Information Bottleneck Learning Using Privileged Information for Visual Recognition

Saeid Motiian Marco Piccirilli Donald A. Adjeroh Gianfranco Doretto
West Virginia University
Morgantown, WV 26508
{samotiian, mpiccir1, daadjeroh, gidoretto}@mix.wvu.edu

Abstract

We explore the visual recognition problem from a main data view when an auxiliary data view is available during training. This is important because it allows improving the training of visual classifiers when paired additional data is cheaply available, and it improves the recognition from multi-view data when there is a missing view at testing time. The problem is challenging because of the intrinsic asymmetry caused by the missing auxiliary view during testing. We account for such view during training by extending the information bottleneck method, and by combining it with risk minimization. In this way, we establish an information theoretic principle for leaning any type of visual classifier under this particular setting. We use this principle to design a large-margin classifier with an efficient optimization in the primal space. We extensively compare our method with the state-of-the-art on different visual recognition datasets, and with different types of auxiliary data, and show that the proposed framework has a very promising potential.

1. Introduction

Large amounts of good quality labeled data for training visual classifiers are hard to obtain because they might be expensive, or require too much time to be collected, or because of other reasons. Typically, this problem is addressed by injecting domain, or prior knowledge, into the modeling framework to regularize the learning, and obtain a better classifier [22]. On the other hand, there are situations where training labeled samples might be easily augmented with extra information. For instance, in object recognition, a labeled image sample, representing the *main* data view, might have been annotated also with attributes describing semantic properties of depicted objects, or with a bounding box that specifies the location of the target object, or with image tags describing the context of the image. This extra information can be seen as an *auxiliary* data view of the image sample. In this work, we aim at improving visual recognition based on a main data view, by leveraging the auxiliary view available only during training, thus mitigating the lack of good quality labeled data. See Figure 1.

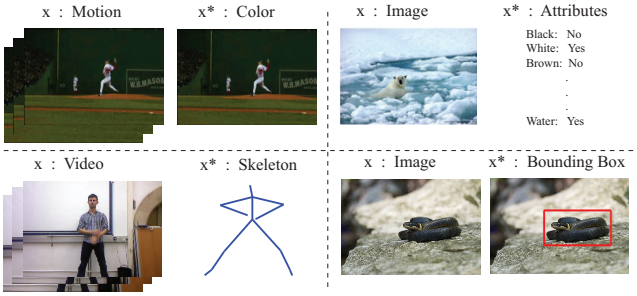

Figure 1. **Visual recognition with auxiliary data.** Visual recognition entails learning classifiers based on a *main* data view (e.g., motion information for recognizing actions, or image information for recognizing animals and objects, or video information for gesture recognition). We extend the information bottleneck method to leverage an *auxiliary* data view during training (e.g., color for actions, skeleton data for gestures, attributes for animals, and bounding boxes for objects), for learning a better visual classifier.

The problem outlined above has received limited attention. It is different from domain adaptation and transfer learning [1, 48, 32], where the source and target domains are closely related but statistically different. Here instead, the main view used for testing is present also in training, along with the paired auxiliary view to form the source domain. Indeed, our problem is more related to multi-view and multi-task learning [9, 40, 14, 28, 45]. However, 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. The fact that the auxiliary view is missing is what makes this problem challenging, because it cannot be combined like the others in multi-view learning.

We address the auxiliary view problem from an information theoretic perspective, where we learn how to extract information from the main data view, in a way that is optimal for visual recognition, and that speaks also on behalf of the missing auxiliary view. The information bottleneck (IB)

method [36] is a tool for extracting *latent information* from the main view, in a way that satisfies two complementary goals. The first is to compress the data as much as possible. The second is to preserve all the information that is *relevant* for the task at hand (e.g., predicting the labels of a visual recognition task). However, the IB method is not directly applicable to our problem because the latent information is not extracted in a way that speaks also on behalf of the auxiliary view. Therefore, our first contribution is to extend the IB method to take that aspect into account. Since the auxiliary view is not available at testing time, it was named *privileged* in [38], which first formalized this learning paradigm. Thus, we refer to our IB extension as the *information bottleneck method with privileged information (IBPI)*.

The IBPI method is a sound information theoretic principle for explicitly extracting relevant latent information, but gives an implicit, hence computationally hard, way for learning a visual classifier based on such information. Our second contribution is a modified version of IBPI that allows learning explicitly any type of visual classifier based on risk minimization. Our third contribution is the application of the modified IBPI method for learning a large-margin classifier, called *large-margin IBPI (LMIBPI)*, for which it is possible to use kernels, and for which we provide an optimization procedure guaranteed to converge in the primal space for improved computational efficiency.

