Information Bottleneck Domain Adaptation with Privileged Information for Visual Recognition

Saeid Motiian(✉) and Gianfranco Doretto(✉)

Lane Department of Computer Science and Electrical Engineering,
West Virginia University, Morgantown, USA
{samotiian,gidoretto}@mix.wvu.edu

Abstract. We address the unsupervised domain adaptation problem for visual recognition when an auxiliary data view is available during training. This is important because it allows improving the training of visual classifiers on a new target visual domain when paired additional source data is cheaply available. This is the case when we learn from a source of RGB plus depth data, for then test on a new RGB domain. The problem is challenging because of the intrinsic asymmetry caused by the missing auxiliary view during testing and from which discriminative information should be carried over to the new domain. We jointly account for the auxiliary view during training and for the domain shift by extending the information bottleneck method, and by combining it with risk minimization. In this way, we establish an information theoretic principle for learning any type of visual classifier under this particular settings. We use this principle to design a multi-class large-margin classifier with an efficient optimization in the primal space. We extensively compare our method with the state-of-the-art on several datasets, by effectively learning from RGB plus depth data to recognize objects and gender from a new RGB domain.

1 Introduction

We address the visual recognition problem that involves the classification of a *target data view*, representing the *target domain*, when the training data is composed by unlabeled target domain data and also by *source domain* data, given by a labeled *main data view* paired with an *auxiliary data view*. An important scenario where this problem arises is when dealing with multi-sensory or multimodal data. For example, acquiring RGB plus depth (RGB-D) data is inexpensive (as confirmed by the availability of public labeled datasets [27,28]); however, using them as source for training a visual classifier that is going to be used only on RGB data triggers at least two important observations. First, if the target RGB data has a marginal distribution that is different from the distribution of the source RGB data, then we expect performance to deteriorate. This is due to the well known visual domain adaptation problem, also framed as visual dataset

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 630–647, 2016.
DOI: 10.1007/978-3-319-46478-7_39

bias [43], or covariate shift [40], for which several approaches have been developed [1,15,20–22,35].

The second observation is that domain adaptation methods do not leverage the depth labeled data that RGB-D datasets inherit, and that could be seen as the auxiliary view to the main RGB view. On the other hand, in absence of covariate shift it has been shown that auxiliary data during training could be used to improve recognition performance [45]. Therefore, it is natural to ask whether that improvement could be carried over to a new target RGB domain for visual recognition.

The problem outlined above has received very limited attention. It is different from domain adaptation and transfer learning [3] (where source and target domains are closely related), because of the presence of the auxiliary view as part of the source. It is also different from the Learning Using Privileged Information (LUPI) paradigm [45] (where the auxiliary view would represent privileged information), because the main view and the target view are related but affected by domain bias. Compared to multi-view and multi-task learning [14,18,34,46,49], instead, rather than having all views or task labels available or predicted during testing, here one view is missing, and a single task label is predicted based on a biased view. Therefore, the asymmetry of the missing auxiliary view already poses a challenge (because it cannot be combined like the others in multi-view learning), which becomes even greater when there is a mismatch between the distributions of the source main view and the target view.

We address the auxiliary view problem and the unsupervised domain adaptation (UDA) problem jointly by taking an information theoretic approach. See Fig. 1. We develop a framework in two steps. First, we assume that the target domain view is available as a third labeled view during training. In this way, we derive a model for extracting information from the main and the target views in a way that is optimal for visual recognition, and that speaks also on behalf of the auxiliary view. Subsequently, we show how the model changes in the unsupervised case, with unlabeled target data, effectively posing a UDA problem with auxiliary view. This leads to the independence between the information extracted from the main view and the information extracted from the target view, which ultimately should be used for classification. The framework naturally suggests that the link between the two can be reestablished by imposing the distributions of the two information to be described by the same set of parameters. This is in contrast with current approaches that mostly rely on minimizing the maximum mean discrepancy (MMD) [23], or the Kullback-Leibler (KL) divergence [8] between source and target distributions.

In particular, we rely on the information bottleneck (IB) method [42] as a tool for extracting *latent information* that compresses the available views as much as possible while preserving all the information that is relevant for the task at hand, which is predicting the labels of a visual recognition task. However, the original IB method assumes no domain bias and much less knows about carrying auxiliary information over to a new domain. Therefore, our first contribution is to extend the IB method accordingly, which we call *information bottleneck domain adaptation with privileged information (IBDAPI)*. IBDAPI is

an information theoretic principle for extracting relevant information from the target view, but gives an implicit, hence computationally hard, way for learning a visual classifier based on such information. Hence, our second contribution is a modified version of IBDAPI that allows learning explicitly any type of visual classifier based on risk minimization. As a third contribution we use the modified IBDAPI for learning a large-margin multi-class classifier, called *large-margin IBDAPI (LMIBDAPI)*, for which we provide an optimization procedure guaranteed to converge in the primal space for improved computational efficiency. Finally, we provide an extensive validation of LMIBDAPI against the state-of-the-art on several datasets with very promising results, where we show that we improve object and gender recognition from a new RGB data domain by learning from a RGB-D source.

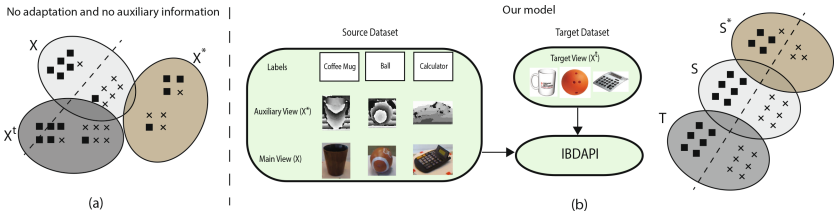

Fig. 1. Domain adaptation with auxiliary information. (a) Since target data distribution $p(X^t)$, and source data distribution $p(X)$ differ by a *covariate shift*, the classifier boundary is suboptimal. Even more so because the paired source auxiliary data X^* is not used for training. (b) Labeled paired source auxiliary data (e.g., depth data) is used, along with unlabeled target data, to improve visual recognition on the target domain via the *information bottleneck domain adaptation with privileged information (IBDAPI)* principle. IBDAPI learns a compressed representation where the mapped source data (S and S^*), as well as the mapped target data (T) become more separable.

