created by defining the set L of possible labels for the objects in a class of scenes, e.g. 'car', 'building' and 'tree' for outdoor scenes, and 'table', 'chair' and 'shelf' for indoor scenes, and by fixing the considered set R of relations between objects. Spatial relations such as 'contains', 'intersects' and 'touches' are arguably most prevalent in visual scenes, but R may include also other relations, even relating local features of different objects, such as the binary relation 'has_same_color'.

To turn a set of scene images into a set of CCPs, the image first must be segmented into regions containing single objects. Many procedures for image segmentation exist (see e.g. the survey in [24]); we simply assume here that the image is already segmented. The visual features of the separate segments represent the input to the local classifier, which needs to be trained on a training set of segments representing objects O from training CCPs $\langle S, O, e \rangle$ and the corresponding set L of labels. The extension e of the relations R must be extracted for each CCP separately from the information provided by the scene image and its segmentation, e.g. by computing spatial relations wrt. their bounding boxes or polygon coordinates. Further, implicitly entailed spatial relations can be derived e.g. by employing a spatial reasoning calculus such as RCC8 [16].

Suppose we examine the confusion matrix for the local classifier on the validation set for indoor scenes, and we observe that doors are often misclassified as tables (their surfaces look nearly identical). We then could add a constraint c to the encoding E which states that tables are not contained in walls. We compute f_c and t_c by counting the

objects in the training set contained in a wall that are non-tables resp. tables; presumably the resulting weight $\ln(f_c/t_c)$ is quite high. After having added several constraints to E, we test how removing single constraints (or sets of them) affects the accuracy on the validation set. In that, we might observe that the constraint prohibiting tables in walls actually decreases the overall accuracy, even if it is mostly satisfied on the training data. Indeed, possibly some doors are still wrongly labeled as 'table' while the correct label 'wall' of a wall is changed to an incorrect one. This might have further implications; e.g. if a constraint states that windows only occur in walls, many windows are misclassified too. This illustrates the importance of constraint selection for HC construction.

5 Evaluation

In this section, we evaluate two concrete HCs for two different sets of benchmark instances in order to empirically investigate the effect of applying context constraints. Our goal is to ascertain (Q1) whether a higher classification accuracy is achievable by employing an HC instead of a local classifier, (Q2) which influence the quality of the local classifier has on HC performance, and (Q3) which impact constraint selection and influence tuning have on the solution quality.

We expected that HCs improve the accuracy provided that the local classifier yields sufficiently many correct labels, as a basis to correct the other labels; furthermore, that the accuracy gain can be increased by tuning an HC on the validation set.

Experimental Setup. For experimentation, we implemented an HC framework in Python that enables construction and evaluation of HCs for object classification in scene images. As local classifier, we used *Logistic Regression* from the *scikit-learn* library [13], which we trained on features extracted from image segments obtained by the *bag of keypoints* approach [4], which uses vector quantization of invariant image descriptors. For creating the visual vocabulary, we employed *k-means clustering* and *Scale Invariant Feature Transform* (*SIFT*) descriptors [11]; they are suited for our purpose as they are invariant wrt. transformations, varying illumination and overlapping objects. To detect and compute SIFT descriptors, we used the *OpenCV*[2] library.

Furthermore, we used the *Shapely*[3] package for Python to calculate concrete spatial relations between object-polygons in each scene, based on the *DE-9IM* model [22]. Moreover, for computing the optimal solution of an HC encoding, we utilized CLASP 3.1.2 and GRINGO 4.5.1 [7].

Benchmark Instances. The experiments have been conducted on two sets of scene images from the *LabelMe* dataset [18]. We used a custom segmentation obtained manually, as for testing the impact of context constraints the quality of the available segmentations as well as the user-defined labels varied considerably. The data sets used in our experiments, the segmentation data, the constraint encodings and all results are available at http://tinyurl.com/hc-experiments (linking to a Google Drive folder).