Our fourth contribution is an extensive validation of LMIBPI against the state-of-the-art. We perform experiments where we improve visual recognition of gestures by training with auxiliary 3D joint information, we improve object classification with auxiliary object bounding box information, we improve animal recognition with auxiliary attribute information, and we improve action recognition with auxiliary visual features.

2. Related work

In computer vision auxiliary information has been incorporated into the learning process in several forms. For example, in attribute based approaches [11, 8] labeled data is used for training extra attribute classifiers to extract mid-level features. Similarly, [37, 23] learn to extract mid-level features by training data from additionally annotated concepts. Our framework differs from those because the auxiliary information can be generic, and because it is used for improving the classifier performance in a single optimization framework, whereas attribute classifiers may be disconnected from the main classification task. Another form of auxiliary information is given by the structure of the hidden domain of latent models for object detection and gesture recognition [10, 30]. Compared to those approaches we use information from auxiliary labeled data.

More related to our framework are the approaches that consider the auxiliary information to be supplied by a teacher during training. This is the learning using privileged information (LUPI) paradigm introduced in [38]. One LUPI implementation is the SVM+ [38, 21], which uses the privileged data as a proxy for predicting the slack variables. This is equivalent to learning an oracle that tells which sample is easy and which one is hard to predict. The same idea has been used in the learning to rank approach introduced in [33], where it is shown that different types of privileged information, such as bounding boxes, attributes, text, and annotator rationales [7] can be used for learning a better classifier for object recognition.

Our approach exploits the privileged information differently. An information theoretic framework learns how to compress the source domain for doing prediction in a way that is as informative of the privileged source domain as possible, regardless of the type of classifier used, and without tying privileged information to slack variables. This is done by extending the original IB method [36], often used for clustering [35], and also used in [3] for incorporating "negative information" that is irrelevant for the task at hand, and that should not be learned by the representation. This is similar to [44], where negative information is used for face recognition with discounted pose-induced similarity.

The LUPI paradigm has recently been used for boosting [4], for object localization in a structured prediction framework [12], for facial feature detection [33], for metric learning [13, 46], in a logistic classification framework [43], in a max-margin latent variable model [41], and in support of domain adaptation applications [5, 24]. In the above methods, either the problem settings, or the approaches taken are significantly different from the information theoretic principles that are driving our program. Other recent approaches include [50, 49], which focus on the missing view problem by discriminatively learning projections to a shared latent subspace. This approach relates more to multi-view learning, but only the main view pipeline is used for testing, without considering the intrinsic asymmetry of the LUPI framework, as pointed out in [42]. There they propose two principles to learn with auxiliary information based on looking at it as additional features, or as additional labels, where they make assumptions on its informative content. We also introduce a new principle that shares the benefits of their framework, but by using an information theoretic approach we have no need to make distinctions between the types of auxiliary information, and we have no need to state requirements on the information content.

3. Learning using privileged information

Traditional supervised learning assumes that a training dataset made of N pairs $(x_1, y_1), \cdots, (x_N, y_N)$ is given, where the feature $x_i \in \mathcal{X}$ is a realization from a random variable X, the label $y_i \in \mathcal{Y}$ is a realization from a random variable Y, and the pairs are i.i.d. samples from a joint probability distribution $p(X, Y)$. Under this setting the goal is to learn a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y}$ by searching

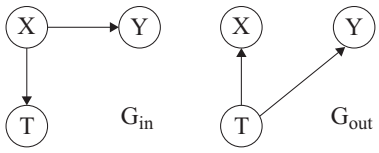

Figure 2. **Information Bottleneck.** Structural representation of G_{in} and G_{out} used by the original two-variable information bottleneck method [36].

over a space of admissible functions $\mathcal{F}$.

The Learning Using Privileged Information (LUPI) paradigm as defined in [38] assumes that every training data pair comes with *auxiliary* information, augmenting the training dataset to $(x_1, x_1^*, y_1), \cdots, (x_N, x_N^*, y_N)$. The auxiliary feature $x_i^* \in \mathcal{X}^*$ is a realization from the random variable X^*. The triplets are now i.i.d. samples from the joint distribution $p(X, X^*, Y)$. Under LUPI settings, the goal is to learn a prediction function $f^* : \mathcal{X} \to \mathcal{Y}$ by searching $\mathcal{F}$. Note that in order to predict a label y, at testing time f^* uses only data from the *main* space $\mathcal{X}$. Therefore, the data from the auxiliary space $\mathcal{X}^*$ is only available during training, which is why it is called *privileged*. From the same amount of training samples N, the LUPI classifier f^* will improve the performance of the traditional classifier f [29]. On the other hand, how to *best* exploit privileged information for learning f^* remains an open problem.

