On-the-fly Domain Adaptation of Binary Classifiers

Sébastien Piérard SEBASTIEN.PIERARD@ULG.AC.BE
Alejandro Marcos Alvarez AMARCOS@ULG.AC.BE
Antoine Lejeune ANTOINE.LEJEUNE@ULG.AC.BE
Marc Van Droogenbroeck M.VANDROOGENBROECK@ULG.AC.BE
Montefiore Institute, University of Liège, Belgium

Abstract

This work considers the on-the-fly domain adaptation of supervised binary classifiers, learned off-line, in order to adapt them to a target context. The probability density functions associated to negative and positive classes are supposed to be mixtures of the source distributions. Moreover, the mixture weights and the priors are only available at runtime. We present a theoretical solution to this problem, and demonstrate the effectiveness of the proposed approach on a real computer vision application. Our theoretical solution is applicable to any classifier approximating Bayes' classifier.

1. Introduction

Supervised machine learning techniques traditionally assume that the samples contained in the learning set and those encountered in the test set are drawn from the same distribution. However, this assumption is usually not verified in practice (Daumé III & Marcu, 2006), and, in that case, the model learned from the learning set is suboptimal with respect to the test set (Lin et al., 2002; Pan & Yang, 2010). Domain adaptation methods (Daumé III & Marcu, 2006; Mansour et al., 2008) are designed to tackle this limitation by adapting the model trained on the learning set, referred to as the source domain, to the test set, referred to as the target domain.

Traditional domain adaptation methods assume that the target distribution is fixed and known. However, in fields such as computer vision or spam detection, the target distribution can change over time or can just not be unique. For example, different video games played on the same game console could well expect different human poses, i.e. the set of expected human poses is dependent on the game itself. In the field of spam detection, different users could consider the same email differently based on their preferences. In both cases, it is suboptimal, or even just not possible, to specify a unique target distribution that would fit all possible target distributions.

In the case the target distribution changes, traditional domain adaptation techniques usually require that the classifier is re-trained so that it is adjusted to the new target distribution. This requirement might be very time-consuming and can be a severe limitation in real-time applications like those encountered in computer vision, or in cases where there exist many different target domains. Domain adaptation methods performing on-the-fly adaptation, without the need of re-training the classifier, are thus an active research topic. To tackle this problem, some approaches propose to use the ideas of transfer learning within the framework of online learning (Ge et al., 2013; Zhao & Hoi, 2010), thus allowing the classifier to automatically adapt to a change in the distribution, even if the change occurs gradually during the course of utilization. Traditionally, these approaches first learn some classifiers on the source domains, and then iteratively update the classifier each time a new data point from the target domain is becoming available (Ge et al., 2013; Zhao & Hoi, 2010). Other approaches (Jain & Learned-Miller, 2011) propose an adaptation procedure that considers the source classifiers as a basis, and adapts this basis to any new data point from the target domain. Finally, the approach proposed in this paper instead considers that the adaptation of the classifier learned from the source domain is performed only once, when some details of the target distribution are known. This procedure still allows to re-adapt the classifier if the target distribution changes over time.

Appearing in *Proceedings of BENELEARN 2014*.

In this paper, we tackle a very general unsupervised domain adaptation problem in the case of binary classification where several source, as well as target, domains are considered. More specifically, the conditional distribution of each class in the target distribution is represented as a mixture of the conditional distributions of each class in the source domains. The approach proposed in this paper is an unsupervised domain adaptation method, as no assumption is made regarding the existence of labeled data in the target domain. However, the approach can easily be extended to take advantage of labeled data in the target domain.

The domain adaptation technique proposed in this paper adapts, in an on-the-fly and efficient manner, a set of base classifiers, learned on source distributions, to a given target distribution. The main motivation here is to avoid re-training the classifier each time the target distribution changes. In this setting, we assume that the base classifiers are learned once and for all on the source distributions. The adaptation is then performed through a non-linear combination of the base classifiers trained on the source distributions. Once the target distribution is known, the final classifier is created on-the-fly by simply combining the outputs of the base classifiers. The combination formula is theoretically derived such that the final classifier is optimal according to Bayes' rule for the target distribution.

One advantage of our method is that the base classifiers are trained only once on the source data. The adaptation does not require another training procedure. Another advantage of the proposed method is that the knowledge of the target application is only needed when the model is to be applied.

We now illustrate our domain adaptation setting on two examples, namely the human motion analysis problem in computer vision and the spam detection problem. One typical approach (Barnich et al., 2006) to solve the human motion analysis problem consists in segmenting the images and then classifying the segmented parts into two classes: humans and non-humans. However, the distribution of human silhouettes strongly depends on several factors such as the season (people wear different clothes in winter and in summer), the geographical location (kimonos are less likely in Europe than in Japan) or the person activity (a silhouette depends on the pose, and the distribution of poses is activity dependent). Learning a general model that can cope with each possible context is thus impossible and domain adaptation techniques are a promising way to develop methods able to adapt to different contexts. A similar setting can be found in the context of spam detection. Assume that you have at your disposal three databases containing emails labeled as spam and non-spam respectively in English, French and Spanish. If a user is likely to receive emails in the three different languages, a spam filter should use the three databases to learn a spam detection model. On the other hand, if a user receives emails only in English, then there is no reason to include spam examples from the French or Spanish databases. In such a situation, it would be desirable to develop a classifier tailored to a specific user such that the classifier best suits the user's needs. Obviously, learning a specific classifier for each user is intractable, and traditional domain adaptation techniques cannot be used in this situation.