We use (E1) a set of *indoor office scences* and (E2) a set of *outdoor street scenes*, each containing 120 images, which we split into a training set and validation set of 30

[2] http://opencv.org/

[3] https://pypi.python.org/pypi/Shapely

Fig. 2. Example of a typical indoor and outdoor scene from the LabelMe dataset.

images each, and a test set of 60 images. A typical scene from each data set is shown in Figure 2. For both types, we defined 12 labels for the objects that occur most frequently:

- indoor: 'chair' (c), 'monitor' (mn), 'keyboard' (k), 'mouse' (ms), 'table' (t), 'book' (bk), 'shelf' (s), 'wall' (wl), 'board' (br), 'person' (p), 'door' (d) and 'window' (wi),
- outdoor: 'sign' (sg), 'person' (p), 'tree' (tr), 'window' (wi), 'door' (d), 'street' (st), 'car' (c), 'sky' (sk), 'building' (b), 'sidewalk' (si), 'wheel' (wh) and 'trunk' (trn).

Indoor scenes contain 7 to 23 objects, outdoor scenes 7 to 28. In total, (E1) contains 2046 objects, and (E2) has 2280 objects. We extracted the binary spatial relations 'contains', 'close_to', 'above', 'under', 'overlaps', 'contains_in_bottom_part' and 'higher' from the images for use in our constraint encodings, from which we created an HC for each dataset.

Experimental Results. After training the local classifiers on all objects in the training sets, we applied them to the validation sets; for indoor scenes, the average accuracy was 46.3 % and for outdoor scenes 59.7 %. We then constructed HCs, following the methodology from Section 4, by setting up 20 constraints in each case and selecting a subset of 13 constraints after testing different combinations. The accuracy increased for the indoor validation set to 57.1 % and for the outdoor validation set to 69.0 %.

In addition, we tested different influence factors, viz. 0.1, 1, 10, and 100 for the classifier weights; for both data sets, factor 10 yielded the best results due to less erroneous changes of labels correctly predicted by the local classifier. This value is thus suggestive as a default for our use case, and we fixed the influence factors to these values.

We then applied the final HCs to the test sets. Overall, for indoor scenes the accuracy increased from 46.5 % to 55.5 %, and the HC was better than the local classifier on 38 scenes and worse on 11 out of 60. For outdoor scenes, the accuracy increased from 58.1 % to 73.2 %, and the HC was better than the local classifier on 57 cases and worse in 1 case. Thus (Q1), whether an HC can be better than a local classifier, has in these use cases a positive answer. The test results are summarized in Table 1.

Regarding (Q3), i.e., the impact of constraint selection and influence tuning, simply adding all 20 constraints increased the accuracy for indoor scenes from 46.5 % to 52.2 % and for outdoor scenes from 58.1 % to merely 59.6 %; this confirms that constraint selection is a crucial step in HC construction. Influence tuning also proved beneficial and helped to increase the accuracy further in both cases (cf. Table 1).

To provide more details on the effect of the context constraints on the particular labels, Figure 3 shows the confusion matrices of the local classifiers and the final HCs wrt. both test sets. As can be seen e.g. from the rows for books and shelves in the matrix

data set	local classifier	HC -sel -tun	HC -sel +tun	HC +sel -tun	HC +sel +tun
(E1) validation	46.3 %	53.3 %	52.9 %	57.1 %	58.0 %
(E1) test	46.5 %	52.2 %	52.7 %	54.4 %	55.5 %
(E2) validation	59.7 %	63.3 %	67.7 %	68.8 %	70.6 %
(E2) test	58.1 %	59.6 %	69.0 %	71.0 %	73.2 %

Table 1. Results for local classifier and HC with (+sel) or without (-sel) constraint selection and with (+tun) or without (-tun) influence tuning.