4. The information bottleneck method

We summarize the information bottleneck (IB) method [36] that was extended to the multivariate case in [34]. We are given a set of random variables $\mathbf{X} = \{X_1, \cdots, X_n\}$, distributed according to a known $p(\mathbf{X})$, a set of *latent* variables $\mathbf{T} = \{T_1, \cdots, T_k\}$, and a Bayesian network with graph G_{in} over $\mathbf{X} \cup \mathbf{T}$, defining which subset of $\mathbf{X}$ is compressed by which subset of $\mathbf{T}$. Another Bayesian network, G_{out}, also defined over $\mathbf{X} \cup \mathbf{T}$, is given and represents which conditional dependencies and independencies we desire $\mathbf{T}$ to be able to generate. The joint distribution $q(\mathbf{X}, \mathbf{T}) \doteq q(\mathbf{T}|\mathbf{X})p(\mathbf{X})$ is unknown.

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.* This idea is implemented with the *multi-information* of $\mathbf{X}$, which is the information shared by $X_1, \cdots, X_n$, i.e.,

$$\mathcal{I}(\mathbf{X}) = D_{KL}[p(\mathbf{X}) \| p(X_1) \cdots p(X_n)] \,, \tag{1}$$

where D_{KL} indicates the Kullback-Leibler divergence [6]. Therefore, the multivariate IB method looks for $q(\mathbf{T}|\mathbf{X})$ that minimizes the functional

$$\mathcal{L}[q(\mathbf{T}|\mathbf{X})] = \mathcal{I}^{G_{in}}(\mathbf{X}, \mathbf{T}) + \gamma(\mathcal{I}^{G_{in}}(\mathbf{X}, \mathbf{T}) - \mathcal{I}^{G_{out}}(\mathbf{X}, \mathbf{T})) \tag{2}$$

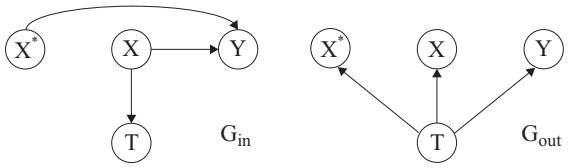

Figure 3. **Information Bottleneck with Privileged Information.** Structural representation of G_{in} and G_{out} used by the information bottleneck method with privileged information.

where γ strikes a balance between compression and the ability to satisfy the independency requirements of G_{out}. The multi-information $\mathcal{I}^G$ with respect to a Bayesian network G defined over $\mathbf{X} \sim p(\mathbf{X})$ is computed as in [34], i.e.,

$$\mathcal{I}^G(\mathbf{X}) = \sum_i I(X_i; \mathbf{Pa}_{X_i}^G) \,, \tag{3}$$

where $I(X_i; \mathbf{Pa}_{X_i}^G)$ is the mutual information between X_i and $\mathbf{Pa}_{X_i}^G$, the set of variables that are parents of X_i in G.

Let us refer to Figure 2 for an example, where $\mathbf{X} = \{X, Y\}$, and $\mathbf{T} = T$. 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 T, and then predicting Y from T. In G_{in} the edge $X \to Y$ indicates the relation defined by $p(X, Y)$. Moreover, since T will compress X, this is indicated by the edge $X \to T$, establishing that T is completely determined given the variable it compresses. The graph G_{out} instead, reflects the idea that we would like T to capture from X all the necessary information to perform the best possible prediction of Y. This means that knowing T makes X and Y independent, or equivalently that $I(X; Y|T) = 0$. To evaluate (2), instead, we obtain $\mathcal{I}^{G_{in}} = I(T; X) + I(Y; X)$, and $\mathcal{I}^{G_{out}} = I(X; T) + I(Y; T)$, and since $I(Y; X)$ is constant, (2) collapses to the original two-variable IB method [36].

5. IB with privileged information

Here we combine the ideas of Sections 3 and 4 for developing a new information bottleneck principle, which accounts for privileged information. Specifically, let us assume that X, X^*, and Y are three random variables with known distribution $p(X, X^*, Y)$. Also, it is assumed that both X and X^* contain information about Y. If properly extracted, such information could be used for predicting Y. However, we assume that only the information carried by X can be used to predict Y. We pose the question of whether by doing so it is still possible to learn a model capable of exploiting the information carried by X^*.