This paper is organized as follows. Section 2 introduces our problem statement and Section 3 presents our domain adaptation technique. Finally, our method is applied on a typical computer vision problem in Section 4, and Section 5 gives a short conclusion.

2. Problem statement

Let us assume that we have a dataset containing pairs (x, y) such that x represents an object, and $y \in \{-1, +1\}$ represents its associated class label. Let us further assume that the negative pairs of this dataset, *i.e.* those pairs for which $y = -1$, are partitioned into γ^- sets denoted S_k^-, with $k = 1 \dots \gamma^-$. The positive pairs are similarly partitioned into γ^+ sets S_l^+, with $l = 1 \dots \gamma^+$. The probability density function (*pdf*) underlying each set is thus a conditional *pdf* denoted by $\rho_k(x|y=-1)$ and $\rho_l(x|y=+1)$, respectively for S_k^- and S_l^+. We use the shorthand notation $\rho_k^-(x)$ and $\rho_l^+(x)$ to denote these distributions.

The target distribution is expressed as a mixture of the source distributions and is given by

$$\rho(x) = p^- \rho^-(x) + p^+ \rho^+(x) \tag{1}$$

with

$$\rho^-(x) = \sum_{k=1}^{\gamma^-} \beta_k^- \rho_k^-(x) \ , \ \rho^+(x) = \sum_{l=1}^{\gamma^+} \beta_l^+ \rho_l^+(x) \tag{2}$$

where $p^- = P(y=-1)$ and $p^+ = P(y=+1)$ are the priors, and $\beta_k^- \geq 0$, $\beta_l^+ \geq 0$ are the mixture weights such that $\sum_{k=1}^{\gamma^-} \beta_k^- = 1$ and $\sum_{l=1}^{\gamma^+} \beta_l^+ = 1$. These weights determine the target distribution and are assumed unknown during the learning procedure.

It can be noticed that one special case of this setting is the case where the target distribution is a simple mixture of the source distributions, as studied in (Mansour et al., 2008). Indeed, if $\gamma^- = \gamma^+ = \gamma$

and $\beta_j^- = \beta_j^+ = \beta_j$, then $\rho(x) = \sum_{j=1}^{\gamma} \beta_j \rho_j(x)$ with $\rho_j(x) = p^- \rho_j^-(x) + p^+ \rho_j^+(x)$.

Unlike traditional domain adaptation methods, we do not require that data is provided for the target domain. We rather require that the mixture weights are known. In some cases, it is reasonable to assume that the weights are known. In other cases, such an assumption is unrealistic and other means have to be used to estimate them.

We now give two situations in which it is reasonable to assume that the mixture weights are known. First, let us consider the installation of a video surveillance system. In that case, the weights can be manually set by a human expert so that the system is tailored to the context of use. In that case, the mixture weights are known because they are set by the installer. A second example can be found in the context of video game. A game console knows which game is being played, and therefore knows what the users are expected to do. Adjusting the mixture weights at runtime is thus straightforward, and does not require to estimate them from the data acquired by the sensor.

For those applications for which we cannot assume that the mixture weights are known, we can still estimate them from data in the target domain. One way to solve this non-trivial task is to use, for example, expectation-maximization (*EM*) (Dempster et al., 1977). Estimating the mixture weights from target data using *EM* is a recurring task in domain adaptation methods (Daumé III & Marcu, 2006; Storkey & Sugiyama, 2006).

3. Proposed method

A straightforward strategy to adapt a classifier to the mixture weights and to the priors is to construct a new database drawn from the current *pdf* according to the current mixture weights and priors, and to learn a new classifier from it. This procedure has to be repeated each time the weights or the priors change. In order to obtain unbiased results with respect to the priors, that database would ideally contain n^- negative learning samples and n^+ positive learning samples, with $n^-/(n^- + n^+) = p^-$ and $n^+/(n^- + n^+) = p^+$. Moreover, in order to take the mixture weights into account, the database has to be populated with $n^- \beta_k^-$ samples drawn *i.i.d.* from ρ_k^-, $\forall k \in \{1, \ldots, \gamma^-\}$, and with $n^+ \beta_l^+$ samples drawn *i.i.d.* from ρ_l^+, $\forall l \in \{1, \ldots, \gamma^+\}$.

Despite its simplicity, this adaptation strategy presents two drawbacks in practice. Firstly, it is not suited for real-time applications in which the priors or the mixture weights change quickly, since learning new models is usually a time consuming task. Moreover, for most systems replicated many times and whose behavior depends on the context of use (such as video surveillance systems or spam detection), learning for each replica a new model adapted to the context is clearly impractical, even if the context is static for each replica. Secondly, it is impossible to construct a database with the ideal blending proportions when a prior or some mixture weights are low without facing the classical difficulties related to imbalanced datasets (Chan & Stolfo, 1998; Japkowicz & Stephen, 2002; Barandela et al., 2003; Batista et al., 2004).

