On different modes of continuous learning of visual properties [*]

Danijel Skočaj, Barry Ridge, and Aleš Leonardis
University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25, SI-1001 Ljubljana, Slovenia
{*danijel.skocaj, barry.ridge, ales.leonardis*}*@fri.uni-lj.si*

O različnih načinih inkrementalnega učenja vizualnih lastnosti

Za vsak spoznavni sistem, tudi umetni, je zelo pomembno, da se je sposoben učiti in pridobljeno znanje nadgrajevati. V tem članku obravnavamo različne načine inkrementalnega učenja, ki to omogoča. Predstavimo učenje, pri katerem uporabnik oz. učitelj zagotovi umetnemu sistemu vse potrebne informacije, ki jih potrebuje, nato učenje, pri katerem sistem zahteva od uporabnika informacije glede na stopnjo nedoločenosti, ter učenje, pri katerem sistem nadgrajuje svoje znanje popolnoma brez pomoči uporabnika. V članku tudi predstavimo metodo, ki omogoča inkrementalno učenje vizualnih lastnosti predmetov na vse tri načine. Z eksperimentalnimi rezultati vse tri pristope tudi ovrednotimo.

1 Introduction

In a real world environment, a cognitive system should possess the ability to learn and adapt in a continuous, open-ended, life-long fashion from the variable input that such an environment would present. As an example of such a learning framework, we need look no further than at the successful application of *continuous learning* in human beings. For example, a child will learn to recognise what a cat is by seeing a few examples of one. Later, as the child encounters more cats that are different to the original examples, he/she will not only recognise the new cats as being cats, but will also update his/her representation of what a cat is, based on the salient properties of the new examples and without having visual access to the previous examples.

While the primary focus of this idea is on the incremental nature of the knowledge update, another key aspect is the scrutinisation of various visual features and the determination of which features are useful for representing the visual attributes of the object in question. Since a continuous learning framework would not retain complete data from previously learned samples, it would not have the luxury of being able to reference specific details across multiple samples in order to learn. Given this restriction, continuous learning perhaps lends itself to an abstract multi-modal system involving interaction with a user.

In this paper we discuss such cross-modal learning, namely association between *words* and simple *visual features*, such as hue or intensity values of the corresponding pixels. In particular we will present a method for learning *visual attributes* (e.g., colour, shape) and their *qualitative values* (e.g., red, yellow; circular, square). The problem of coupling words and images involves computer vision and linguistic methods, therefore it has been tackled by the researchers from both communities (see e.g., [1, 3]). In their work the emphasis is on association mechanisms, which are mainly based on batch approaches. In this paper we instead focus on an incremental learning paradigm and different types of incremental learning mechanisms that require different levels of supervision provided by a tutor.

The paper is organised as follows. In the next section we propose a general framework for continuous learning. In Section 3 we present a specific method for incremental learning and embed it in the proposed framework. We then present the experimental results in Section 4. Finally, we summarise the paper and outline some work in progress.

2 Continuous learning framework

The interaction between a tutor and an artificial cognitive system plays an important role in a continuous learning framework. One goal of the learning mechanism could be to find associations between words spoken by the tutor and visual features automatically extracted by the cognitive visual system, i.e. to ground the semantic meaning of the visual objects and their properties into the visual features [2].

When implementing a continuous learning mechanism, two main issues have to be addressed. Firstly, the representation, which is used for modeling the observed world, has to allow for updates when pre-

[*] This research has been supported in part by the following funds: Research program Computer Vision P2-0214 (RS), EU project CoSy, and EU project VISIONTRAIN.

sented with newly acquired information. This update step should be efficient and should not require access to the previously observed data while still preserving the previously acquired knowledge. Secondly, a crucial issue is the quality of the updating, which highly depends on the correctness of the interpretation of the current visual input. With this in mind, several learning strategies can be used, ranging from completely supervised to completely unsupervised learning. In this paper we discuss three such strategies:

- **Tutor-driven approach (*TD*).** The correct interpretation of the visual input is always correctly given by the tutor.
- **Tutor-supervised approach (*TS*).** The system tries to interpret the visual input. If it succeeds to do this reliably, it updates the current model, otherwise asks the tutor for the correct interpretation.
- **Exploratory approach (*EX*).** The system updates the model with the automatically obtained interpretation of the visual input. No intervention from the tutor is provided.

