Semi-Supervised Boosting using Visual Similarity Learning*

Christian Leistner Helmut Grabner Horst Bischof
Graz University of Technology
Institute for Computer Graphics and Vision
{leistner, hgrabner, bischof}@icg.tugraz.at

Abstract

The required amount of labeled training data for object detection and classification is a major drawback of current methods. Combining labeled and unlabeled data via semi-supervised learning holds the promise to ease the tedious and time consuming labeling effort. This paper presents a novel semi-supervised learning method which combines the power of learned similarity functions and classifiers. The approach capable of exploiting both labeled and unlabeled data is formulated in a boosting framework. One classifier (the learned similarity) serves as a prior which is steadily improved via training a second classifier on labeled and unlabeled samples. We demonstrate the approach on challenging computer vision applications. First, we show how we can train a classifier using only a few labeled samples and many unlabeled data. Second, we improve (specialize) a state-of-the-art detector by using labeled and unlabeled data.

1. Introduction

In recent years, there was significant progress on methods for visual object recognition and categorization. For example, the performance on the Caltech 101 dataset was in 2004 approximately 16%, now the best performing approaches obtain close to 70% [11]. Besides novel methods for local image representations, there was a significant progress in using advanced machine learning methods (*e.g.*, Boosting [9], support vector machines [25]). Further, if enough labeled training data exists these approaches can obtain very high recognition performances (*e.g.*, [26]). However, for most practical problems (with many classes and high variability within the classes) there is simply not enough labeled data available, whereas hand-labeling is tedious and expensive, in some cases not even feasible.

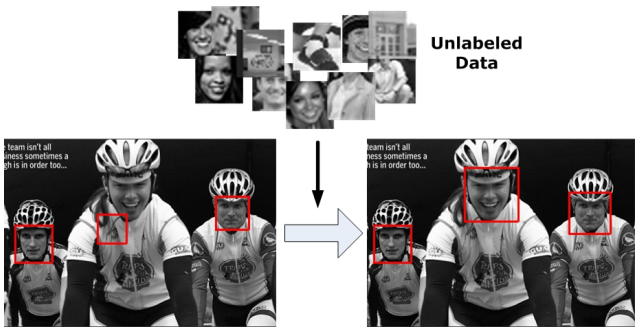


Figure 1. A trained classifier/detector can be substantially improved (increase detection rate, reduce false positive rate and better alignment of detections) given additionally unlabeled data.

The lack of sufficient labeled training data is the reason for the recent attention towards unsupervised and semi-supervised training algorithms. The key-idea of semi-supervised learning is to exploit labeled samples as well as a large number of unlabeled samples for obtaining an accurate decision border (see Zhu [28] for a recent overview of approaches). This differs from the conventional "missing data" problem in that the size of the unlabeled data exceeds that of the labeled by far. The central issue of semi-supervised learning is how to exploit this huge amount of information. Hence, a number of different algorithms have been proposed, *e.g.*, transductive support vector machines [4], graph-based semi-supervised learning [3, 23, 29], semi-supervised linear discriminant analysis [6], discriminative-generative methods [1], and even self-taught semi-supervised learning [21]. Very recently Mallapragada *et al*. [19] proposed a semi-supervised boosting method which outperforms other approaches on standard machine learning benchmark problems.

In computer vision, Cohen *et al*. [7] use both labeled and unlabeled data to improve on face detectors. In [27] a semi-supervised approach for detecting objects in aerial images has been developed. Various methods [13, 15, 24] have been

*This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04, the FFG project EVis under the FIT-IT program and the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

used in the context of image retrieval. Also related but only inspired by semi-supervised learning is the work of Li *et al*. [18] which presents an incremental approach to learn object categories using internet search as an additional information.

A fundamental requirement for many semi-supervised learning approaches is the need for a similarity measure in order to measure the distance between samples (labeled and unlabeled) in feature space. Learning distance/similarity functions is a closely related strand of research which has received considerable attention in machine learning. There is large amount of work on defining image-to-kernel functions, *e.g*., [12, 16] for image comparison. Yet, it is not always possible to define such "good" measures, *e.g*., in object categorization where one deals with a high intra-class variance and low interclass variance. Therefore, *e.g*., Nowak *et al*. [20] investigated learning these measures. Hertz *et al*. [14] proposed a large margin boosting formulation for learning distance functions to be used in clustering. Very recently Frome *et al*. [11] have proposed a large margin formulation for learning globally consistent local distance functions.

The main contribution of this paper is to combine similarity learning and semi-supervised boosting and thereby enabling the application on challenging computer vision tasks. Our work is based on SemiBoost proposed in [19] which, however, is limited to fixed similarities. In contrast, this paper demonstrates that also the similarity measure can be learned leading to a very flexible versatile method. Besides illustrating results on artificial data we demonstrate the algorithm on two tasks. First, we show how to train a state-of-the art classifier from a few labeled examples and many unlabeled samples. Second, we demonstrate that the novel semi-supervised boosting method can be used to improve a state-of-the-art detector using unlabeled samples.

The rest of the paper is organized as follows. In Sec. 2 we review Boosting on labeled and unlabeled data. Sec. 3 brings together learning of visual similarities and SemiBoost. Experiments on computer vision applications are shown in Sec. 4 and, finally, Sec. 5 concludes the paper.