As a practical alternative to the previous strategy, we propose to learn offline $\gamma^- \times \gamma^+$ classifiers and to combine at runtime their outputs in such a way that the resulting classifier is adapted to the target distributions ρ^- and ρ^+ and to the priors p^- and p^+. In other words, we propose to simulate the behavior of the previous strategy (see Section 3.1) at a lower computational cost thanks to an original combination rule. Let x be an object to classify and $z^T(x)$ be the output of a binary classifier adapted to the target domain. We note $z_{k,l}(x)$ the output of a binary classifier learned off-line from a database populated with $n_{k,l}^-$, $n_{k,l}^+$ samples taken randomly in ρ_k^-, ρ_l^+ respectively. The method we propose is organized in three steps. First, the outputs of the $\gamma^- \times \gamma^+$ models learned offline are computed:

$$\begin{pmatrix} z_{1,1}(x) & z_{1,2}(x) & \ldots & z_{1,\gamma^+}(x) \\ z_{2,1}(x) & z_{2,2}(x) & \ldots & z_{2,\gamma^+}(x) \\ \vdots & \vdots & \ddots & \vdots \\ z_{\gamma^-,1}(x) & z_{\gamma^-,2}(x) & \ldots & z_{\gamma^-,\gamma^+}(x) \end{pmatrix}. \quad (3)$$

Second, the *pdfs* ρ^- and ρ^+ are taken into account and $z^T(x)$ is estimated from this matrix with a combination rule which depends on the mixture weights (see Theorem 1 in Section 3.2). Finally, the priors p^- and p^+ are taken into account (see Eq. 14 in Section 3.4). Note that some redundancy exists in the matrix given in Eq. 3. Therefore, it is possible to interrogate (and, in some circumstances, to store) fewer models than $\gamma^- \times \gamma^+$ (see Section 3.3). In order to derive our combination rule, we assume that the classifiers $z_{k,l}(x)$, used to compute the matrix given in Eq. 3, approximate Bayes' classifier.

3.1. Behavior of the target and base classifiers

When classifying an object x, Bayes' classifier determines the most probable class $\hat{y}(x) \in \{-1, +1\}$, given the knowledge of the attributes describing this object.

For binary classifiers, the decision rule is given by:

$$\hat{y}(x) = \text{sign}\left(P(y(x) = +1|x) - \frac{1}{2}\right). \quad (4)$$

Using Bayes' rule, we have

$$P(y(x) = +1|x) = \frac{p^+ \rho^+(x)}{p^- \rho^-(x) + p^+ \rho^+(x)}. \quad (5)$$

Assuming the independence of the objects to be classified, every machine learning method that aims at minimizing the error rate approximates Bayes' classifier. These methods estimate from the database, either explicitly or implicitly, the *pdfs* ρ^+ and ρ^-, as well as the priors. As the only information contained in the database about the priors is the proportion of samples of the various classes, we assume that real classifiers estimate the priors as $p^+ = n^+/(n^- + n^+)$ and $p^- = n^-/(n^- + n^+)$ where n^- $(\neq 0)$ and n^+ $(\neq 0)$ are the number of negative and positive samples in the database. Therefore, the learners compute an approximation $z(x)$ of $P(y(x) = +1|x)$:

$$z(x) = \frac{n^+ \rho^+(x)}{n^- \rho^-(x) + n^+ \rho^+(x)}. \quad (6)$$

In the following sections, we describe a domain adaptation technique that combines the outputs of several models and that compensates for their biases. In practice, we need to be able to detect when $z(x)$ is undefined. Therefore, we assume that $z(x)$ takes a special value (denoted $\star$) when $\rho^-(x) = \rho^+(x) = 0$. This happens when x is neither drawn from ρ^- nor ρ^+.

3.2. Base classifiers combination rule

Recall that $z_{k,l}(x)$ denotes the output of a binary classifier learned off-line from a database populated with $n^-_{k,l}$, $n^+_{k,l}$ samples taken randomly from ρ^-_k, ρ^+_l respectively. Eq. 6 becomes

$$z_{k,l}(x) = \frac{n^+_{k,l} \rho^+_l(x)}{n^-_{k,l} \rho^-_k(x) + n^+_{k,l} \rho^+_l(x)}. \quad (7)$$

The ratio of $\rho^+_l(x)$ and $\rho^-_k(x)$ plays a central role in the theory developed hereafter. For the sake of conciseness, we denote it by

$$R_{k,l}(x) = \frac{\rho^+_l(x)}{\rho^-_k(x)} = \frac{n^-_{k,l}}{n^+_{k,l}} \frac{z_{k,l}(x)}{1 - z_{k,l}(x)}. \quad (8)$$