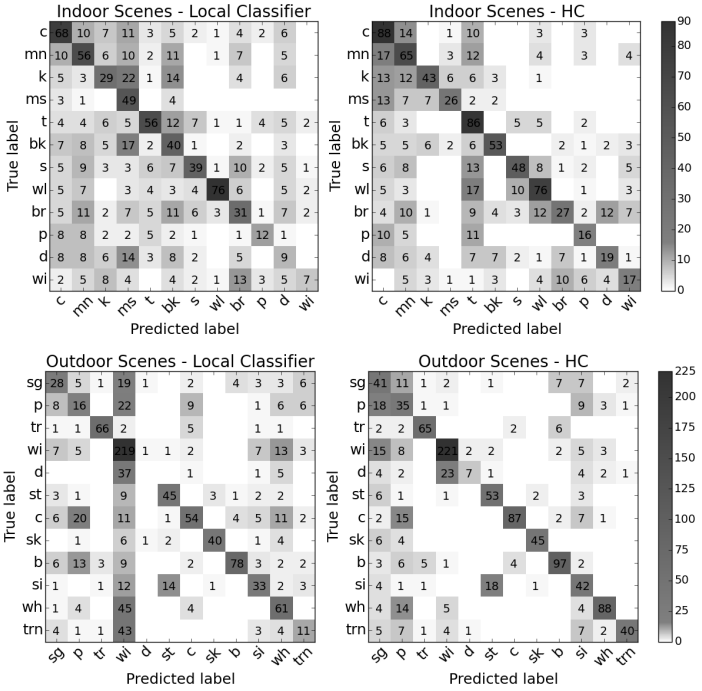

Fig. 3. Confusion matrices of local classifier and HC test results for indoor and outdoor scenes.

for the indoor local classifier, it misclassifies them more than half of the time. By adding a constraint that books are contained in shelves (weight 5.067) and that shelves contain books (weight 3.967), the number of correctly classified shelves increased from 56 to 86, and for books from 40 to 53. Similarly, considering the matrix for the outdoor local classifier, adding a constraint that windows are contained in buildings or in the upper parts of cars (weight 3.863) decreases wrong window classifications from 269 to 37.

As for (Q2), we artificially decreased the quality of the local classifier by training it on a gradually shrunken training set. Notably the benefit of adding context constraints decreased with the accuracy of the local classifier, and when it was below ≈35 % for indoor resp. ≈45 % for outdoor scenes, the local classifier outperformed the HC.

Finding the optimal solution for an HC encoding for a given scene by CLASP usually took just a few seconds, on a Linux machine with an 2.5 GHz Intel Core i5 CPU and 8 GB RAM. We used a timeout of 20 seconds, which was only reached by few instances containing many objects, and did not show to have a negative impact on the results.

6 Discussion and Conclusion

In this paper, we have introduced a general framework for solving CCPs and a methodology for its application. Our tests show that classifying of objects can by significantly improved by considering their semantic context. At the same time, the achievable improvement highly depends on the selected constraints and their interaction, as well as on the quality of the local classifier. If the latter labels most objects incorrectly, the context constraints intuitively lack a reasonable base for correction as wrong labels do not help to infer correct labels of other objects. Overall, we found that best HC results can be obtained when the local classifier performs reasonably well but there is still room for improvement, and when the right set of constraints is selected using a validation set.

As context information is valuable for simultaneous object classification, many approaches—mainly in Computer Vision—exploit scene information and provide either a statistical summary of the image (also called *Gist*) as additional input to the classifier, or exploit relationships between particular objects in a scene (often called the *semantic context*) [14]. Rabinovich and Belongie [14] argue that by considering semantic context, stronger contextual constraints can be imposed (e.g. also spatial relations), and show empirically that they can greatly improve recognition performance. Most approaches using semantic context for label prediction employ some kind of *graphical model*, e.g. *conditional random fields* [15] or *Markov logic networks* [3, 23, 12] in which the mutual influence of labelings is directly encoded by conditional probabilities. Another approach that is very effective for object classification in complex scenes [1] and for collective classification in general [21] is the *iterative classification algorithm* (*ICA*) [20], which iteratively predicts and updates the label of each object based on the current labeling.

Clearly, our approach is related to approaches that consider semantic context or use graphical models. Those above are different from ours as they usually employ probabilistic inference methods such as *Markov chain Monte Carlo* and do not use combinatorial optimization techniques. In addition, often only simple relations such as the co-occurrence frequency of objects were addressed [15, 1]. In contrast, we consider diverse relations between objects extracted from an image (e.g. their position, height and spatial relation to other objects) and they can be combined into more complex relations.