2. Boosting on labeled and unlabeled data

2.1. Supervised Boosting

In supervised learning one deals with a labeled dataset $\mathcal{D}^L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|\mathcal{D}^L|}, y_{|\mathcal{D}^L|})\} \subseteq \mathcal{X} \times \mathcal{Y}$ where $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^M$ and $y_i \in \mathcal{Y}$. In this paper, we focus on the binary classification problem, therefore $\mathcal{Y} = \{+1, -1\}$ and the samples are split into two sets $\mathcal{X}^L = \mathcal{X}^+ \cup \mathcal{X}^-$ of all samples with a positive class and the set of all samples with negative class, respectively. Then, a classifier $H : \mathcal{X} \to \mathcal{Y}$ is trained using the labeled samples.

Boosting[1] in general converts a weak learning algorithm into a strong one [9]. A strong classifier $H_N(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n h_n(\mathbf{x})$ is a linear combination of N weak classifiers $h_n(\mathbf{x})$ which have only to be slightly better than random guessing. The weak classifiers are trained using a weighted training set $\mathcal{D}^L$. This is done by adaptive logistic regression (Friedman *et al*. [10]). Boosting minimizes an exponential loss function on the training data

$$\mathcal{L}_{\mathcal{D}^L} = \sum_{\mathbf{x} \in \mathcal{D}^L} \mathcal{L}(\mathbf{x}, y) = \sum_{\mathbf{x} \in \mathcal{D}^L} e^{-yH(\mathbf{x})}. \tag{1}$$

It is easy to show [10] that $\mathbb{E}(e^{-yH(\mathbf{x})})$, where $\mathbb{E}(\cdot)$ is the expectation operator, is minimized by AdaBoost and hence

$$P(y = 1|\mathbf{x}) = \frac{e^{H(\mathbf{x})}}{e^{H(\mathbf{x})} + e^{-H(\mathbf{x})}}. \tag{2}$$

2.2. Semi-Supervised Boosting

Unsupervised methods aim to find an interesting (natural) structure in $\mathcal{X}$ using only unlabeled data $\mathcal{D}^U = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{D}^U|}\} \subseteq \mathcal{X}$. Semi-supervised learning uses both labeled $\mathcal{D}^L$ and unlabeled $\mathcal{D}^U$ data. There exist different methods how the unlabeled data can be included in the learning process [28]. In this paper, we focus on inductive learning, where in addition to assigning labels to the unlabeled samples, also a classifier is provided.

d'Alche–Buc *et al*. [8] were the first to extend boosting to semi-supervised learning by using a semi-supervised learning algorithm as a weak classifier. Hertz *et al*. [14] proposed to include unlabeled data as prior for the weak classifiers. Bennett *et al*. [5] extend the loss function

$$\mathcal{L}_{\mathcal{D}^L \cup \mathcal{D}^U} = \sum_{\mathbf{x} \in \mathcal{D}^L} e^{-yH(\mathbf{x})} + C \sum_{\mathbf{x} \in \mathcal{D}^U} e^{-|H(\mathbf{x})|} \tag{3}$$

in order to take the unlabeled data into account ($C \geq 0$ is a constant introduced to weight the importance between the labeled and the unlabeled data). In these approaches, the unlabeled data is used to regularize the decision boundary where the boundary which passes through a region with low density of unlabeled examples is preferred over heavily popularized regions in feature space.

2.3. SemiBoost

In contrast to the methods mentioned above, this work combines ideas from graph theory and clustering. Thus, we build on SemiBoost [19] which, additionally, guides the learning process using pairwise similarities. Depending on how the samples are provided, three different loss functions are defined which are then additively combined. The goal is to use boosting in order to minimize the combined loss.

[1] This paper solely focuses on the discrete version of AdaBoost.

Labeled Samples
As in "standard" boosting, we use the exponential loss for samples $\mathbf{x}_i \in \mathcal{X}^L$ with correct label y_i as

$$\mathcal{L}^L(\mathbf{x}_i, y_i) := e^{-2y_i H(\mathbf{x}_i)}. \quad (4)$$

Note, the factor 2 in the exponential function is used to simplify the notation, but of course, does not change the minimum.

Pair of Labeled and Unlabeled Samples
Given a sample $\mathbf{x}_i \in \mathcal{X}^L$ labeled with y_i and a second unlabeled sample $\mathbf{x}_j \in \mathcal{X}^U$. We define the loss between labeled and unlabeled examples as

$$\mathcal{L}^{LU}(\mathbf{x}_i, y_i, \mathbf{x}_j) := S(\mathbf{x}_i, \mathbf{x}_j) e^{-2H(\mathbf{x}_j) y_i}, \quad (5)$$

where $S(\mathbf{x}_i, \mathbf{x}_j)$ is a similarity measure of the two examples. The intuition behind this is, that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are very similar also the labels should be the same.