We further divide *tutor-supervised learning* into two sub-approaches:

- **Conservative approach (*TSc*).** The system asks the tutor for the correct interpretation of the visual input whenever it is not completely sure that its interpretation is correct.
- **Liberal approach (*TSl*).** The system relies on its recognition capabilities and asks the tutor only when its recognition is very unreliable.

Similarly, we also allow for *conservative* and *liberal exploratory* sub-approaches (*EXc*, *EXl*).

To formalise the description of these approaches, let us assume that the recognition algorithm always gives one of the following five answers when asked to confirm the interpretation of the current visual scene (e.g., the question may be: "Is this red?"): *'yes' (YES), 'probably yes' (PY), 'probably no' (PN), 'no' (NO),* and *'don't know' (DK).* Table 1 presents actions that are undertaken after an answer from the recognition process is obtained. The system can either *ask* the tutor for the correct interpretation of the scene (or the tutor provides it without being asked), *update* the model with its interpretation, or do nothing. As can be seen from Table 1, the system can communicate with the tutor all of the time (*TD* learning), often (*TSc*), occasionally (*TSl*) or even never (*EX* learning).

To speed up the initial phase of the learning process and to enable development of consistent basic concepts, one could start with mainly tutor-driven learning with many user interactions. These concepts would then be used to detect new concepts with limited help from the user. Later on in the process, when the ontology is sufficiently large, many new concepts could be acquired without user interaction.

Table 1: Update table.

	YES	PY	PN	NO	DK
TD	ask	ask	ask	ask	ask
TSc	upd	ask	ask	/	ask
TSl	upd	upd	/	/	ask
EXc	upd	/	/	/	/
EXl	upd	upd	/	/	/

3 Our method

The main task of the learning algorithm is to assign associations between visual features and attribute values. It has to consider two main issues: *consistency* and *specificity*. It must determine the visual features that are *consistent* over all images sharing a particular visual attribute and that are, at the same time, *specific* for that visual attribute only.

With these requirements in mind, we have designed algorithms for incremental learning and recognition of visual properties based on a generative representation of features associated with visual attributes. Each visual attribute is associated with a visual feature that best models the corresponding images according to the consistency and specificity criteria mentioned above. The learning algorithm thus selects from N_F one-dimensional features (e.g., median hue value, area of segmented region, etc.), the feature whose values are most consistent over all images sharing the same visual attribute (i.e. the variance is small and the feature values are concentrated around the mean value). At the same time it also ensures that the same does not hold true for some other visual attribute, thus satisfying the specificity criterion. A visual attribute value is therefore represented with the mean and variance of the best feature.

The main idea is described in an algorithmic form in Algorithm 1. In the basic batch form, the algorithm requires all training images to be given in advance, together with a list of attribute values (e.g., red, large, square) for each image. Since the mean and variance of a set of feature values can be calculated in an incremental way without losing any information, this algorithm can be incrementalised. Algorithm 2 shows the pseudo-code of one update step. Using this algorithm, the model can be sequentially updated by considering only one image at a time.

Once the models of visual attributes have been acquired, the system is able to recognise visual properties of a novel object using Algorithm 3 (e.g., answering the question "Is this red?"). If the value of a feature associated with a particular attribute value is quite close to the values observed during learning (i.e. it is very close to the mean of previously observed values), then the system answers 'yes'. Based on the distance from the typical value of the feature (considering variance as well), the system may also answer "probably yes", "probably not", or "no". By changing the thresholds $Tyes$, Tpy, and Tno, we can achieve more conservative or more liberal behaviour of the recognition algorithm. Combining this recognition algorithm with the incremental learning algorithm and by considering the update table presented in Table 1, we arrive at the incremental learning framework described in the previous section.

Algorithm 1 : Batch learning

```
Input: Set of training images $\mathcal{X}$, list of attribute values $\mathcal{AV}_i$ for every training image $X_i$
Output: Models of attribute values $mAV_i, i = 1 \dots N_{AV}$
1: Extract all visual features $F_j, j = 1 \dots N_F$ from every training image in $\mathcal{X}$.
2: for each $AV_i, i = 1 \dots N_{AV}$ do
3:     Find a set of images $\mathcal{X}_i$ containing all images labeled with $AV_i$.
4:     Calculate means and variances of the values of every feature $F_j$ over all images in $\mathcal{X}_i$.
5: end for
6: Calculate min and max of all the values of every feature $F_j$.
7: Normalise all the variances with the obtained intervals of feature values, i.e.,
   $nvar_{ij} = var_{ij}/(maxVar_j - minVar_j)^2$,
   $i = 1 \dots N_{AV}, j = 1 \dots N_F$.
8: for each $AV_i, i = 1 \dots N_{AV}$ do
9:     Select the feature $F_j$ with the smallest normalised variance $nvar_{ij}$.
10:    Store mean and variance of the selected $F_j$ to form $mAV_i$, a model of $AV_i$.
11: end for
```