Before establishing the combination function that allows to adapt the base classifiers to the target *pdfs* ρ^- and ρ^+, we have to underline that some care must be taken when manipulating the ratio $R_{k,l}(x)$. It is defined, finite, and strictly positive (and therefore invertible) as long as $\rho^-_k(x) \neq 0$ and $\rho^+_l(x) \neq 0$. Fortunately, considering only the mixture components such that x belongs to the support does not change our problem setting. Some care must also be taken for the mixture weights since nothing prevents them to be null when the characteristics of the target domain are difficult to predict. In the following, we consider the minimal subsets $\sigma^-(x)$ and $\sigma^+(x)$ of components that need to be taken into account in the mixtures without changing the problem. They are given by

$$\sigma^-(x) = \left\{k \in \left\{1, \ldots \gamma^-\right\} \,|\, \beta^-_k \neq 0 \wedge \rho^-_k(x) \neq 0\right\},$$
$$\sigma^+(x) = \left\{l \in \left\{1, \ldots \gamma^+\right\} \,|\, \beta^+_l \neq 0 \wedge \rho^+_l(x) \neq 0\right\},$$

since

$$\rho^-(x) = \sum_{k=1}^{\gamma^-} \beta^-_k \rho^-_k(x) = \sum_{k \in \sigma^-(x)} \beta^-_k \rho^-_k(x),$$

$$\rho^+(x) = \sum_{l=1}^{\gamma^+} \beta^+_l \rho^+_l(x) = \sum_{l \in \sigma^+(x)} \beta^+_l \rho^+_l(x).$$

The sets $\sigma^-(x)$ and $\sigma^+(x)$ can be determined on-the-fly based on the outputs $z_{k,l}(x)$ since Eq. 7 implies that, as soon as $z_{k,l}(x) \in \{1, \star\}$ for any l, then $\rho^-_k(x) = 0$. This, in turn, implies that $z_{k,l}(x) \in \{1, \star\}$ for every l. Similarly, if $z_{k,l}(x) \in \{0, \star\}$ for any k, then this implies that $\rho^+_l(x) = 0$, which implies, in turn, that $z_{k,l}(x) \in \{0, \star\}$ for every k. This reasoning shows how the sets can be determined on-the-fly with the outputs of the base classifiers. Therefore, to be able to determine the sets $\sigma^-(x)$ and $\sigma^+(x)$, at least one element should be determined in each row and in each column of the matrix given in Eq. 3. If an element is 1 or $\star$, then the component corresponding to the row should be discarded in $\sigma^-(x)$. Similarly, if an element is 0 or $\star$, then the component corresponding to the column should be discarded in $\sigma^+(x)$. It follows that one needs to interrogate only $\max(\gamma^-, \gamma^+)$ models in order to determine these sets.

Note that three particular cases may be encountered. If the set $\sigma^-(x)$ is empty, then the object x cannot be explained by any component of the negative mixture, and $y(x) = +1$. Similarly, $y(x) = -1$ when $\sigma^+(x)$ is empty. If both $\sigma^-(x)$ and $\sigma^+(x)$ are empty, then the target distribution is not correctly represented by the mixtures, and nothing can be said about the class of x. In the following, we suppose that $\sigma^-(x)$ and $\sigma^+(x)$ are nonempty sets.

Once $\sigma^-(x)$ and $\sigma^+(x)$ have been determined, the domain adaptation is achieved with one of the equations given in the next theorem.

Theorem 1. *If the values $z_{k,l}(x)$ are known for all $k \in [1, \gamma^-]$ and all $l \in [1, \gamma^+]$, then it is possible to guess the output $z(x)$ of a model that could be learned from a database populated with n^- samples drawn randomly from $\rho^- = \sum_{k=1}^{\gamma^-} \beta_k^- \rho_k^-$ and n^+ samples drawn at random from $\rho^+ = \sum_{l=1}^{\gamma^+} \beta_l^+ \rho_l^+$, even if this model is not available. From Eq. 2 and 5, it follows that the guessed value of the output should be*

$$z^T(x) = \frac{n^+ \sum_{l=1}^{\gamma^+} \beta_l^+ \rho_l^+(x)}{n^- \sum_{k=1}^{\gamma^-} \beta_k^- \rho_k^-(x) + n^+ \sum_{l=1}^{\gamma^+} \beta_l^+ \rho_l^+(x)}. \tag{9}$$

It can be expressed under multiple (mathematically equivalent) forms, for example

$$z^T(x) = \left(1 + \sum_{k \in \sigma^-(x)} \left(\sum_{l \in \sigma^+(x)} \frac{\beta_l^+}{\beta_k^-} \frac{n^+}{n^-} R_{k,l}(x) \right)^{-1} \right)^{-1}, \tag{10}$$

or

$$z^T(x) = \left(1 + R_{c^-,c^+}(x) \frac{\sum_{k \in \sigma^-(x)} \beta_k^- n^- R_{k,c^+}(x)^{-1}}{\sum_{l \in \sigma^+(x)} \beta_l^+ n^+ R_{c^-,l}(x)}\right)^{-1} \tag{11}$$

for any $c^- \in \sigma^-(x)$ and $c^+ \in \sigma^+(x)$.

As the purpose of this theorem is to simulate the behavior of a model learned from a virtual database, n^- and n^+ can be chosen arbitrarily and we get rid of the impact of this arbitrary choices in Section 3.4.