Pair of Two Unlabeled Samples
Given two unlabeled samples $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}^U$, we define a loss which forces an agreement if the samples are similar. This is done by defining

$$\mathcal{L}^{UU}(\mathbf{x}_i, \mathbf{x}_j) := S(\mathbf{x}_i, \mathbf{x}_j) \cosh(H(\mathbf{x}_i) - H(\mathbf{x}_j)). \quad (6)$$

Since $\cosh(x) \geq 1$ is a symmetric function which has its minimum at $x = 0$ it measures the agreement of the two classifier responses. For simplicity we assume that $S(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric[2] and thus we can rewrite Eq. (6) as

$$\mathcal{L}^{UU}(\mathbf{x}_i, \mathbf{x}_j) := \frac{1}{2} S(\mathbf{x}_i, \mathbf{x}_j) \left[e^{H(\mathbf{x}_i) - H(\mathbf{x}_j)} + e^{H(\mathbf{x}_j) - H(\mathbf{x}_i)} \right]. \quad (7)$$

The Combined Loss
By summing over the labeled and unlabeled examples, respectively, we define a combined objective function.

$$\begin{aligned}\mathcal{L} = & \tfrac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x} \in \mathcal{X}^L} e^{-2yH(\mathbf{x})} + \\ & + \tfrac{1}{|\mathcal{X}^L||\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^L} \sum_{\mathbf{x}_j \in \mathcal{X}^U} S(\mathbf{x}_i, \mathbf{x}_j) e^{-2y_i H(\mathbf{x}_j)} + \\ & + \tfrac{1}{|\mathcal{X}^U||\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^U} \sum_{\mathbf{x}_j \in \mathcal{X}^U} S(\mathbf{x}_i, \mathbf{x}_j) e^{H(\mathbf{x}_i) - H(\mathbf{x}_j)}. \quad (8)\end{aligned}$$

Note, instead of using the size of the training sets for normalization one can also use other weighting terms emphasizing different criteria (*e.g.*, labeld data). Following the derivation of the AdaBoost algorithm on labeled examples we can optimize this objective function in a greedy manner by splitting off the n-th weak classifier. Thus, we solve the optimization problem by looking for the best weak classifier h_n and weight α_n, which are added to the ensemble:

$$(\alpha_n, h_n) = \arg \min_{\alpha_n, h_n} (\mathcal{L}) \quad (9)$$

Due to the limited space we solely show three important steps of the whole derivation. Rewriting the loss defined in Eq. (8) by taking the results from [10] and [19] into account, we get

$$\begin{aligned}\mathcal{L} \leq & \tfrac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x} \in \mathcal{X}^L} w_n(\mathbf{x}, y) e^{-2y\alpha_n h_n(\mathbf{x})} + \\ & + \tfrac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x} \in \mathcal{X}^U} \left[p_n(\mathbf{x}) e^{-\alpha_n h_n(\mathbf{x})} + q_n(\mathbf{x}) e^{\alpha_n h_n(\mathbf{x})} \right] \quad (10)\end{aligned}$$

where the term $w_n(\mathbf{x}, y) = e^{-2yH_{n-1}(\mathbf{x})}$, is the weight of a labeled sample and the terms

$$\begin{aligned}p_n(\mathbf{x}) = & e^{-2H_{n-1}(\mathbf{x})} \tfrac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x}_i \in \mathcal{X}^+} S(\mathbf{x}, \mathbf{x}_i) + \quad (11) \\ & + \tfrac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^U} S(\mathbf{x}, \mathbf{x}_i) e^{H_{n-1}(\mathbf{x}_i) - H_{n-1}(\mathbf{x})}, \\ q_n(\mathbf{x}) = & e^{2H_{n-1}(\mathbf{x})} \tfrac{1}{|\mathcal{X}^L|} \sum_{\mathbf{x}_i \in \mathcal{X}^-} S(\mathbf{x}, \mathbf{x}_i) + \quad (12) \\ & + \tfrac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x}_i \in \mathcal{X}^U} S(\mathbf{x}, \mathbf{x}_i) e^{H_{n-1}(\mathbf{x}) - H_{n-1}(\mathbf{x}_i)}\end{aligned}$$

can be interpreted as confidences of an unlabeled sample belonging to the positive ($p_n(\mathbf{x})$) and negative class ($q_n(\mathbf{x})$), respectively. This can be upper bounded by

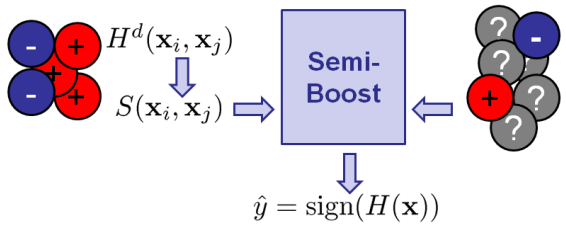
$$\begin{aligned}\mathcal{L} \leq & \tfrac{1}{|\mathcal{X}^L|} (e^{\alpha_n} - e^{-\alpha_n}) \textstyle\sum_{\substack{\mathbf{x} \in \mathcal{X}^L \\ h_n(\mathbf{x}) \neq y}} e^{-2yh_n(\mathbf{x})} + \\ & + \tfrac{1}{|\mathcal{X}^L|} e^{-\alpha_n} \sum_{\mathbf{x} \in \mathcal{X}^L} w_n(\mathbf{x}, y) + \\ & + \tfrac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x} \in \mathcal{X}^U} (p_n(\mathbf{x}) + q_n(\mathbf{x}))(e^{-\alpha_n h_n(\mathbf{x})} + e^{\alpha_n h_n(\mathbf{x})} - 1) + \\ & - \tfrac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x} \in \mathcal{X}^U} (p_n(\mathbf{x}) - q_n(\mathbf{x}_j)) \alpha_n h_n(\mathbf{x}) \quad (13)\end{aligned}$$

in order to make it better suitable for a boosting algorithm. The solution can be obtained in two steps: (i) find the weak classifier $h_n(\mathbf{x})$ and (ii) the corresponding weight α_n. When minimizing with respect to $h_n(\mathbf{x})$ this is equivalent to minimizing Eq. (14), because the other terms do not affect the location of the minima. The classifier is trained in order to minimize the weighted error of the samples. For a labeled sample $\mathbf{x} \in \mathcal{X}^L$ this is equal to standard boosting using the weight $w_n(\mathbf{x})$. The second term of Eq.(14) considers the unlabeled samples. In order to minimize it, the unlabeled sample $\mathbf{x} \in \mathcal{X}^U$ should be assigned the (pseudo)-label