An approach similar to ours is presented in [19], where spatial context is also formalized as constraints to increase collective classification accuracy. However, *Fuzzy CSPs* and *Branch and Bound* are used instead of probabilistic semantics, and only basic relation types are considered. From a bird's eye view, our probabilistic approach achieves a higher accuracy gain with considerably less training data, but further research (requiring an implementation of [19] and suitable benchmarks) is needed for a clear picture.

Regarding future work, we aim to integrate our framework into the HEX formalism [5], which extends ASP with access to external information and would allow us to interface external sources such as an ontology reasoner and a spatial reasoning calculus such as RCC8 directly from within our encoding. In addition, the local classifier could be implemented as an external source as well, resulting in a more modular approach that would also allow information to flow from the program back to the classifier.

References

1. Angin, P., Bhargava, B.: A Confidence Ranked Co-Occurrence Approach for Accurate Object Recognition in Highly Complex Scenes. Journal of Internet Technology 14(1), 13–19 (2013)
2. Buccafurri, F., Leone, N., Rullo, P.: Enhancing Disjunctive Datalog by Constraints. IEEE Trans. Knowl. Data Eng. 12(5), 845–860 (2000)
3. Chechetka, A., Dash, D., Philipose, M.: Relational Learning for Collective Classification of Entities in Images. In: Statistical Relational Artificial Intelligence, AAAI Workshop 2010. AAAI Workshops, vol. WS-10-06. AAAI (2010)
4. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV 2004 (2004), 16 pp
5. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In: Kaelbling, L.P., Saffiotti, A. (eds.) International Joint Conference on Artificial Intelligence, IJCAI 2005. pp. 90–96. Professional Book Center (2005)
6. Galleguillos, C., Rabinovich, A., Belongie, S.J.: Object categorization using co-occurrence, location and appearance. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2008. IEEE Computer Society (2008)
7. Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., Schneider, M.T.: Potassco: The potsdam answer set solving collection. AI Commun. 24(2), 107–124 (2011)
8. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. 9(3/4), 365–386 (1991)
9. Getoor, L.: Introduction to statistical relational learning. MIT press (2007)
10. Lee, J., Wang, Y.: Weighted rules under the stable model semantics. In: Baral, C., Delgrande, J.P., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016. pp. 145–154. AAAI Press (2016)
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. pp. 1150–1157 (1999)
12. Marton, Z.C., Rusu, R.B., Jain, D., Klank, U., Beetz, M.: Probabilistic categorization of kitchen objects in table settings with a composite sensor. In: International Conference on Intelligent Robots and Systems, IEEE/RSJ 2009. pp. 4777–4784. IEEE (2009)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
14. Rabinovich, A., Belongie, S.J.: Scenes vs. objects: A comparative study of two approaches to context based recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2009. pp. 92–99. IEEE Computer Society (2009)
15. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.J.: Objects in context. In: IEEE 11th International Conference on Computer Vision, ICCV 2007. pp. 1–8. IEEE Computer Society (2007)
16. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W.R. (eds.) Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning, KR 1992. pp. 165–176. Morgan Kaufmann (1992)
17. Richardson, M., Domingos, P.M.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
18. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. International Journal of Computer Vision 77(1-3), 157–173 (2008)

19. Saathoff, C., Staab, S.: Exploiting spatial context in image region labelling using fuzzy constraint reasoning. In: Ninth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2008. pp. 16–19. IEEE Computer Society (2008)
20. Sen, P., Namata, G., Bilgic, M., Getoor, L.: Collective classification. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 189–193. Springer (2010)
21. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine 29(3), 93–106 (2008)
22. Strobl, C.: Dimensionally extended nine-intersection model (DE-9IM). In: Shekhar, S., Xiong, H. (eds.) Encyclopedia of GIS., pp. 240–245. Springer (2008)
23. Tran, S.D., Davis, L.S.: Event modeling and recognition using markov logic networks. In: Forsyth, D.A., Torr, P.H.S., Zisserman, A. (eds.) 10th European Conference on Computer Vision
24. Zhang, H., Fritts, J.E., Goldman, S.A.: Image segmentation evaluation: A survey of unsupervised methods. Computer Vision and Image Understanding 110(2), 260–280 (2008)