Theorem 1 is the main theoretical result of this paper. It shows that it is not necessary to learn a new model each time the mixture weights change. Therefore, the classifier can be adapted on-the-fly, depending on the context encountered in the target application. It should be stressed that the combination rules given in Eq. 10 and 11 are not linear. This stands in contrast with the combination rules (namely the *linear combining rule* and the *distribution weighted combining rule*) proposed, without any theoretical foundation, by Mansour *et al.* (Mansour et al., 2008).

For the sake of simplicity, let us consider the particular case where no mixture weight is zero and the values of $z_{k,l}(\cdot)$ are restricted to the range $(0, 1)$ $\forall k, l$. If the negative density has a sole component ($\gamma^- = 1$), then both Eq. 10 and 11 can be simplified and are reduced to the same expression:

$$z^T(x) = \frac{1}{1 + \frac{n^-}{n^+} \frac{1}{\sum_{l=1}^{\gamma^+} \beta_l^+ R_{1,l}(x)}}. \tag{12}$$

In a similar way, if $\gamma^+ = 1$, we also obtain an identical simplified expression for Eq. 10 and 11:

$$z^T(x) = \frac{1}{1 + \frac{n^-}{n^+} \sum_{k=1}^{\gamma^-} \beta_k^- \frac{1}{R_{k,1}(x)}}. \tag{13}$$

In the general case, multiple combination functions exist, as stated in Theorem 1. This is due to the fact that there exist known relationships between the elements of the matrix given in Eq. 3, as long as it has multiple rows and columns. Those relationships can be used to minimize the amount of classifiers to interrogate, and therefore the computational cost of the method. This is shown in the following section.

3.3. Minimizing the computational cost

From the computational point of view, there is a large difference between Eq. 10 and 11. In Eq. 10, $|\sigma^-(x)| \times |\sigma^+(x)| \leq \gamma^- \times \gamma^+$ models need to be interrogated. On the other hand, in Eq. 11, only $|\sigma^-(x)| + |\sigma^+(x)| - 1 \leq \gamma^- + \gamma^+ - 1$ models need to be interrogated, plus the $\max(\gamma^-, \gamma^+)$ models needed to determine $\sigma^-(x)$ and $\sigma^+(x)$. If reducing further the computational cost is necessary, dynamically determining the time budget available for questioning each model can be envisaged (see for example (Schwing et al., 2011)). Note that, with some a priori knowledge about $\sigma^-(\cdot)$ and $\sigma^+(\cdot)$ and some combination rules, it is possible to reduce the number of models to be learned. This is because the a priori knowledge one has about the classification problem can permit to deduce that some models will never be questioned.

The fact that $z(x)$ can be determined from a subset of models is explained by the following theorem.

Theorem 2. *Under the assumption that $z_{k,l} \in (0, 1)$ $\forall k, l$, the outputs of the $\gamma^- \times \gamma^+$ models are not independent when $\gamma^- > 1$ and $\gamma^+ > 1$.*

Fig. 1 shows three examples of dependencies. In Fig. 1(a), $R_{1,1}(x)\, R_{3,4}(x) = R_{3,1}(x)\, R_{1,4}(x)$. Therefore, if three values among $z_{3,1}(x)$, $z_{1,1}(x)$, $z_{3,4}(x)$, and $z_{1,4}(x)$ are known, then the fourth one can be determined. The same principle can be used to establish the dependency for the other examples.

As the outputs of the $\gamma^- \times \gamma^+$ models are not independent, it is not necessary to interrogate all of them. Fig. 2 shows three examples of sets of models that suffice. The combination rule in Eq. 10 corresponds to Fig. 2(a), whereas the one in Eq. 11 corresponds to Fig. 2(b) (the example is shown for $c^- = 2$ and $c^+ = 3$). Intuitively, we expect that there exist many more combination rules than the two ones given in Theorem 1. All are equivalent under the assumption that $z_{k,l}(x)$ is noise-free $\forall k, l$, but some of them might

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$
$z_{5,1}$	$z_{5,2}$	$z_{5,3}$	$z_{5,4}$

(a) example 1

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$
$z_{5,1}$	$z_{5,2}$	$z_{5,3}$	$z_{5,4}$

(b) example 2

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$
$z_{5,1}$	$z_{5,2}$	$z_{5,3}$	$z_{5,4}$

(c) example 3

Figure 1. The $\gamma^- \times \gamma^+$ models are highly dependent. This figure represents their outputs for $\gamma^- = 5$ and $\gamma^+ = 4$. Three sets of dependent predictions are highlighted in gray (it is assumed that $z_{k,l} \in (0, 1)$). Each highlighted value can be determined from the other ones.