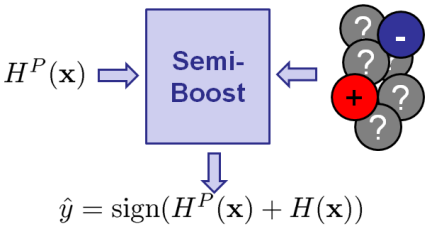
[2] In general the approach works also with asymmetric similarities (*e.g.*, KL-divergence) [19].

$$h_n(\mathbf{x}) = \arg\min_{h_n} \left(\frac{1}{|\mathcal{X}^L|} \sum_{\substack{\mathbf{x}\in\mathcal{X}^L \\ h_n(\mathbf{x})\neq y}} w_n(\mathbf{x},y) - \frac{1}{|\mathcal{X}^U|} \sum_{\mathbf{x}\in\mathcal{X}^U} (p_n(\mathbf{x}) - q_n(\mathbf{x}))\alpha_n h_n(\mathbf{x}) \right) \tag{14}$$

$$\alpha_n = \frac{1}{4}\ln\left(\frac{\frac{1}{|\mathcal{X}^U|}\left(\sum_{\substack{\mathbf{x}\in\mathcal{X}^U \\ h_n(\mathbf{x})=1}} p_n(\mathbf{x}) + \sum_{\mathbf{x}\in\mathcal{X}^U \wedge h_n(\mathbf{x})=-1} q_n(\mathbf{x}) \right) + \frac{1}{|\mathcal{X}^L|} \sum_{\substack{\mathbf{x}\in\mathcal{X}^L \\ h_n(\mathbf{x})=y}} w_n(\mathbf{x},y)}{\frac{1}{|\mathcal{X}^U|}\left(\sum_{\substack{\mathbf{x}\in\mathcal{X}^U \\ h_n(\mathbf{x})=1}} q_n(\mathbf{x}) + \sum_{\substack{\mathbf{x}\in\mathcal{X}^U \\ h_n(\mathbf{x})=-1}} p_n(\mathbf{x}) \right) + \frac{1}{|\mathcal{X}^L|} \sum_{\substack{\mathbf{x}\in\mathcal{X}^L \\ h_n(\mathbf{x})\neq y}} w_n(\mathbf{x},y)} \right) \tag{15}$$



(a) learning similarity



(b) improving classifier

Figure 2. SemiBoost combined with a learned similarity measure from given labeled samples (a) and an improvement with new data (b).

$z_n(\mathbf{x}) = \text{sign}(p_n(\mathbf{x}) - q_n(\mathbf{x}))$ and should be sampled according the confidence weight $|p_n(\mathbf{x}) - q_n(\mathbf{x})|$. The weight α_n is obtained by taking the derivative of Eq. (10) with respect to α_n and setting it to zero. The minimum is found as shown in Eq. (15).

Summarizing, the algorithm minimizes an objective function which takes labeled and unlabeled data into account using the similarity between samples. When no unlabeled data is used (*i.e.*, $\mathcal{X}^U = \{\}$) Eq. (14) and (15) reduce to the well known AdaBoost formulas.

3. SemiBoost with Learned Visual Similarities

SemiBoost has the power to exploit both labeled and unlabeled samples, if a similarity measure $S(\mathbf{x}_i,\mathbf{x}_j)$ is given. In this section, we first focus on how this similarity can be obtained for images. In the second part we consider a special case and derive a classifier improving strategy inspired by the SemiBoost algorithm. An overview of the two approaches is depicted in Fig. 2.

3.1. Learning the Visual Similarity

In principle, any given similarity measure can be used for $S(\mathbf{x}_i,\mathbf{x}_j)$. However, to be more flexible we propose to learn the similarity (see the brief overview in the Introduction) following the approach in [14] for learning distance functions. A distance function is a function of pairs of data points to the positive real numbers, usually (but not necessarily) symmetric with respect to its arguments. We define the learning problem on the product space of the input as $H^d : \mathcal{X}\times\mathcal{X} \to \mathcal{Y} = [-1\ 1]$. To train a maximum margin classifier the training set

$$\begin{aligned}\mathcal{D}^d = {} & \{(\mathbf{x}_i,\mathbf{x}_j,+1)|y_i = y_j, \mathbf{x}_i,\mathbf{x}_j \in \mathcal{D}^L\} \cup \\ & \cup\{(\mathbf{x}_i,\mathbf{x}_j,-1)|y_i \neq y_j, \mathbf{x}_i,\mathbf{x}_j \in \mathcal{D}^L\}\end{aligned} \tag{16}$$