2 Related Work

This work is related to domain adaptation (DA), where the distributions of the source and target domain data are different. DA is defined in supervised [12,37], semi-super-vised [44,50], and unsupervised (UDA) [21,35] settings. Since we do not use labeled target data during training this work is more closely related to UDA. There are a number of strategies for UDA. One is to reweigh labeled instances from the source domain in a way that compensates for the difference in the source and target distributions before training a classifier [26,40]. The most popular strategy is to look for a common space where the projected features become domain invariant and then a classifier is learned. Transfer Component Analysis (TCA) [35] searches a latent space where the variance of the data is preserved as much as possible. A number of methods exploit multiple intermediate subspaces for linking source and target distributions. Sampling Geodesic Flow

(SGF) [22] samples subspaces along a geodesic curve on a Grassmann manifold. The Geodesic Flow Kernel method (GFK) [21] extends SGF where the intermediate subspaces are integrated to define a cross-domain similarity measure. Landmark (LMK) [20] further extends GFK by selecting path landmarks from the source domain. Domain Invariant Projection (DIP) [1] focusses on learning a domain invariant subspace representation, and Subspace Alignment (SA) [15] demonstrated that it is possible to map directly the source to the target subspace without necessarily passing through intermediate steps. More recently, [2] applied manifold learning to achieve the above goal by minimizing the Hellinger distance between cross-domain data distributions. Our approach is more closely related to those that jointly look for a feature subspace that minimizes the distribution mismatch, as well as the classifier loss. Among those we mention [16,39] because they do so based on information theoretic measures, like we do. Unlike all the approaches discussed so far, our framework is concerned with exploiting auxiliary data for UDA. In addition, it is different than multi-view domain adaptation methods [51] because we only have single view features in the target domain, rather than multiple types. Moreover, it is also different than multi-domain adaptation methods [11] because we consider a source domain with an auxiliary view.

The only work addressing the same problem as ours is [7], and extended in [30] for web data. They jointly learn a multiclass large-margin classifier, as well as two projections for the main and the auxiliary views, respectively. This is done while maximizing the correlation among views, as well as minimizing the distribution mismatch according to the MMD. On the other hand, we extend the IB method into a general principle that handles the auxiliary view as well as the distribution mismatch from a single information theoretic point of view. Computationally, this entails the estimation of only one projection, rather than two. It allows handling source data points with missing auxiliary view, and we also provide an implementation of a large-margin multiclass classifier in the primal space for improved computational efficiency.

Our approach is also related to the approaches that consider the auxiliary information to be supplied by a teacher during training. This is the LUPI paradigm introduced in [45]. One LUPI implementation is the SVM+ [29,45], later extended to a learning to rank approach in [38], where it is shown that different types of auxiliary information, such as bounding boxes, attributes, and text can be used for learning a better classifier for object recognition. Compared to those approaches, our information theoretic framework learns how to compress the target view for doing prediction in a way that is as informative of the auxiliary view as possible, regardless of the type of classifier used. This is done by extending the original IB method [42]. Other implementations of the LUPI paradigm include [6] for boosting, [17] for object localization in a structured prediction framework, and [47]. However, none of them address the data distribution mismatch between source and target domain.

3 Problem Statement

We are given a training dataset made of triplets $(x_1, x_1^*, y_1), \cdots, (x_N, x_N^*, y_N)$. The feature $x_i \in \mathcal{X}$ is a realization from a random variable X, the feature $x_i^* \in \mathcal{X}^*$ is a realization from a random variable X^*, and the label $y_i \in \mathcal{Y}$ is a realization from a random variable Y. The triplets are i.i.d. samples from a joint probability distribution $p(X, X^*, Y)$. In addition, we are given the data $x_1^t, \cdots, x_M^t$, where $x_i^t \in \mathcal{X}$ is a realization from a random variable X^t, and the data points are i.i.d. samples from a distribution $p(X^t)$. We assume that there is a *covariate shift* [40] between X and X^t, i.e., there is a difference between $p(X)$ and $p(X^t)$. We say that X represents the *main data view*, that X^* represents the *auxiliary data view*, and that X^t represents the *target data view*. The main and auxiliary views represent the *source domain*, and the target view the *target domain*. Under this settings the goal is to learn a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that during testing is going to perform well on data from the target view.

The problem just described is different from the traditional unsupervised domain adaptation (UDA), because we also aim at exploiting the auxiliary data view during training for learning a better prediction function. On the other hand, the presence of the auxiliary view is reminiscent of the Learning Using Privileged Information (LUPI) paradigm as defined in [45], but there is a fundamental difference. In the LUPI framework the prediction function is used only on the main view, and the domain adaptation task is absent. While it has been shown that auxiliary data improves the performance of a traditional classifier [36], how to best carry this improvement over to a new target domain is still an open problem.

4 The Multivariate Information Bottleneck Method

To make the paper more self-contained, we summarize the multivariate extension to the information bottleneck (IB) method [42]. Please refer to [41] for an in-depth treatment. Let us consider a set of random variables $\mathbf{X} = \{X_1, \cdots, X_n\}$, and a set of *latent* variables $\mathbf{T} = \{T_1, \cdots, T_n\}$. $\mathbf{X}$ is distributed according to a known $p(\mathbf{X})$. A Bayesian network with graph G_{in} over $\mathbf{X} \cup \mathbf{T}$, defines a distribution $q(\mathbf{X}, \mathbf{T}) = q(\mathbf{T}|\mathbf{X})p(\mathbf{X})$, and in particular it defines which subset of $\mathbf{X}$ is compressed by which subset of $\mathbf{T}$, through $q(\mathbf{T}|\mathbf{X})$. In addition, another Bayesian network, G_{out}, is also defined over $\mathbf{X} \cup \mathbf{T}$, and represents which conditional dependencies and independencies we would like $\mathbf{T}$ to be able to approximate.

