Classifier Adaptation at Prediction Time

Amelie Royer[1], Christoph H. Lampert[2]
[1]ENS Rennes. [2]IST Austria.

Recent object recognition systems, such as convolutional neural networks, yield remarkable accuracy despite challenging large-scale settings. They are usually trained and evaluated on large sets of i.i.d test images. However, many real-life applications do not fit this purpose. For instance, in the task of guiding a robot in an office environment, classes "chair" and "desk" are significantly more likely to occur than "giant panda" or "canoe". Moreover, smaller local visual contexts arise from the sequentiality of the input image stream. This change in distribution between training and prediction time is a problem of *domain adaptation*, and is often seen as a potential cause for loss of accuracy which is to be prevented. In this work, we argue the opposite, and aim to take advantage of the label correlations at prediction time to increase the classification accuracy. This problem is similar to the work of Jia and Darrell in [1] on adapting a classifier to a given subset of queries from the ImageNet hierarchy. However, while they assume the queries are i.i.d. samples from an unknown sub-tree, we do not make any assumptions on the relations between classes at prediction time, and we aim to exploit sequential information. Our main contribution is a method for adapting a pre-trained classifier on-the-fly at prediction time. We also propose several approaches for generating "realistic" sequences of queries (i.e. images belonging to a joint semantic context) as a framework to evaluate our approach.

We consider a base pre-trained multi-class classifier, f, which assigns a class label $y \in \mathcal{Y}$ to an input image $x \in \mathcal{X}$. We also distinguish the data distribution at training time (denoted by $P(x,y)$) from the one at prediction time ($Q(x,y)$). Finally, we assume f to be probabilistic, i.e. $f(x) = \operatorname{argmax}_{y\in\mathcal{Y}} f_y(x)$, where the scores $f_y : \mathcal{X} \to [0,1]$ reflect the conditional probabilities $P(\cdot\,|y)$. In a realistic setting, the labels distribution at prediction time, $Q(y)$, may differ from the one used for training the classifier f, $P(y)$. We assume the latter distribution is known (in practice, for large data sets, training labels are often uniformly distributed), and that $Q(y)$ is a static distribution we have no knowledge of. On the contrary, the objects' visual appearance does not change between training and prediction time, hence $Q(x|y) = P(x|y)$. Under these assumptions, we use Bayes' rule to derive a classifier g which is the class-prior adaptation of f at prediction time [3]:

$$g(x) = \operatorname{argmax}_{y\in\mathcal{Y}} g_y(x) \qquad \text{with } g_y(x) = \frac{Q(y) f_y(x)}{P(y)} \tag{1}$$

Given $(x_n, y_n)_n$, a sequence of images at prediction time and their ground-truth classes, we estimate the class distribution using a classical Bayesian approach, with a symmetric Dirichlet distribution, $Dir(\alpha)$, as prior [2, Chapter 3]. The optimal estimate after seeing the i first queries is given by:

$$\hat{Q}_i(y) = \frac{n_i(y) + \alpha}{i + \alpha|\mathcal{Y}|}, \qquad \text{with } n_0(y) = 0 \tag{2}$$

where $n_i(y)$ is the number of occurrences of class y among the i first queries. We compute these estimates iteratively: given an incoming query x_i, we first predict a label $g^{(i)}(x_i)$ based on the current estimates. A feedback on the prediction is then received, and used to update $n_i(y)$ by a score $\delta_i(y)$, leading to the updated classifier adaptation, $g^{(i+1)}$.

In a classical online setting, the true class of the query, y_i, is given as feedback, thus computing $n_i(y)$ exactly is trivial (Eq. 3). In a reinforcement scenario (Eq. 4), only the information of whether the prediction $g^{(i)}(x_i)$ is correct or not is available: When the prediction matches the true label, we come back to the previous case, and in the opposite case, we increase $n_i(y)$ by a small uniform weight, except for the wrongly predicted label. Finally, no feedback is given in the unsupervised setting, hence we have to rely on the current estimates of the distribution for the update (Eq. 5).