is built by taking pairs of images of "same" and "different" class. Using pairs allows us to create a large number of training samples while having only a few labeled starting samples. The symmetry of the distance is not satisfied automatically, therefore it has to be enforced by introducing each pair twice, *i.e.*, both $(\mathbf{x}_i,\mathbf{x}_j)$ and $(\mathbf{x}_j,\mathbf{x}_i)$. Then, as in [14] we use boosting to learn a classifier. The trained and normalized classifier $H^d(\mathbf{x}_i,\mathbf{x}_j) \in [-1\ 1]$ is interpreted as a distance $d(\mathbf{x}_i,\mathbf{x}_j) = 1 - \frac{1}{2}(H^d(\mathbf{x}_i,\mathbf{x}_j)+1)$. Furthermore, this can then be converted to a similarity measure, *e.g.*, by a radial basis function

$$S(\mathbf{x}_i,\mathbf{x}_j) = e^{\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)^2}{\sigma^2}\right)}, \tag{17}$$

where σ^2 is the scale parameter.

3.2. Similarity as Prior Classifier

Let us consider the case that we have given a prior classifier $H^P(\mathbf{x})$ which can already (partially) solve our problem. We show that we can approximate the similarity $S(\mathbf{x}_i,\mathbf{x}_j)$ using this prior. First, we show how the training can be done. Second, for evaluation we can use the prior by combining it with the (newly) trained classifier. Thereby, we benefit from the information which is already encoded in the prior classifier. Roughly speaking, the newly trained classifier can be rather "small", only correcting the mistakes of $H^P(\mathbf{x})$.

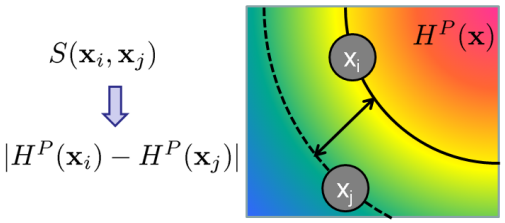


Figure 3. The similarity between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is approximated by the difference of the responses from an a-priori given classifier $H^P(\cdot)$.

Training

We assume that we have access to a prior classifier $H^P(\mathbf{x}) \in [-1 \quad 1]$ (*e.g.*, an already trained face detector). The classifier has to provide a confidence measure of its classification. The more confident the decision is the higher the absolute value of the response (*e.g.*, boosting can be used to train such a classifier and the responds can be translated into a probability using Eq. (2)). Thus, we define as distance measure

$$d(\mathbf{x}_i, \mathbf{x}_j) = |H^P(\mathbf{x}_i) - H^P(\mathbf{x}_j)| \tag{18}$$

as the absolute difference of the classifier response to the decision boundary. In other words, samples are similar if they have similar classifier response. The principle is visualized in Fig. 3. The distance is converted to a similarity using Eq. (17) as described in the previous subsection. Now, we are able to proceed training on the proposed SemiBoost manner.

Classifier Combination

If we train a SemiBoost classifier $H(\mathbf{x})$ using the prior classifier $H^P(\mathbf{x})$ as similarity measure, it makes sense to use this prior knowledge for the final classification process as well (*i.e.*, combine the two classifiers). This is closely related to the approach proposed by Schapire *et al.* [22]. Similarly, we use the prior knowledge as the 0^{th} weak classifier $h_0(\mathbf{x}) = \sigma^{-1}(P^P(y = 1|\mathbf{x}))$ where $P^P(y = 1|\mathbf{x})$ is the a-priori probability of the sample corresponding to the positive class and $\sigma^{-1}(\cdot)$ is the inverse function of our logistic model (see Eq. (2)). Since we use boosting to train the prior classifier, we end up with $h_0(\mathbf{x}) = H^P(\mathbf{x})$ which is included in the combined classifier $H^C(\mathbf{x}) = H^P(\mathbf{x}) + H(\mathbf{x})$.

Similar to the standard boosting we can take a look at the expected value of the loss function [10] and compared to Eq. (2) we get for the combined classifier

$$P(y = 1|\mathbf{x}) = \frac{e^{H^P(\mathbf{x})+H(\mathbf{x})}}{e^{H^P(\mathbf{x})+H(\mathbf{x})} + e^{-H^P(\mathbf{x})-H(\mathbf{x})}}. \tag{19}$$

If we are only interested in the decision we see that a sample is classified as positive if we set $P(y = 1|\mathbf{x}) \geq 0.5$ and after some mathematical rewriting we get

$$\hat{y} = \text{sign}\left(\sinh(H^P(\mathbf{x}) + H(\mathbf{x}))\right) = \text{sign}\left(H^P(\mathbf{x}) + H(\mathbf{x})\right). \tag{20}$$

The interpretation is as follows. A label switch can happen, *i.e.*, $H(\mathbf{x})$ can overrule $H^P(\mathbf{x})$, if the combined term has a different label as the prior $H^P(\mathbf{x})$. As can be easily seen, this is the case if $|H| > |H^P|$. Therefore, the more confident the prior is, the harder it is that the label changes. We do not make any statements whether this is a correct or incorrect label switch. Note, the prior classifier can be wrong, but it has to provide an "honest" decision. Meaning, if it is highly confident it must be ensured to be a correct decision. There are also relations to the co-training [2] assumptions, *i.e.*, a classifier should be never "confident but wrong". By rewriting Eq. (20) as $\hat{y} = \text{sign}(\sinh(H^P(\mathbf{x}) + H(\mathbf{x})) = \text{sign}(\cosh(H(\mathbf{x}))\sinh(H^P(\mathbf{x})) + \cosh(H^P(\mathbf{x}))\sinh(H))$ one sees that it is a weighted combination. The factor obtained by $\cosh(\cdot) \geq 1$ weights the decision of the asymmetric $\sinh(\cdot)$ function for the respectively other classifier. By an additional scaling factor more emphasis can be put either on the prior or the newly trained classifier, however, this is not explored in this paper.