The compression requirements defined by G_{in}, and the desired independencies defined by G_{out}, are incompatible in general. Therefore, *the multivariate IB method computes the optimal* $\mathbf{T}$ *by searching for the distribution* $q(\mathbf{T}|\mathbf{X})$*, where* $\mathbf{T}$ *compresses* $\mathbf{X}$ *as much as possible, while the distance from* $q(\mathbf{X}, \mathbf{T})$ *to the closest distribution among those consistent with the structure of* G_{out} *is minimal*. The multivariate IB method [41] implements this idea by using the *multi-information* of $\mathbf{X}$, which is the information shared by $X_1, \cdots, X_n$, i.e., $\mathcal{I}(\mathbf{X}) = D_{KL}[p(\mathbf{X}) \| p(X_1) \cdots p(X_n)]$, where D_{KL} indicates the Kullback-Leibler

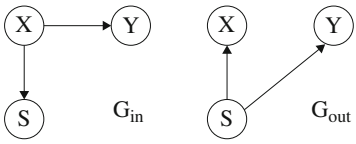

Fig. 2. Information bottleneck. Structural representation of G_{in} and G_{out} used by the original two-variable information bottleneck method [42].

divergence [8] between $p(\mathbf{X})$ and $p(X_1)p(X_2)\cdots p(X_n)$. The resulting multivariate IB method looks for $q(\mathbf{T}|\mathbf{X})$ that minimizes the functional

$$\mathcal{L}[q(\mathbf{T}|\mathbf{X})] = \mathcal{I}^{G_{in}}(\mathbf{X},\mathbf{T}) + \gamma(\mathcal{I}^{G_{in}}(\mathbf{X},\mathbf{T}) - \mathcal{I}^{G_{out}}(\mathbf{X},\mathbf{T})), \tag{1}$$

where γ strikes a balance between the compression requirements set by G_{in}, and the independency goals set by G_{out}.

Let us refer to Fig. 2 for an example, where $\mathbf{X} = \{X, Y\}$, and $\mathbf{T} = S$. We interpret X as the *main data* we want to compress, and from which we would like to predict the *relevant information* Y. This is achieved by first compressing X into S, and then predicting Y from S. In G_{in} the edge $X \to Y$ indicates the relation defined by $p(X, Y)$. The edge $X \to S$ instead, shows that S is completely determined given X, which is the variable it compresses. On the other hand, the structure of G_{out} is such that S should capture from X all the necessary information to best predict Y. Equivalently, this means that knowing S should make X and Y independent, i.e., the *mutual information* [8] between X and Y, conditioned on S, should be $I(X; Y|S) = 0$.

In general, to compute the functional in (1), if G is a Bayesian network structure over $\mathbf{X} \sim p(\mathbf{X})$, then $\mathcal{I}^G$, the multi-information with respect to G [41], is computed as

$$\mathcal{I}^G(\mathbf{X}) = \sum_i I(X_i; \mathbf{Pa}^G_{X_i}), \tag{2}$$

where $I(X_i; \mathbf{Pa}^G_{X_i})$ represents the mutual information between X_i and $\mathbf{Pa}^G_{X_i}$, the set of variables that are parents of X_i in G. If we apply the multivariate IB method (1) to the two-variable case in Fig. 2, we obtain $\mathcal{I}^{G_{in}} = I(S; X) + I(Y; X)$, and $\mathcal{I}^{G_{out}} = I(X; S) + I(Y; S)$. Since $I(Y; X)$ is constant, the functional in (1) collapses to the original two-variable IB method [42].

5 IB for UDA with Auxiliary Data

We use the multivariate IB framework of Sect. 4 to develop a new information bottleneck principle, which simultaneously accounts for the use of auxiliary data, as well as the adaptation to a new target domain. Specifically, let us assume that X, X^*, X^t and Y are four random variables with known distribution $p(X, X^*, X^t, Y)$. We develop the principle in two steps. First, we assume that the target view is an additional view of the source domain, and we extend

the IB method to handle the auxiliary the main and the target views in the source, and the main and target views in the target domain. Then, we assume that the target view does not carry information about Y, and we address the covariate shift.

5.1 Incorporating Auxiliary Data

We assume that both X, X^*, and X^t carry information about Y. In addition, only the information carried by X and X^t can be used to predict Y. We want to design a principle for learning a model for prediction that also exploits the information carried by X^*.

The straightforward application of the multivariate IB method suggests to compress X into a latent variable S, and X^t into a latent variable T, as much as possible, while making sure that information about Y is retained. These two competing goals are depicted by the graphs G_{in} and G_{out} in Figs. 3(a) and (b). Therefore, the IB method would seek for the optimal representation given by $q(X^t, X, X^*, Y, S, T) = q(S, T|X, X^t)p(X^t, X, X^*, Y)$, where $q(S, T|X, X^t)$ is such that $I(X; Y|S)$ and $I(X^t; Y|T)$ are as close to zero as possible. On the other hand, since X^* has knowledge about Y (as highlighted by the connection $X^* \rightarrow Y$ in G_{in}), we observe that $I(X^*; Y|S)$ and $I(X^*; Y|T)$ could be arbitrarily high. This means that knowing S and T still leaves with X^* substantial information about Y.

We address the problem just outlined by modifying G_{out} as in Fig. 3(c), where the edges $S \rightarrow X^*$ and $T \rightarrow X^*$ have been added. In this way, knowing S and T makes not only X and Y independent, as well as X^t and Y, but also makes X^* and Y independent. This also means that the optimal $q(S, T|X, X^t)$ will minimize $I(X; Y|S)$ and $I(X^t; Y|T)$, as well as $I(X^*; Y|S)$ and $I(X^*; Y|T)$. In particular, the multi-informations of G_{in} and G_{out} in Figs. 3(a) and (c) are given by

$$\mathcal{I}^{G_{in}} = I(S; X) + I(T; X^t) + I(Y; X^t, X, X^*), \tag{3}$$

$$\mathcal{I}^{G_{out}} = I(S; X) + I(T; X^t) + I(S, T; X^*) + I(S, T; Y). \tag{4}$$

By plugging (3) and (4) into (1), since $I(Y; X^t, X, X^*)$ is constant, the functional for learning the optimal representation for S and T is given by

$$\mathcal{L}[q(S, T|X, X^t)] = I(S; X) + I(T; X^t) - \gamma I(S, T; X^*) - \gamma I(S, T; Y), \tag{5}$$

where γ strikes a balance between compressing X and X^t and imposing the independency requirements.