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$
$z_{5,1}$	$z_{5,2}$	$z_{5,3}$	$z_{5,4}$

(a) example 1

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$
$z_{5,1}$	$z_{5,2}$	$z_{5,3}$	$z_{5,4}$

(b) example 2

$z_{1,1}$	$z_{1,2}$	$z_{1,3}$	$z_{1,4}$
$z_{2,1}$	$z_{2,2}$	$z_{2,3}$	$z_{2,4}$
$z_{3,1}$	$z_{3,2}$	$z_{3,3}$	$z_{3,4}$
$z_{4,1}$	$z_{4,2}$	$z_{4,3}$	$z_{4,4}$
$z_{5,1}$	$z_{5,2}$	$z_{5,3}$	$z_{5,4}$

(c) example 3

Figure 2. This figure represents the outputs of the $\gamma^- \times \gamma^+$ models for $\gamma^- = 5$ and $\gamma^+ = 4$. All values can be determined from the highlighted ones (it is assumed that $z_{k,l} \in (0, 1)$).

be preferable in case of noisy predictions.

3.4. Adaptation to the priors

When there is some class overlapping (*i.e.* $\rho^-(x) > 0$ and $\rho^+(x) > 0$) and $p^+ \neq n^+/(n^- + n^+)$, the classifier $\hat{y}(x) = \text{sign}\left(z(x) - \frac{1}{2}\right)$ is biased since $z(x) \neq P(y(x) = +1|x)$ (*cf.* Eq. 5 and 6). Since we assume p^+ is unknown, one cannot choose n^+ and n^- such that $p^+ = n^+/(n^- + n^+)$ when the learning database is populated. In order to compensate for the bias introduced by the proportion of samples in the learning database, we need to establish the relation between $z(x)$ and $P(y(x) = +1|x)$. Starting from Eq. 5 and 6, and after some algebraic calculus, we find

$$P(y(x) = +1|x) = \frac{n^- p^+ z(x)}{n^+ p^- + (n^- p^+ - n^+ p^-) z(x)}. \quad (14)$$

Similar correction functions are known in the literature for various classifiers (Chan & Ng, 2005; Elkan, 2001; Weiss & Provost, 2003). This result indicates how the classifier can be adapted on-the-fly (that is without learning another model) when the priors change over time, when they are unknown at learning time, or when it is impossible to respect the priors when the learning set is collected. Thus, Eq. 14 allows to concentrate on the representation of the underlying *pdfs* instead of on the balance of positive and negative samples when the learning database is populated. Note that a classifier thresholding $P(y(x) = +1|x)$ and a classifier thresholding $z(x)$ share the same *Receiver Operating Characteristic* (ROC) curve. Nevertheless, the correction proposed in Eq. 14 is valuable in case where an accurate probability estimate is needed.

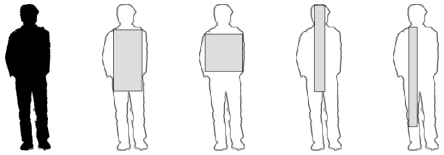

Figure 3. A silhouette s and a few largest axis-aligned rectangles included in it (*i.e.* a few elements of $\Psi(s)$).

4. Experiments

To evaluate the effectiveness of the proposed approach, we consider a classifier that differentiates between human and non-human silhouettes. These ones are obtained either by segmenting images, or by applying a background subtraction algorithm on video streams and separating the blobs with a connected components analysis. The methods that consider the silhouette as a whole and perform a single classification are, in general, highly sensitive to huge defects. In contrast, part-based methods split the silhouettes in a set of smaller regions to decrease the influence of defects. For example, Barnich *et al.* (Barnich et al., 2006) proposed to rely on regions that do not have any anatomical meaning: the largest axis-aligned rectangles (*i.e.* those whose borders are parallel to the image borders) that can be wedged into the silhouette (see Fig. 3).

In the following, we consider an approach similar to that of Barnich *et al.*, but take advantage of the results presented in this paper and tune the classifier in real-time to compensate for the biases with respect to the context. Conventionally, -1 represents the non-human class, and $+1$ represents the human class. The silhouette classification is organized in two stages.

A low level classifier predicts for each rectangle r its probability to be issued from a human silhouette $P(y(r) = +1|r)$. We use the *ExtRaTrees* (Geurts et al., 2006) with 100 trees per model, and consider the proportion of trees voting for the positive class as a good approximation of $z(r)$. In the learning (test) set, 50 (100) rectangles are selected randomly in each silhouette. The rectangles are described by scale-invariant and size-invariant attributes. We use the horizontal and vertical positions of the rectangle (defined *w.r.t.* the position of the silhouette), and the horizontal and vertical sizes of the rectangle (defined *w.r.t.* the size of the silhouette). Since they are weak features, the two classes (that is the rectangles issued from a human silhouette and those issued from a non-human silhouette) are highly overlapping in the space of rectangles. Adapting the output of the low level classifier to the priors, as presented in Section 3.4, is therefore mandatory. Moreover, our domain adaptation technique is also applied in this low level stage.

In the high level stage, the silhouettes are fed to a granulometric filter computing the set of all the largest rectangles that can be wedged inside and that are aligned with the image axes. Only a random subset $\Psi(s)$ of 100 rectangles is used to predict the class (human or non-human) of the silhouette s:

$$\hat{y}(s) = \text{sign}\left(\sum_{r\in\Psi(s)}\left(P(y(r) = +1|r) - \frac{1}{2}\right)\right). \quad (15)$$

For to ease of interpretation, the results presented in Section 4.2 are related to this high level classifier.