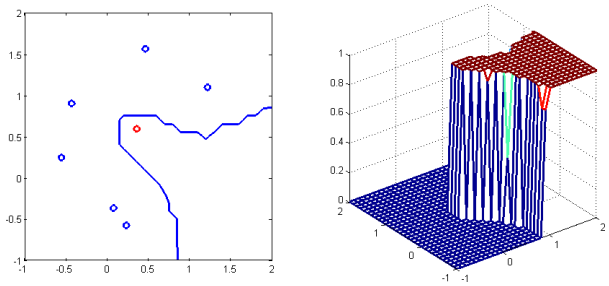
To sum up, after training $H(\mathbf{x})$ the expected target of an example is obtained by a combined decision. The combined classifier can now be interpreted as improving $H^P(\mathbf{x})$ using labeled and unlabeled samples. Note, that we train $H(\mathbf{x})$ using SemiBoost using labeled and unlabeled data, since $H^P(\mathbf{x})$ is used to calculate the similarity via (Eq. (18) and Eq. (17)) these two classifiers are tightly coupled via the training process and Eq. (20) is not just a simple sum rule. If we use a complex (many weak classifiers) classifier and have a lot of training data $H(\mathbf{x})$ will "absorb" the whole knowledge of $H^P(\mathbf{x})$, therefore the usual setting is that we us a rather small $H(\mathbf{x})$ to only correct $H^P(\mathbf{x})$.

4. Experiments

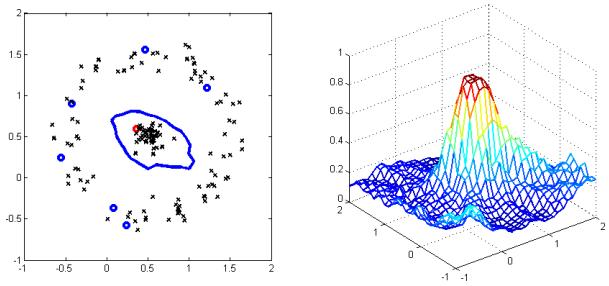
We start with an illustration of the proposed algorithm on a toy example. Then, we show how SemiBoost performs on images with learned pairwise visual similarities. Finally, we demonstrate the classifier improving strategy for face and car detection.

4.1. Toy Experiment

We consider a two class classification problem depicted in Fig. 4. The underlying data generating process produces positive samples around the point $(0.5, 0.5)$ and negative samples at a circle centered at the same point with radius 1 (both with variance 0.1). First, we train a "common" boosting classifier (as weak classifiers a linear separator is used) on just the labeled examples (red and blue circles). Second, we use our proposed SemiBoost approach. Additionally, we use 100 unlabeled points (black crosses) drawn from the



(a) Boosting with labeled data only



(b) SemiBoost: Boosting with labeled and unlabeled data

Figure 4. Toy Example 1: Positive and negative labeled (red and blue circles) and unlabeled samples (back crosses) are used for learning via "common" boosting (a) and using the proposed SemiBoosting approach (b) which additionally takes unlabeled data into account.

distributions above. As distance measure the Euclidean distance is used and converted to a similarity measure using Equation (17) with $\sigma^2 = 0.01$. The left side of the plot shows the samples and the decision border. The right side of each subfigure depicts the probability for the positive class $P(y = 1|\mathbf{x})$. Fig. 4(a) shows a weak decision due to the limited number of samples, Fig. 4(b) using additional unlabeled data an essentially improved decision is obtained by SemiBoost.

The second toy example (Fig. 5) shows improvement of a prior classifier. We build an "honest" prior by estimating the positive and negative probability using a kernel density estimation (Gaussian-distribution with $\sigma^2 = 0.05$) on the labeled samples. This prior serves as similarity measure for SemiBoost, which is used to train a small classifier (10 weak classifiers). The combined classifier performs better than the prior and the newly trained alone, respectively.

4.2. Classification from Few Labeled Examples

We collected a set of 1100 30×30 car patches with the help of a simple motion detector from a common traffic scene (similar to Fig. 9) and, additionally, 1100 random negative patches from the same scene. 300 positives and 300 negatives were kept as an independent test set. First, we trained a similarity using only 15 random positive and negative samples. The distance function is learned on pairs