Algorithm 2 : Update step

```
Input: Models of attribute values $mAV_i, i = 1 \dots N_{AV}$, feature statistics $FS$, new image $X$ and corresponding attribute value $AV$
Output: Updated $mAV_i, i = 1 \dots N_{AV}$ and $FS$
1: If the model for $AV$ has not been learned yet, initialise it.
2: Update feature means and variances related to $AV$ (stored in $FS$).
3: Update total feature mins and maxs.
4: Proceed with the steps 7-11 of Algorithm 1.
```

4 Experimental results

We tested the algorithms by running a number of experiments on both artificial and real data. Basic shapes of various different colours and sizes were selected as test objects. Some of them are depicted in Fig. 1. We considered three visual attributes (colour, size and shape), and ten values of these visual attributes altogether (red, green, blue, yellow; small, large; square, circular, triangular, and rectangular).

Algorithm 3 : Recognition

```
Input: Image $X$, question "Is this AV?"
Output: Answer.
1: If the model for $AV$ has not been learned yet, answer 'Don't know.'
2: Determine which feature $F_j$ the attribute value $AV$ is associated with in the model $mAV$.
3: Extract the value of this feature $F_j$ from the image $X$.
4: Calculate $d = (F_j - mAV.mean)/\sqrt{mAV.var}$.
5: If $d \in [0, Tyes]$, answer 'Yes.'
6: If $d \in (Tyes, Tpy]$, answer 'Probably yes.'
7: If $d \in (Tpy, Tno]$, answer 'Probably not.'
8: If $d \in (Tno, \infty)$, answer 'No.'
```

The objects were first perspective-rectified and segmented from the background. Then the visual features were extracted. We used six simple one-dimensional features; three colour features (median of hue, saturation and intensity over all pixels in the segmented region) and three simple shape descriptors (area, perimeter and compactness of the region). The main goal was to find associations between ten given attribute values and six extracted features.

We put half of the images in the training set and other half in the test set (64 per half in the case of synthetically generated images and 70 per half in the case of real images). We embedded the proposed learning method in the learning framework and kept incrementally updating the representations with the training images using different learning strategies. At each step, we evaluated the current knowledge by recognising the visual properties of all test images. The evaluation measure we used is *recognition score*, which rewards successful recognition (true positives and true negatives) and penalises incorrectly recognised visual properties (false positives and false negatives).

Results (the curves of the evolution of the recognition score through time) of the experiment on the synthetic images (averaged over 40 trials on different sets of generated images with added noise) are presented in Fig. 2(a). All different learning strategies

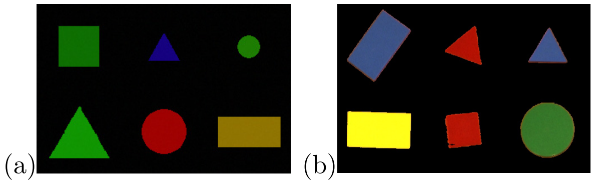

Figure 1: (a) Synthetic images. (b) Perspective-rectified and segmented real images.

presented in Section 2 were tested. First, we applied the incremental learning process from the very beginning, starting with one training image (denoted as *TSc1*, *TSl1*, etc.). Then we repeated the experiment by first applying the batch algorithm on the first 10 images, and then continuing by incrementally adding the rest of the images (*TSc10*, *TSl10*, etc.). Fig. 2(a) shows the plots of recognition scores, while Fig. 2(b) plots the number of questions the system asked the tutor at each step (i.e., how much data were given to the system by the tutor).