4.1. Experimental setup

In our experiments, we consider an application in which people can perform $\gamma^+ = 2$ activities. The realistic poses in the first one are those of a walker, whereas the realistic poses in the second activity have a higher variability. These two sets of poses are named *strongly constrained poses*, and *weakly constrained poses*, respectively. The largest axis-aligned rectangles that can be wedged into strongly constrained silhouettes follow ρ_1^+ and those issued from weakly constrained silhouettes follow ρ_2^+. We suppose that the probabilities associated to each activity (that is β_1^+ and β_2^+) are unknown at learning time, but can be estimated at runtime. We consider only $\gamma^- = 1$ component for the non-human objects, which are drawn from ρ_1^-.

(a) Human silhouettes corresponding to strongly constrained poses ($S_1^+ \sim \rho_1^+$).

(b) Human silhouettes corresponding to weakly constrained poses ($S_2^+ \sim \rho_2^+$).

(c) Non human silhouettes ($S_1^- \sim \rho_1^-$).

Figure 4. Subsets of the database used in our experiments.

The dataset used in our experiments, is depicted in Fig. 4. We have $15,000$ silhouettes ($5,000$ non-human, $5,000$ strongly constrained, and $5,000$ weakly constrained) for learning and $15,000$ silhouettes for the assessment. Each model is learned with $5,000$ non-human silhouettes and $5,000$ human silhouettes.

4.2. Results

Fig. 5 compares the results obtained with four methods, in various contexts. The context is specified by $p^+ = 1 - p^-$ and $\beta_1^+ = 1 - \beta_2^+$.

The most naive domain adaptation method consists in using a single model for all contexts. Let us denote $\mathcal{M}_\beta$ a model learned from a training set populated with 5000 non-human silhouettes, 5000β strongly, and $5000(1-\beta)$ weakly constrained human silhouettes. Note that β is the proportion of strongly constrained poses in the learning set, not to be confused with β_1^+ which is the proportion of strongly constrained poses encountered in the target application. We have tried three models: $\mathcal{M}_{0.0}$, $\mathcal{M}_{0.5}$, and $\mathcal{M}_{1.0}$. As expected, $\mathcal{M}_{0.0}$ behaves better when there are only weakly constrained poses in the application, and $\mathcal{M}_{1.0}$ behaves better when there are only strongly constrained poses in the application. Otherwise, the *pdf* associated to the class $+1$ is not correctly represented in the learning set. In Fig. 5, the color (thin) lines correspond to $\mathcal{M}_{0.0}$, $\mathcal{M}_{0.5}$, and $\mathcal{M}_{1.0}$. It turns out that these models fail to generalize: $\mathcal{M}_{0.0}$ performs poorly when β_1^+

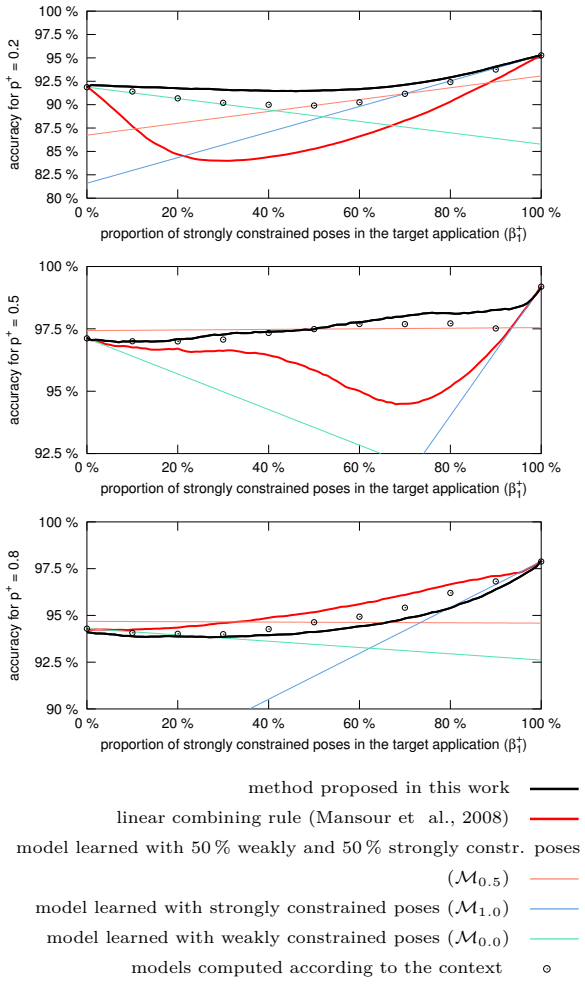

Figure 5. Results of our experiments. The graphs show the accuracy with respect to the proportion β_1^+ of strongly constrained poses encountered at runtime when the probability p^+ to observe a human silhouette in the target context is 0.2, 0.5, and 0.8. The black curve corresponds to the method proposed in this work. It is obtained using Theorem 1 and Eq. 14. The other curves take only advantage of Eq. 14. Our dynamically configurable classifier outperforms the classifiers fixed at learning time.

is large, $\mathcal{M}_{1.0}$ performs poorly when β_1^+ is small, and $\mathcal{M}_{0.5}$ is unable to reach optimal performance when β_1^+ is 0 % or 100 %. This implies that some domain adaptation is needed.