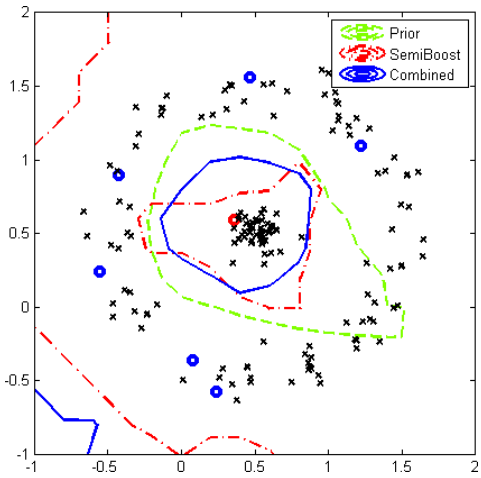
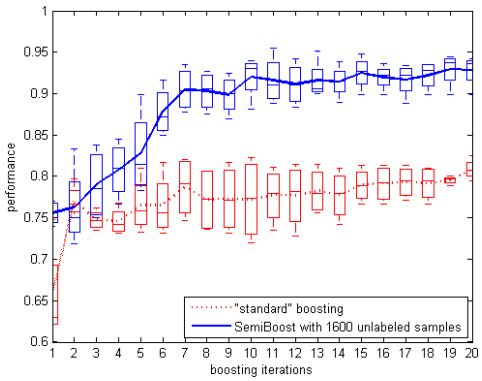
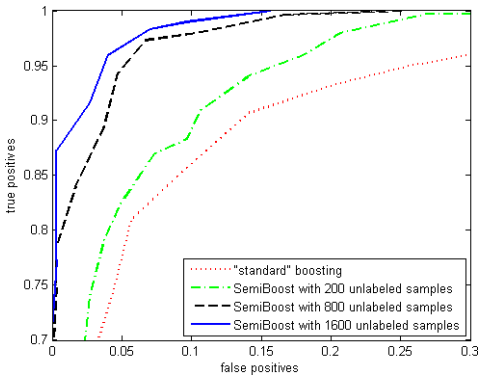


Figure 5. Toy Example 2: The decision boundary of an "honest" prior (green) is "corrected" by a SemiBoost classifier (red) and the combined decision boundary (blue) is archived.



(a) classifier improvement over the boosting iterations



(b) ROC depending on the number of unlabeled samples

Figure 6. Learning of a car-detector: Performance of the proposed approach improves significantly compared to the common approach (no unlabeled data is used) both when (a) including more weak classifiers and (b) use more unlabeled data.

of images as explained in Sec. 3.

In order to train the classifiers we use simple Haar-like

(a) prior (b) trained (c) combined

Figure 7. Detection results of a face detector (a) which serves as prior to build a SemiBoost classifier using additional unlabeled data. This classifier alone (b) has not the power of delivering good results but the combination improves the result essentially.

features as in [26]. Fig. 6(a) compares common boosting and the proposed SemiBoost approach on the test set (the boxplot was obtained by repeating the experiment 5 times). 1600 additionally unlabeled samples were used. As can be seen, the performance increases continuously when adding further weak classifiers and significantly outperforms the standard boosting. Fig. 6(b) shows the performance via ROC-curves of the approach using 200, 800 and 1600 unlabeled samples, respectively.

4.3. Improving a Detector

For each of the following two experiments, we first train a Viola/Jones detector [26] on labeled data. The response of the last cascade layer is used as our prior classifier. The final detection results are obtained by non-maxima suppression as post processing step. Note, our approach is substantially different from other detector improving methods, *e.g.* [17] which is based on co-learning that requires different visual cues.

Face Detector

Fig. 7(a) depicts the results by applying the prior classifier trained on the frequently used MIT+CMU faces where state-of-the-art results are achieved. Then, the classifier was applied on 300 random images downloaded from Google-Image search with the keyword "team". The delivered detections (>4000) are used as additional unlabeled data. The 50 most confident detections were used as positive labeled data and the 50 least confident detections were used as negative ones for training the SemiBoost classifier with only 30 weak classifiers. The proposed combination strategy (Eq. (20)) improved the results (higher detection and lower false positive rate as well as a better alignment of the detections) as shown in Fig. 7(c). Note, the trained classifier alone consists only of 30 weak classifiers which yields poor results when applied on the image (Fig. 7(b)). Of course, when using more weak classifiers it will learn the prior information as well. Additionally, Fig. 8 shows representative examples which were obtained by our approach.

Scene Adaption

A car detector was trained for a specific scene using 1000 labeled samples (a representative result is illustrated in Fig. 9(a)). When applying this classifier on a different scene with a similar view point, as expected, it performs significantly worse (Fig. 9(b)). Hence, in order to adapt the detector to the different scene, we apply a simple motion detector to get potential positive samples. After collecting 1000 of them and additionally cropping 1000 random sub-patches from the scene these 2000 serve as unlabeled examples to train a SemiBoost classifier with 30 weak classifiers. A typical frame superimposed with the detection result is shown in Fig. 9(c). The detection results improved (much lower false positive rate and higher detection rate) as shown in Fig. 9(d).