While this approach is well-suited when $Q(y)$ is assumed to be static, we also propose a *dynamic* variant of the classifier adaptation, more fitted to scenarios where the labels distribution in the prediction phase may vary over time. The idea is to compute the same estimates as before, but over the L most recently seen labels rather than all the previous ones: This sliding window construction amounts to gradually forgetting the outdated history, hence we now focus on the context arising from the more recent queries.

$$\delta_i(y) = [\![y = y_i]\!] = 1 \text{ if } y = y_i, \text{ otherwise } 0 \tag{3}$$

$$\delta_i(y) = \begin{cases} [\![y = y_i]\!], & \text{if } y_i = g^{(i)}(x_i) \\ \frac{[\![y \neq y_i]\!]}{|\mathcal{Y}|-1}, & \text{otherwise} \end{cases} \tag{4}$$

$$\delta_i(y) = \mathbb{E}_{\tilde{y}\sim Q_i(\tilde{y}|x_i)}([\![y = \tilde{y}]\!]) = \frac{g_y^{(i)}(x_i)}{\sum_{\tilde{y}} g_{\tilde{y}}^{(i)}(x_i)} \tag{5}$$

We conduct experiments on ImageNet's ILSVRC2010 and ILSVRC2012 validation sets (1k classes, 50k images each). We evaluate against a pre-trained neural network classifier from the CCV library with state-of-the-art accuracy, and a multi-class SVM combined with Platt scaling for probabilistic outputs which we trained using the JSGD toolkit. The choice of these two very different settings emphasizes the fact that the adaption performances are not biased towards the pre-trained classifier. To the best of our knowledge, there is no database offering realistic sequences of images with correlated labels at prediction time, hence we propose several ways of generating such sequences. The first method (*TXT*) relies on natural language: We create each sequence by browsing a classic English book and extracting ImageNet classes along the way. The second approach is to build a 2D projection of the class space by exploiting a semantic distance between ImageNet classes (e.g. the *least common ancestor-distance* as induced by the WordNet hierarchy). We used *Multi Dimensional Scaling* and *Kernelized Sorting* for the projection step, leading to two different databases (*MDS* and *KS*). The label sequences are then generated as a random walk on this space.

Table 1: *Top-5* classification error rates without classifier adaptation (CNN), with regular adaptation (Adapt) or with dynamic adaptation (Dyn.) for the online and unsupervised setting, on the ILSVRC2012 dataset. Bold text marks the lowest entries in each setting if the difference is 10^{-3}-significant according to a Wilcoxon signed rank test.

	CNN (ILSVRC12)	Adapt (Online)	Adapt (unsup.)	Dyn. (Online)	Dyn. (unsup.)
TXT	19.8 ± 1.9	$\mathbf{12.8 \pm 1.9}$	$\mathbf{14.8 \pm 1.8}$	14.5 ± 1.7	16.4 ± 1.7
MDS	16.1 ± 6.5	$\mathbf{5.2 \pm 3.0}$	$\mathbf{6.8 \pm 3.6}$	6.6 ± 2.6	8.1 ± 2.9
KS	16.4 ± 1.8	15.2 ± 1.7	16.5 ± 1.7	$\mathbf{11.8 \pm 1.3}$	$\mathbf{14.2 \pm 1.5}$
RND	$\mathbf{16.5 \pm 0.6}$	18.7 ± 0.7	18.0 ± 0.7	17.2 ± 0.6	16.9 ± 0.6

Table 1 summarizes the main results of our experiments with the CCV neural network classifier. We conclude that the proposed adaptation scheme improves the accuracy of recognition systems in a real world setting, even for state-of-the-art classifiers and few to no online feedback. This especially holds for the TXT and MDS data sets. The dynamic variant proves its usefulness on the KS sequences, for which the labels distribution tend to vary quickly over time. Finally, in the usual i.i.d. setting, there is no context to benefit from through adaptation, hence a lower accuracy, which, however, is small enough to be tolerable, especially when using dynamic adaptation.

[1] Yangqing Jia and Trevor Darrell. Latent task adaptation with large-scale hierarchies. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[2] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. Cambridge, 2012.

[3] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural Computation*, 14(1):21–41, 2002.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.