Another naive domain adaptation method consists in storing a lot of models and selecting, at runtime, the model that best suits the context. When the proportion of strongly constrained poses is $\beta_1^+ \simeq \beta$, we select $\mathcal{M}_\beta$. The eleven circular markers in Fig. 5 correspond to $\mathcal{M}_{0.0}$, $\mathcal{M}_{0.1}$, $\mathcal{M}_{0.2}$, ... $\mathcal{M}_{1.0}$. As the conditions in which a model is used correspond to the assumption made when populating the corresponding training set, the results are optimal and outperform those obtained with the first method. However, storing a large amount of models and switching quickly between them is impossible in practice.

The aim of the method proposed in this work is to approximate the results that would be obtained with a high number of models in the second method, but only $\gamma^+ = 2$ models have to be stored. For each rectangle $r \in \Psi(s)$, the model learned only with strongly constrained poses is used to compute $z_{1,1}(r)$, and the one learned only with weakly constrained poses is used to compute $z_{1,2}(r)$. The probability β_1^+ to observe a strongly constrained pose is taken into account when we compute $z(r)$ with Theorem 1. Then, $P(y(r) = +1|r)$ is computed using Eq. 14. Finally, the class of the silhouette is determined using Eq. 15. The results obtained with this dynamic method are represented by the black (thick) curves in Fig. 5. It can be observed that our method provides a good approximation of the second one when $p^+ = 0.5$ or $p^+ = 0.8$ (which was expected), but outperforms the second method when $p^+ = 0.2$. It is to be noted that the results obtained with our combination rule and only two models are similar to those obtained with ten separate models learned in different contexts. Our method thus gives a way to save the learning and storage of eight models.

Fig. 5 also shows the results that are obtained with the linear combining rule proposed in (Mansour et al., 2008): $z^T(r) = \beta_1^+ z_{1,1}(r) + \beta_2^+ z_{1,2}(r)$. Mansour's method slightly outperforms ours when the proportion of strongly constrained poses in the target application is very high, but is strongly outperformed by ours when this proportion is weak to neutral.

5. Conclusion

In this paper, we have considered a binary classification task for which the priors (p^- and p^+), the *pdf* ρ^- associated to the negative class, and the *pdf* ρ^+ associated to the positive class are unknown at learning time, but known at runtime. In this domain adaptation problem, the *pdf* ρ^- (ρ^+) is a mixture with γ^- (γ^+) components parametrized by γ^- (γ^+) degrees of freedom (the mixture weights). Our solution combines and adapts the outputs of $\gamma^- \times \gamma^+$ models learned off-line. We have presented a theoretical solution to adapt, on-the-fly, the output of the classifier according to the context in which it is used, without retraining any classifier. Our approach has been validated on a real computer vision application.

References

Barandela, R., Sánchez, J., García, V., & Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition*, *36*, 849–851.

Barnich, O., Jodogne, S., & Van Droogenbroeck, M. (2006). Robust analysis of silhouettes by morphological size distributions. *Advances Concepts for Intelligent Vision Systems (ACIVS)* (pp. 734–745). Springer.

Batista, G., Prati, R., & Monard, M. (2004). A study of the behavior of several methods for balancing machine learning training data. *Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), special issue on learning from imbalanced datasets*, *6*, 20–29.

Chan, P., & Stolfo, S. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. *International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 164–168). New York, USA: AAAI Press.

Chan, Y., & Ng, H. (2005). Word sense disambiguation with distribution estimation. *19th international joint conference on Artificial intelligence (IJCAI)* (pp. 1010–1015). Edinburgh, UK.

Daumé III, H., & Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, *26*, 101–126.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*, 1–38.

Elkan, C. (2001). The foundations of cost-sensitive learning. *17th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 973–978). Seattle, USA.

Ge, L., Gao, J., & Zhang, A. (2013). OMS-TL: A framework of online multiple source transfer learning. *22nd ACM international conference on Conference on information & knowledge management (CIKM)* (pp. 2423–2428). San Francisco, USA.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3–42.

Jain, V., & Learned-Miller, E. (2011). Online domain adaptation of a pre-trained cascade of classifiers. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 577–584). Colorado Springs.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*, 429–449.

Lin, Y., Lee, Y., & Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning*, *46*, 191–202.

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2008). Domain adaptation with multiple sources. *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada.

Pan, S., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge And Data Engineering*, *22*, 1345–1359.

Schwing, A., Zach, C., Zheng, Y., & Pollefeys, M. (2011). Adaptive random forest – how many "experts" to ask before making a decision? *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1377–1384). Colorado Springs.

Storkey, A., & Sugiyama, M. (2006). Mixture regression for covariate shift. *Advances in Neural Information Processing Systems (NIPS)* (pp. 1337–1344). Vancouver, Canada.

Weiss, G., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, *19*, 315–354.

Zhao, P., & Hoi, S. (2010). OTL: A framework of online transfer learning. *International Conference on Machine Learning (ICML)* (pp. 1231–1238). Haifa, Israel.