In the experiment on the synthetic images, the tutor-driven learning successfully associates the colours of the input objects with the *hue* feature, their sizes with the *area* feature and their shapes with the *compactness* feature. Recognition of visual attributes is very successful; it almost gets the maximal score (640 in this case). Tutor-supervised learning proved to be quite successful as well. In this case conservative strategy yields better results, since it asks the tutor for reliable information more often. This is also evident from Fig. 2(b). In the beginning the system does not have a lot of knowledge, so the tutor is asked for help more frequently. After the knowledge is acquired, the number of questions decreases. The explorative approach, which does not involve interaction with the tutor from the very beginning, does not significantly improve the model. So, as expected, there is a trade-off between the quality of results and the wish to decrease the need for user interaction. Similar conclusions can also be drawn from the results of the experiment on real data shown in Fig 2(c).

5 Conclusion

In this paper we presented a framework for continuous learning, which enables three modes of learning requiring different levels of tutor supervision. We proposed a method for incremental learning of visual properties by building associations between words describing objects' visual properties and visual features extracted from images. By embedding this method into the proposed learning framework, we were able to experimentally evaluate three learning strategies. The main conclusion is that the learning process should start with tutor-driven learning to enable development of consistent basic concepts. Once these concepts are acquired, the system can take the initiative and keep upgrading the knowledge in a tutor-supervised way, and when the knowledge is stable enough, even in an exploratory way.

Beyond this initial work, we aim to improve the learning method as well as to further analyse the proposed framework and evaluate different learning strategies under various conditions and in various applications.

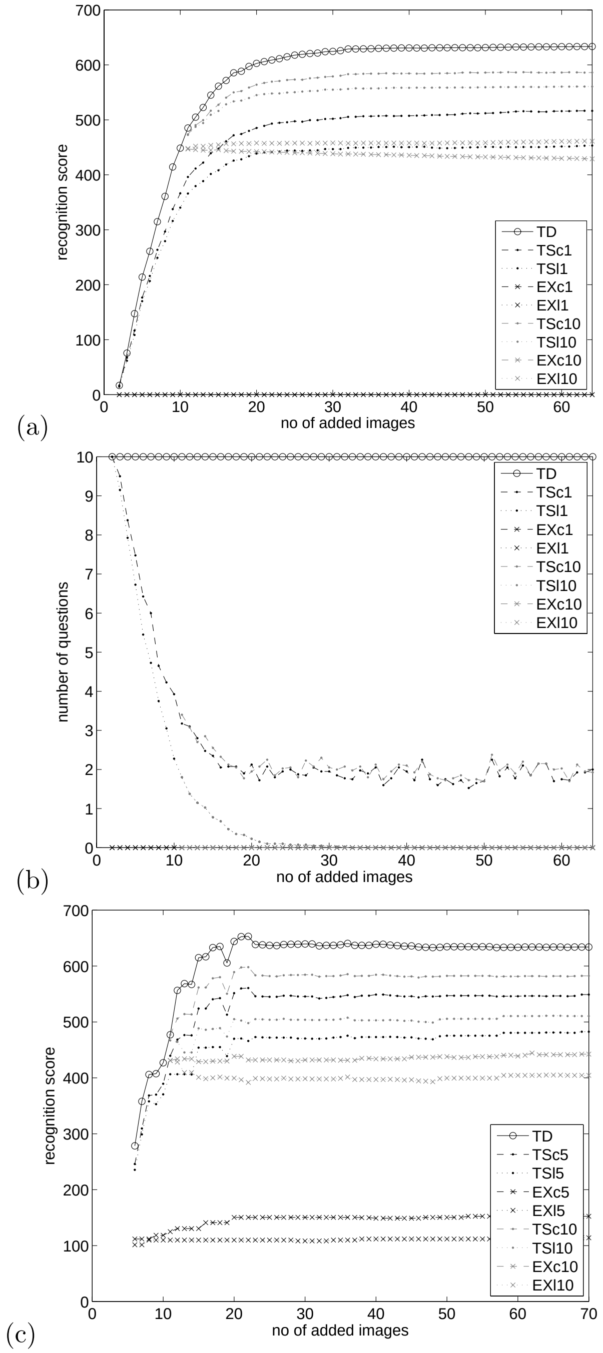

Figure 2: (a) Rec. score and (b) number of questions on synthetic images, (c) rec. score on real images.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, M. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3:1107–1135, 2003.

[2] S. Harnad. The symbol grounding problem. *Physica*, D(43):335–346, 1990.

[3] D. Roy. Learning words and syntax for a visual description task. *Computer Speech and Language*, 16(3):353–385, 2002.