5.2 Adapting to a New Target Domain

Model (5) incorporates the target view X^t under the assumption that it can predict the relevant information Y. This implies a fully supervised scenario, where training data should be given in quadruplets, i.e., (x_i^t, x_i, x_i^*, y_i). On the

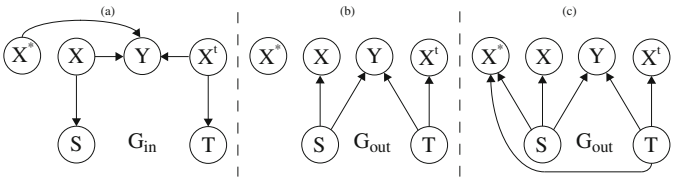

Fig. 3. Information bottleneck with auxiliary data. Structural representation of G_{in} (a), and G_{out} (b,c) used by the information bottleneck method. (b) G_{out} does not leverage the auxiliary data. (c) G_{out} leverages the auxiliary data.

other hand, we are interested in the unsupervised setting, where the training target view is not labeled and not paired with the source data. From a statistical point of view, this assumption corresponds to saying that $p(X^t, X, X^*, Y) = p(X^t)p(X, X^*, Y)$, which leads to a number of consequences. First, the graph G_{in} of Fig. 3(a) now becomes as in Fig. 4(a), where we do not consider the dotted edges for the moment. In addition, it is easy to show that $I(S, T; X^*) = I(S; X^*)$, and that $I(S, T; Y) = I(S; Y)$. Therefore, the graph structure G_{out} in Fig. 3(c) now becomes as in Fig. 4(b). Finally, it is also easy to show that $q(S, T|X, X^t) = q(S|X)q(T|X^t)$. Therefore, the *unsupervised* scenario reduces model (5) to the following

$$\mathcal{L}[q(S|X), q(T|X^t)] = I(S; X) + I(T; X^t) - \gamma I(S; X^*) - \gamma I(S; Y). \tag{6}$$

We note that estimating the optimal compressed representation S and T of X and X^t, by minimizing (6) leads to an ill-posed problem. This is because at convergence $q(T|X^t)$ would simply minimize $I(T; X^t)$. On the other hand, we are interested in addressing the distribution mismatch between the main view and the target view. Therefore, rather than treating $q(S|X)$ and $q(T|X^t)$ as separate free functions, we make the assumption that the compression maps from the main and the target views should cause $q(S|X)$ and $q(T|X^t)$ to be the same, in order to minimize the covariate shift in the compressed domain. If we restrict the search for the optimal representation within a family of distributions parameterized by A, this means that $q(S|X) \doteq q_A(S|X)$ and $q(T|X^t) \doteq q_A(T|X^t)$, i.e., they should have the same parameter. This assumption would impose $q(S|X)$ and $q(T|X^t)$ to no longer be independent, and therefore all the consequences originated by the statistical independence of X^t would be reversed, to a certain extent. In other words, it would be as if the links $X^t \to Y$ in G_{in}, and $T \to X^*$ and $T \to Y$ in G_{out}, were partially restored, which is why they appear with dotted lines in Fig. 4. Finally, this assumption reduces (6) to the proposed principle

$$\boxed{\mathcal{L}[q_A(\cdot|\cdot)] = I(S; X) + I(T; X^t) - \gamma I(S; X^*) - \gamma I(S; Y)} \tag{7}$$

Since the auxiliary view plays the role of privileged information, we call learning representations by minimizing the functional (7) as the *information bottleneck domain adaptation with privileged information (IBDAPI).*

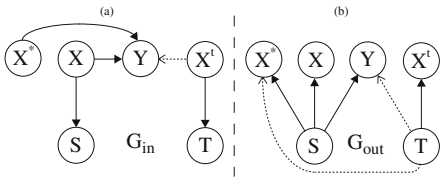

Fig. 4. Information bottleneck domain adaptation with privileged information. Structural representation of G_{in} and G_{out} used by the IBDAPI principle (7).

6 IBDAPI for Visual Recognition

Our goal is to design a framework for visual recognition, where a classification task is based on the *target* view X^t of the visual data, for which some unlabeled samples are given for training. Moreover, at training time labeled samples from a *main* view X are also given, as well as some samples from an *auxiliary* view X^*. We pose no restrictions on the type of auxiliary data available.

The IBDAPI method (7) learns how to compress X and X^t into S and T in a way that is optimal for predicting Y (representing class labels), but also that best exploits the information carried by X^* about Y. Therefore, T appears to be the representation of choice for predicting Y. However, while IBDAPI provides for a compression map defined explicitly by $q_A(\cdot|\cdot)$, the prediction map for doing classification, identified by $q(Y|S)$ is much harder to compute in general. This is why we modify the IBDAPI method into one that is tailored to visual recognition.

We note that the last term in (7) is equivalent to the constraint $I(S;Y) \geq$ *constant* if γ is interpreted as a Lagrange multiplier. This means that S should carry at least a certain amount of information about Y. On the other hand, we are interested in learning a decision function $f : \mathcal{S} \rightarrow \mathcal{Y}$ that uses such information for classification purposes. Therefore, we replace the constraint on $I(S;Y)$ with the risk associated to $f(S)$ according to a loss function ℓ. Thus, for visual recognition, (7) is modified into

$$\boxed{\mathcal{L}[q_A(\cdot|\cdot), f] = I(S;X) + I(T;X^t) - \gamma I(S;X^*) + \beta E[\ell(f(S),Y)]} \qquad (8)$$

where $E[\cdot]$ denotes statistical expectation, and β balances the risk versus the compression requirements. Note that the modified IBDAPI criterion (8) is general, and could be used with any classifier.

6.1 Large-Margin IBDAPI

We use (8) for learning a multi-class large-margin classifier. We parameterize the search space for $q_A(\cdot|\cdot)$ by assuming $S = \phi(X;A)$, as well as $T = \phi(X^t;A)$, where A is a suitable set of parameters. Moreover, $f(S)$ is a k-class decision function given by $Y = \arg\max_{m=1,\cdots,k} \langle w_m, S \rangle$, where $\langle \cdot,\cdot \rangle$ identifies a dot product, and $W = [w_1, \cdots, w_k]$ defines a set of margins. Therefore, based on [9], (8) leads to the following classifier learning formulation, which we refer to as the *large-margin IBDAPI (LMIBDAPI)*

$$\min_{A,W,\xi_i} I(S;X) + I(T;X^t) - \gamma I(S;X^*) + \frac{\beta}{2}\|W\|_2^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \tag{9}$$
$$\text{s.t.} \quad \langle w_{y_i} - w_m, \phi(x_i, A)\rangle \geq e_i^m - \xi_i \ , \quad \xi_i \geq 0 \ , \quad m = 1, \cdots, k \ , \quad i = 1, \cdots, N.$$

where $e_i^m = 0$ if $y_i = m$ and $e_i^m = 1$ otherwise. ξ_i indicates the usual slack variables, and C is the usual parameter to control the slackness.