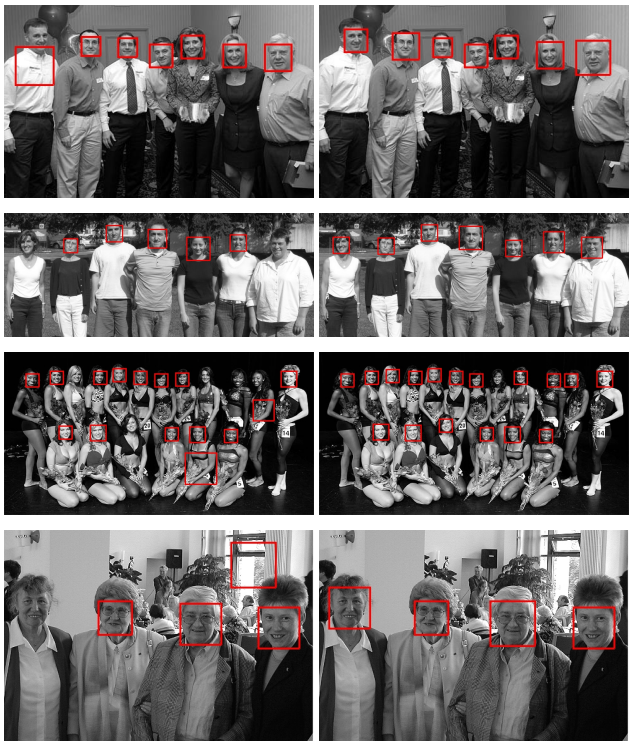
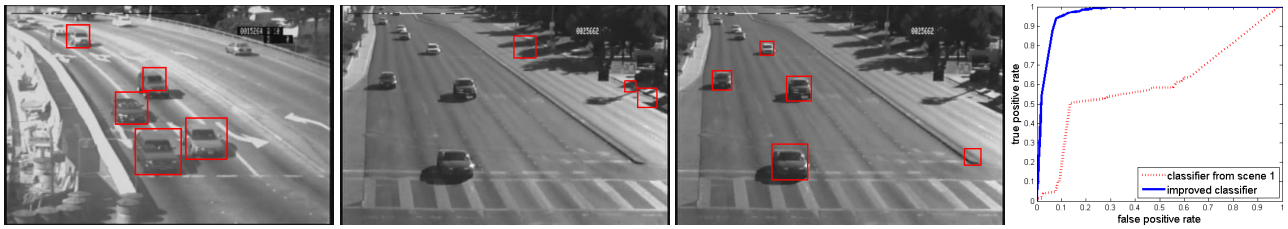


Figure 8. Detection results of a state-of-the-art face detector (left) and the improved results obtained by the proposed strategy (right).

5. Conclusion

In this paper, we have presented a combination of learning visual similarity functions and semi-supervised boosting classifiers. Semi-supervised learning reduces the required amount of labeled training data considerably. This combination has allowed to tackle with challenging vision problems which require specific similarity functions. Furthermore, we have proposed a method to use an a-priori given classifier and improve it by SemiBoost. Experiments illustrate the method on learning and improving object detectors. We are confident to extend this approach for web search applications as well as for on-line learning tasks.



(a) scene 1 (b) scene 2 (c) improved results on scene 2 (d) ROC

Figure 9. A scene specific car detector for scene 1 (a) is applied on a "similar" scene (b). The poor behavior can be significantly improved using unlabeled data taken from the second scene as shown by a typical frame (c) and by a ROC-comparison (d).

References

[1] R. Ando and T. Zhang. A high-performance semi-supervised learning method for text chunking. In *Proc. of the Association for Computational Linguistics*, pages 1–9, 2005.

[2] M.-F. Balcan, A.Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*. MIT Press, 2004.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. 7:2399–2434, 2006.

[4] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *NIPS*, volume 11, pages 368–374. 1999.

[5] K. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. 2002.

[6] D. Cai, D. X. He, and J. Han. Semi-supervised discriminant analysis. In *Proc. ICCV*, 2007.

[7] I. Cohen, N. Sebe, F.G.Cozman, M.C.Cirelo, and T. Huang. Learning bayesian network classifiers for facial expression recognition using both labeled and unlabeled data. In *Proc. CVPR*, volume 1, pages 595–604, 2003.

[8] F. d'Alche Buc, Y. Grandvalet, and C. Ambroise. Semi-supervised marginboost. In *NIPS*. MIT Press, 2002.

[9] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[11] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proc. ICCV*, 2007.

[12] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. ICCV*, 2005.

[13] X. He. Incremental semi-supervised subspace learning for image retrieval. In *Proceedings of the ACM international Conference on Multimedia*, pages 2–8, 2004.

[14] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *Proc. CVPR*, volume 2, pages 570–577, 2004.

[15] S. Hoi and M. Lyu. A semi-supervised active learning framework for image retrieval. In *Proc. CVPR*, volume 2, pages 302–309, 2005.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, volume 2, pages 2169 – 2178, 2006.

[17] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. ICCV*, volume 2, pages 626–633, 2003.

[18] L.-J. Li, G. Wang, and L. Fei-Fei. Optimol: automatic object picture collection via incremental model learning. In *Proc. CVPR*, 2007.

[19] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. Technical report, Department of Computer Science and Engineering, Michigan State University, 2007.

[20] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proc. CVPR*, pages 1–8, 2007.

[21] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proc. ICML*, pages 759–766. ACM Press, 2007.

[22] R. E. Schapire, M.Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *Proc. ICML*, 2002.

[23] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proc. ICML*, pages 824–831, 2005.

[24] T. Steinherz, E. Rivlin, N. Intrator, and P. Neskovic. An integration of online and pseudo-online information for cursive word recognition. *IEEE Trans. on PAMI*, 27(5):669–683, 2005.

[25] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, volume I, pages 511–518, 2001.

[27] J. Yao and Z. Zhang. Semi-supervised learning based object detection in aerial imagery. In *Proc. CVPR*, volume 1, pages 1011–1016, 2005.

[28] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[29] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML*, 2003.