Kernels. We set $S = \phi(X, A) = A\phi(X)$, and $T = \phi(X^t, A) = A\phi(X^t)$, where we require $\phi(X)$ and $\phi(X^t)$ to have positive components and be normalized to 1, and A to be a stochastic matrix, made of conditional probabilities between components of $\phi(X)$ ($\phi(X^t)$) and S (T). This assumption greatly simplifies computing mutual informations. As described in [32], this mapping also allows the use of kernels. X^* is mapped to a feature space with the same requirements by using the same strategy. Thus, without loss of generality, in the sequel we set $S = AX$, and $T = AX^t$.

Mutual informations. $I(S;X)$ and $I(T;X^t)$ are given by

$$I(S;X) = E\left[\sum_{i,j} A(i,j)X(j)\log\frac{A(i,j)}{S(i)}\right] \quad I(T;X^t) = E\left[\sum_{i,j} A(i,j)X^t(j)\log\frac{A(i,j)}{T(i)}\right] \tag{10}$$

where $A(i,j)$ is the entry of A in position i,j, whereas $S(i)$ and $X(j)$ ($T(i)$ and $X^t(j)$) are the components in position i and j of S and X (T and X^t) respectively. Obviously, during training the expectation is replaced by the empirical average. To compute $I(S;X^*)$, it is easy to show that

$$I(S;X^*) = E\left[\sum_{i,j} A(i,\cdot)F(\cdot,j)X^*(j)\log\frac{A(i,\cdot)F(\cdot,j)}{S(i)}\right] \tag{11}$$

where F is also a stochastic matrix such that $X = FX^*$. F can be learned from the source training data with a projected gradient method [31], as described in [32].

Missing auxiliary views. Training samples with missing auxiliary view affect only $I(S;X^*)$. The issue is seamlessly handled by estimating F and the average in (11) by using only the samples that have the auxiliary view.

Optimization. When A is known, (9) is a soft-margin SVM problem. Instead, when the SVM parameters are known, (9) becomes

$$\min_{A} \ I(S;X) + I(T;X^t) - \gamma I(X^*;S) + \frac{C}{N}\sum_{i=1}^{N}\xi_i \tag{12}$$
$$\text{s.t.} \quad \xi_i = \max_{m=1,\cdots,k}\{\langle w_m - w_{y_i}, \phi(x_i, A)\rangle + e_i^m\}\,.$$

Since the soft-margin problem is convex, if also (12) is convex, then an alternating direction method is guaranteed to converge. In general, the mutual informations in (12) are convex functions of $q(S|X)$ and $q(T|X^t)$ [8], while within a range of γ's the third mutual information leaves the sum of the three to be convex. The

last term is also convex, however, the constraints define a non-convex set due to the discontinuity of the hinge loss function. Smoothing the hinge loss turns (12) into a convex problem, and allows to use an alternating direction method with variable splitting combined with the augmented Lagrangian method. This is done by setting $f(A) = I(S;X) + I(T;X^t) - \gamma I(X^*;S)$, $g(B) = \frac{C}{N}\sum_{i=1}^{N}\xi_i$, and then solving $\min_A\{f(A) + g(B) : A - B = 0\}$.

For smoothing the hinge loss we use the Nesterov smoothing technique [33]. Since the objective is to smooth $g(B)$, we proceed by relaxing its minimization into the sum of the minima of the slack variables. Doing so gives $\bar{g}(B)$, the smoothed version of $g(B)$, expressed as

$$\bar{g}(B) = \frac{C}{N}\sum_{i=1}^{N}\mu\ln(\frac{1}{m}\sum_{m=1}^{k}\cosh(\frac{1}{\mu}(\langle w_m - w_{y_i}, \phi(x_i, B)\rangle - e_i^m))) \tag{13}$$

and μ is a smoothing parameter. In this way, the minimization can be carried out with the Fast Alternating Linearization Method (FALM) [19]. This allows simpler computations, and has performance guarantees when ∇f and $\nabla\bar{g}$ are Lipschitz continuous, which is the case, given the smoothing technique that we have used. In particular, given the limited space, we are not able to report all the details of the FALM algorithm that we have used. However, the interested reader is referred to [32], where an almost identical FALM algorithm has been used, which has the same requirement of A and B to be stochastic matrices with normalized columns.

In summary, we provide an optimization procedure guaranteed to converge, which starts by learning F. Then, until convergence alternates between learning a SVM, and solving (12). Note that this iterative optimization is fully conducted in the primal space for best computational efficiency.

Table 1. RGB-D-Caltech256 dataset. Classification accuracies for one-vs-all binary classifications with linear kernels. Main and auxiliary views are KDES features of the RGB and depth of the RGB-D Object dataset [28]. KDES features from the Caltech256 dataset [24] represent the target domain.

	MV and LUPI					UDA				UDA+LUPI	
	SVM	SVM2k	KCCA	SVM+	RankTr	SGF	LMK	SA	LMIBDA	DA-M2S	LMIBDAPI
Calculator	49.83 ± 1.65	50.08 ± 1.87	48.10 ± 2.58	54.61 ± 3.37	53.27 ± 1.26	54.23 ± 1.26	53.71 ± 2.78	54.22 ± 3.32	56.33 ± 2.78	55.63 ± 2.89	**59.52 ± 2.18**
Cereal box	69.10 ± 3.41	67.10 ± 3.60	67.40 ± 3.20	62.78 ± 3.53	63.26 ± 4.98	65.23 ± 3.25	66.81 ± 2.59	67.17 ± 3.89	67.92 ± 2.11	68.50 ± 4.27	**72.60 ± 2.63**
Coffee mug	57.95 ± 3.03	57.61 ± 3.97	57.13 ± 5.99	58.32 ± 3.45	58.36 ± 3.69	66.23 ± 4.21	67.36 ± 3.89	68.12 ± 5.11	68.36 ± 3.11	70.11 ± 5.19	**75.65 ± 3.39**
Keyboard	60.79 ± 6.04	59.77 ± 6.41	59.40 ± 6.08	58.21 ± 3.88	57.98 ± 3.48	61.59 ± 3.27	59.26 ± 3.89	62.65 ± 3.14	63.36 ± 3.25	63.52 ± 4.68	**68.50 ± 3.71**
Flashlight	72.06 ± 2.60	70.86 ± 3.95	70.56 ± 3.20	71.36 ± 2.21	70.68 ± 4.24	72.36 ± 2.78	70.26 ± 2.15	73.25 ± 2.68	72.15 ± 2.14	71.37 ± 2.78	**74.79 ± 2.51**
Lightbulb	67.09 ± 2.32	65.23 ± 2.71	66.69 ± 3.06	68.36 ± 3.77	67.58 ± 2.15	67.99 ± 1.89	66.36 ± 2.11	68.11 ± 1.67	67.23 ± 2.85	68.48 ± 3.81	**71.81 ± 1.49**
Mushroom	49.02 ± 4.45	51.41 ± 3.97	49.04 ± 3.54	54.71 ± 5.86	56.84 ± 4.15	66.36 ± 3.87	64.26 ± 4.15	68.22 ± 3.89	69.26 ± 3.14	70.00 ± 5.10	**70.39 ± 2.96**
Ball	45.19 ± 2.11	48.96 ± 0.78	45.05 ± 4.44	53.27 ± 1.84	54.48 ± 3.25	60.25 ± 2.11	61.36 ± 2.87	63.86 ± 1.89	64.95 ± 2.67	**67.27 ± 5.32**	65.45 ± 3.71
Soda can	52.04 ± 3.46	50.00 ± 3.30	50.09 ± 3.33	52.48 ± 3.76	50.26 ± 1.36	56.58 ± 2.18	55.71 ± 2.65	58.36 ± 2.14	60.33 ± 2.35	59.65 ± 2.63	**62.93 ± 2.84**
Tomato	56.05 ± 3.73	50.76 ± 0.99	53.69 ± 3.03	51.55 ± 3.71	50.23 ± 2.59	63.25 ± 2.17	64.25 ± 1.36	64.33 ± 2.74	64.26 ± 2.36	64.61 ± 3.19	**73.40 ± 2.22**
Average	57.91	57.18	56.71	58.56	58.29	63.41	62.93	64.83	65.42	65.91	**69.50**

7 Experiments

We have performed experiments on several datasets for object and gender recognition, and have compared our approach with several others summarized as follows.

Single-view classifiers: Using only the main view, we use libSVM [5] and LIBLINEAR [13] (indicated as `SVM`) for training binary and multi-class SVM classifiers.

LUPI and multi-view (MV) classifiers: By using the main and auxiliary views, we train the SVM+ [45] (indicated as `SVM+`, the Rank Transfer [38] (indicated as `RankTr`). We also train the SVM2k [14] and test only the SVM that uses the main view (indicated as `SVM2k`), and we perform kernel CCA (KCCA) [25] between main and auxiliary views, map the main view in feature space and train an SVM (indicated as `KCCA`). `SVM+`, `RankTr`, `SVM2k`, and `KCCA`, can be used only for binary classification.

UDA classifiers: We use the main view and the target training data for learning the Sampling Geodesic Flow (`SGF`) [22], the Landmark (`LMK`) [20], the Subspace Alignment (`SA`) [15], the Transfer Component Analysis (`TCA`) [35], and the Domain Invariant Projection (`DIP`) [1] classifiers. In addition, we use LMIBDAPI where we eliminate the auxiliary information by setting $\gamma = 0$ (indicated as `LMIBDA`).

UDA+LUPI classifiers: Besides our approach, indicated as `LMIBDAPI`, we consider the only other approach designed to work in the same settings, which is [7] (indicated as `DA-M2S`).

Model selection: We use the same joint cross validation and model selection procedure described in [38], based on 5-fold cross-validation to select the best parameters and use them to retrain on the complete set. The main parameters to select are C, β, γ, and r, which is the number of columns of A. The C's and β's were searched in the range $\{10^{-3}, \cdots, 10^{3}\}$, the γ's in the range $\{0.1, 0.3, 0.5\}$. r was set by doing PCA on the mapped main view data (through $\phi(\cdot)$), and thresholding at 90 % of the summation of the eigenvalues. In addition, for `DA-M2S` we set two parameters as indicated in [7], while for C and the others we look for those that maximize performance.

Performance: Average classification accuracy and standard deviation are reported. Testing is always done on the target domain data.

Object recognition: We evaluate the proposed approach for object recognition where we use the RGB-D Object dataset [28] as source domain, and the Caltech256 dataset [24] as target domain. We follow the same protocol outlined in [7], where we consider the 10 classes reported in Table 1, which are in common between the two datasets. Instances in the RGB-D Object are given as videos, and we uniformly sample frames every two seconds, obtaining 2056 training images. All the images of the 10 Caltech256 classes instead are used as unlabeled training target data.

Following [7], kernel descriptor (KDES) features [4], which perform well on the RGB-D Object dataset, are computed from the color and depth images to represent the main and the auxiliary views, respectively, and KDES features from the color images of the Caltech256 represent the target view. For each view

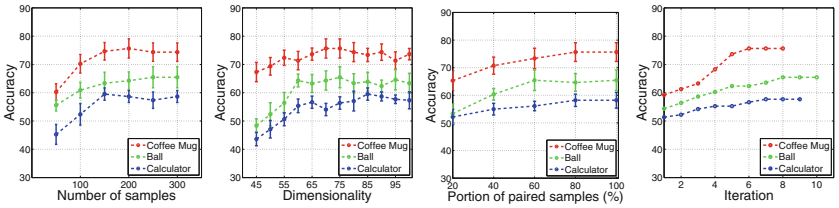

Fig. 5. RGB-D-Caltech256 dataset. Classification accuracy variation for three classes of Table 1. In particular, from left to right: Accuracy variation against M, the number of training target domain samples; Accuracy variation against r, the dimensionality of T and S; Accuracy variation against the fraction of available auxiliary data; Convergence rate of the accuracy against the number of iterations of the learning procedure.

we compute the Gradient KDES and the LBP KDES and we concatenate them. We set the vocabulary size to 1000, and use three level of pyramids.

Table 2. RGB-D-Caltech256 dataset. Classification accuracies for the multi-class classification with Gaussian kernels. Main and auxiliary views are KDES features of the RGB and depth of the RGB-D Object dataset [28]. KDES features from the Caltech256 dataset [24] represent the target domain.

	UDA						UDA+LUPI	
SVM	SGF	LMK	SA	TCA	DIP	LMIBDA	DA-M2S	LMIBDAPI
18.23	19.41	19.69	19.83	25.07	25.47	27.23	29.47	**34.22**

For each of the 10 object classes, Table 1 shows the accuracies for the one-vs-all binary classification with linear kernels. Here we randomly selected 50 positive and 50 negative training samples from the source domain, and the experiment was repeated 10 times. We observe that on average the multi-view based methods perform on par with the SVM, and the LUPI methods better exploit the information from the auxiliary view, but they all suffer from the lack of adaptation. The UDA methods perform better overall, highlighting the need to address the domain shift before taking advantage of the auxiliary view. In particular, we notice that LMIBDA, which does not use the auxiliary view, is an effective UDA approach. The last two columns address domain shift while leveraging the auxiliary view information, and show that the proposed LMIBDAPI provides state-of-the-art performance on this task.

Table 2 shows the classification accuracies for the multiclass classification case using Gaussian kernels, where all the source samples are used for training. Even for this case, UDA methods improve upon the baseline SVM, and LMIBDA performs effectively, while LMIBDAPI confirms to have the best performance.

Figure 5 shows how the one-vs-all binary classification accuracy for three classes of Table 1 varies with respect to a number of parameters. The leftmost

plot shows how the accuracy changes against the number M of training target domain samples. After a number of samples (about 200 in this case), the model saturates and additional samples will no more compensate for data shift. The second plot from the left shows that increasing r (i.e., the dimensionality of S and T), does not help beyond a certain limit (here between 60 and 70). Once it is reached, the model has enough capacity to extract all the necessary information for prediction. Beyond that limit the accuracy does not improve anymore and shows a noisy behavior. Choosing r below the limit reduces the capacity and thus prediction accuracy. The second plot from the right shows the accuracy variation against the fraction of available auxiliary data (or conversely, the fraction of missing auxiliary data). Note that handling missing auxiliary data is peculiar to our approach. The plot shows that at least 20 % of missing auxiliary data is tolerated without performance drop. Finally, the rightmost plot shows the rate of convergence of the optimization procedure, which occurs monotonically. We found that no more than 10 iterations were normally enough to reach convergence, which is fairly good.

Table 3. Office dataset. Classification accuracy for domain adaptation over the 31 categories of the Office dataset [37]. $\mathcal{A}$, $\mathcal{W}$, and $\mathcal{D}$ stand for Amazon, Webcam, and DSLR domain.

	SVM-s	SVM-t	LMK	HFA	GFK	SDASL	LMIBDA
$\mathcal{A} \rightarrow \mathcal{W}$	51.95	80.94	81.15	78.61	83.26	85.40	**86.10**
$\mathcal{A} \rightarrow \mathcal{D}$	54.92	82.90	82.31	83.71	82.72	**85.77**	85.31
$\mathcal{W} \rightarrow \mathcal{A}$	49.21	63.91	60.24	65.65	65.92	67.26	**67.41**
$\mathcal{W} \rightarrow \mathcal{D}$	83.26	81.91	82.26	86.10	84.28	86.18	**87.15**
$\mathcal{D} \rightarrow \mathcal{A}$	48.51	62.98	62.18	64.60	65.45	66.76	**66.82**
$\mathcal{D} \rightarrow \mathcal{W}$	80.35	82.65	83.45	81.69	82.69	**84.65**	83.36

Table 3 shows the classification accuracy of the proposed approach for UDA without auxiliary data on the Office dataset [37], which contains 31 object classes for 3 domains: Amazon, Webcam, and DSLR, indicated as $\mathcal{A}$, $\mathcal{W}$, and $\mathcal{D}$, for a total of 4,652 images. The first domain consists of images downloaded from online merchants, the second consists of low resolution images acquired by webcams, the third consists of high resolution images collected with digital SLRs. The table notation $\mathcal{A} \rightarrow \mathcal{W}$ indicates that $\mathcal{A}$ was the source domain, and $\mathcal{W}$ the target. All the source data was used for training, whereas the target data was evenly split into two halves: one used for training and the other for testing. We used the 1000-way fc8 classification layer computed by DeCAF [10] as image features, and Gaussian kernels set up as detailed in [50]. We compared LMIBDA against LMK, the heterogeneous domain adaptation method (HFA) [12], the geodesic flow kernel method (GFK) [21], and against a recent semi-supervised domain adaptation method (SDASL) [50], which uses some labeled target data for training. The

SVM trained on the source and on the target domain data, indicated as SVM-s and SVM-t, is also reported for reference. The main result is that even with this more popular domain adaptation dataset, the proposed approach, restricted to UDA only, has performance comparable to the state-of-the art.

Gender recognition: We evaluate the proposed approach also for gender recognition where we use the RGB-D face dataset EURECOM [27] as source domain, and the RGB dataset Labeled Faces in the Wild-a (LFW-a) [48] as target domain. The EURECOM dataset consists of pairs of RGB and depth images from 196 females and 532 males captured with the Kinect sensor, and we removed the profile face images, which had only one manually annotated eye position. The LFW-a dataset contains images from 2,960 females and 10,184 males captured in uncontrolled conditions.

We resized the main, the auxiliary, and the target view face images to 120×105 pixels, and divide them into 8×7 non-overlapping subregions of 15×15 pixels. From each subregion of an image we extract the Gradient-LBP features, shown to be effective for gender recognition [27], and concatenate them into a single feature vector.

We perform a gender recognition experiment by combining the female source pairs with 196 randomly selected male source pairs to have a balanced gender representation. In addition, we randomly sample 3000 unlabeled target face images for training. The experiment is repeated 10 times, and the classification accuracies of all the methods are reported in Table 4. The results show a pattern similar to the one found for object recognition in Tables 1 and 2. One difference might be that in this experiment leveraging the auxiliary depth information seems to be as important as addressing the RGB domain shift. This is because the performance increase of the best LUPI methods is comparable to the performance increase of the best UDA methods. We also note that even here, LMIBDA confirms to be an effective UDA method by surpassing all the UDA and LUPI methods. Finally, although DA-M2S marginally improves by leveraging auxiliary information and addressing domain shift, the proposed LMIBDAPI provides a remarkable performance increase.

Table 4. EURECOM-LFW-a dataset. Classification accuracies for the male vs. female classification with Gaussian kernels. Main and auxiliary views are Gradient-LBP features of the RGB and depth of the EURECOM dataset [27]. Gradient-LBP features from the LFW-a dataset [48] represent the target domain.

	MV and LUPI			UDA						UDA+LUPI	
SVM	SVM2k	KCCA	SVM+	SGF	LMK	SA	TCA	DIP	LMIBDA	DA-M2S	LMIBDAPI
64.82 ± 1.35	67.15 ± 1.25	63.85 ± 1.34	67.31 ± 1.96	67.81 ± 1.45	64.88 ± 1.31	67.11 ± 1.45	65.24 ± 0.88	64.84 ± 4.80	68.11 ± 1.64	68.22 ± 1.41	**72.43 ± 1.34**

8 Conclusions

We developed an unsupervised domain adaptation approach for visual recognition when auxiliary information is available at training time. We extended the IB

principle to IBDAPI, a new information theoretic principle that jointly handles the auxiliary view and the mismatch between the source and target distributions. We provided a modified version of IBDAPI based on risk minimization for learning explicitly any type of classifier, where training samples with missing auxiliary view can be handled seamlessly. We used this principle for deriving LMIBDAPI, a large-margin classifier with a fast optimization procedure in the primal space that converges in about 10 iterations. We performed experiments on object and gender recognition on a new target RGB domain by learning from a different RGB plus depth dataset. We observed that without using auxiliary data LMIBDA performs UDA with performance comparable with the state-of-the art. In addition, LMIBDAPI consistently outperformed the state-of-the-art, confirming its ability to carry the content of the auxiliary information over to a new domain.

References

1. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: IEEE ICCV, pp. 769–776 (2013)
2. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Domain adaptation on the statistical manifold. In: CVPR, pp. 2481–2488 (2014)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. **79**(1–2), 151–175 (2009)
4. Bo, L., Ren, X., Fox, D.: Depth kernel descriptors for object recognition. In: IROS (2011)
5. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 2701–2727 (2011)
6. Chen, J., Liu, X., Lyu, S.: Boosting with side information. In: ACCV, pp. 563–577 (2012)
7. Chen, L., Li, W., Xu, D.: Recognizing RGB images by learning from RGB-D data. In: CVPR, pp. 1418–1425, June 2014
8. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley and Sons, Inc., New York (1991)
9. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. JMLR **2**, 265–292 (2001)
10. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: a deep convolutional activation feature for generic visual recognition (2013). arXiv:1310.1531
11. Duan, L., Xu, D., Tsang, I.W.H.: Domain adaptation from multiple sources: a domain-dependent regularization approach. IEEE TNNLS **23**(3), 504–518 (2012)
12. Duan, L., Xu, D., Tsang, I.W.: Learning with augmented features for heterogeneous domain adaptation. In: Proceedings of the International Conference on Machine Learning, pp. 711–718. Omnipress, Edinburgh, June 2012
13. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.-R., Lin, C.J.: LIBLINEAR: a library for large linear classification. JMLR **9**, 1871–1874 (2008)
14. Farquhar, J., Hardoon, D.R., Meng, H., Shawe-Taylor, J., Szedmak, S.: Two view learning: SVM-2K, theory and practice. In: NIPS (2006)

15. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: IEEE ICCV, pp. 2960–2967 (2013)
16. Fernando, B., Tommasi, T., Tuytelaarsc, T.: Joint cross-domain classification and subspace learning for unsupervised adaptation. Pattern Recognit. Lett. **65**, 60–66 (2015)
17. Feyereisl, J., Kwak, S., Son, J., Han, B.: Object localization based on structural SVM using privileged information. In: NIPS (2014)
18. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV (2009)
19. Goldfarb, D., Ma, S., Scheinberg, K.: Fast alternating linearization methods for minimizing the sum of two convex functions. Math. Program. **141**(1–2), 349–382 (2013)
20. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML (2013)
21. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2066–2073. IEEE (2012)
22. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: an unsupervised approach. In: IEEE ICCV, pp. 999–1006 (2011)
23. Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: NIPS (2006)
24. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report, California Institute of Technology (2007)
25. Hardoon, D.R., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: an overview with application to learning methods. Neural Comput. **16**, 2639–2664 (2004)
26. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS (2006)
27. Huynh, T., Min, R., Dugelay, J.: An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data. In: ACCV Workshops, pp. 133–145 (2012)
28. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: IEEE ICRA (2011)
29. Lapin, M., Hein, M., Schiele, B.: Learning using privileged information: SVM+ and weighted SVM. Neural Netw. **53**, 95–108 (2014)
30. Li, W., Niu, L., Xu, D.: Exploiting privileged information from web data for image categorization. In: ECCV, pp. 437–452 (2014)
31. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10), 2756–2779 (2007)
32. Motiian, S., Piccirilli, M., Adjeroh, D., Doretto, G.: Information bottleneck learning using privileged information for visual recognition. In: IEEE CVPR, pp. 1496–1505 (2016)
33. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
34. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML (2011)
35. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE TNN **22**(2), 199–210 (2011)

36. Pechyony, D., Vapnik, V.: On the theory of learning with privileged information. In: NIPS (2010)
37. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV, pp. 213–226 (2010)
38. Sharmanska, V., Quadrianto, N., Lampert, C.: Learning to rank using privileged information. In: IEEE ICCV, pp. 825–832 (2013)
39. Shi, Y., Sha, F.: Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: ICML (2012)
40. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. J. Stat. Plann. Infer. **90**(2), 227–244 (2000)
41. Slonim, N., Friedman, N., Tishby, N.: Multivariate information bottleneck. Neural Comput. **18**(8), 1739–1789 (2006)
42. Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: Allerton Conference on Communication, Control, and Computing, pp. 368–377 (1999)
43. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1521–1528 (2011)
44. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV (2015)
45. Vapnik, V., Vashist, A.: A new learning paradigm: learning using privileged information. Neural Netw. **22**(5–6), 544–557 (2009)
46. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV, pp. 606–613, September 2009
47. Wang, Z., Ji, Q.: Classifier learning with hidden information. In: CVPR, pp. 4969–4977, June 2015
48. Wolf, L., Hassner, T., Taigman, Y.: Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. IEEE TPAMI **33**(10), 1978–1990 (2011)
49. Xu, C., Tao, D., Xu, C.: Large-margin multi-view information bottleneck. IEEE TPAMI **36**(8), 1559–1572 (2014)
50. Yao, T., Pan, Y., Ngo, C.W., Li, H., Mei, T.: Semi-supervised domain adaptation with subspace learning for visual recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
51. Zhang, C., He, J., Liu, Y., Si, L., Lawrence, R.D.: Multi-view transfer learning with a large margin approach. In: KDD (2011)