If we apply the two-variable IB method, we proceed by compressing X 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 two graphs G_{in} and G_{out} in Figure 2. On the other hand, since X^* has knowledge about Y, a more complete Bayesian network

representing all the variables and the compression requirements, is the graph G_{in} in Figure 3, which includes the connection $X^* \rightarrow Y$. Therefore, the optimal representation computed by the two-variable IB method would be given by $q(X, X^*, Y, T) = q(T|X)p(X, X^*, Y)$, where $q(T|X)$ is such that $I(X;Y|T)$ is as close to zero as possible.

We note that the approach outlined above does not make any effort to exploit the information carried by X^*. Indeed, $I(X^*;Y|T)$ could be arbitrarily high, i.e., knowing T still leaves with X^* substantial knowledge about Y. On the other hand, the multivariate IB method allows us to consider more complex independency structures. In particular, we define G_{out} like in Figure 3, where knowing T not only makes X and Y independent, but X^* and Y too. In this way, $q(T|X)$ not only minimizes $I(X;Y|T)$, but also $I(X^*;Y|T)$. More precisely, the multi-informations of G_{in} and G_{out} in Figure 3 are given by

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

$$\mathcal{I}^{G_{out}} = I(T;X) + I(T;X^*) + I(T;Y) \ . \tag{5}$$

By plugging (4) and (5) into (2), since $I(Y;X,X^*)$ is constant, the functional for learning the optimal representation for T is given by

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

where γ strikes a balance between compressing X and imposing the independency requirements. Similarly to the LUPI framework, since it is not possible to directly compress X^* for predicting Y, we can think of X^* as carrying *privileged information* about Y. Therefore, we call learning representations by minimizing (6) as the *information bottleneck method with privileged information (IBPI)*.

6. IBPI for visual recognition

We are interested in designing a framework for visual recognition, where we need to perform a classification task based on a *main* view X of the visual data. However, at training time, for some training samples an *auxiliary* view X^* is also available. We pose no restrictions on the type of auxiliary data available. The task at hand falls into the LUPI category defined in Section 3, except that we also admit training samples with missing auxiliary view.

We want to leverage the IBPI method (6) because it provides a sound principle, grounded on information theory, for extracting information T from the main view X that is not only the most relevant for predicting Y (representing class labels), but also minimizes $I(X^*;Y|T)$, which means that knowing T leaves with X^* minimal information about Y. This suggests that T is the representation of choice for predicting Y. However, similarly to the IB method [34], while IBPI explicitly defines the compression map, T, by searching for $q(T|X)$, the computation of $q(Y|T)$ is much harder in general. For this reason, we introduce a modified IBPI method that is tailored to visual recognition.

Algorithm 1 Projected gradient minimization for F

Chose $0 < \eta < 1, 0 < \nu < 1$.
Initialize F^1. Set $\rho = 1$.
for $k = 1, 2, \cdots$ **do**
  if ρ satisfies (11) **then**
    Repeatedly increase it by $\rho \leftarrow \rho/\eta$ until either ρ does not satisfy (11) or $F(\rho/\eta) = F(\rho)$
  else
    Repeatedly decrease ρ by $\rho \leftarrow \rho/\eta$ until ρ satisfies (11)
  end if
  Set $F^{k+1} = \max\{0, F^k - \rho \nabla_F D_{KL}(\bar{X} \| F^k \bar{X}^*)\}$
  Normalize to 1 the columns of F^{k+1}
end for

We observe that by interpreting γ as a Lagrange multiplier, the last term in (6) corresponds to the constraint $I(T;Y) \geq constant$, enforcing T of carrying at least a certain amount of information about Y. Ultimately, such information should be used for classification purposes, by predicting Y through a function $\tilde{f} : \mathcal{T} \rightarrow \mathcal{Y}$. Therefore, we replace the constraint on $I(T;Y)$ with the risk associated to $\tilde{f}(T)$ according to a loss function ℓ. Thus, for visual recognition, (6) is modified into

$$\boxed{\mathcal{L}[q(T|X), \tilde{f}] = I(T;X) - \gamma I(X^*;T) + \beta E[\ell(\tilde{f}(T), Y)]} \tag{7}$$

where $E[\cdot]$ denotes statistical expectation, and β balances the risk versus the compression requirements. Note that the modified IBPI criterion (7) is general, and could be used with any classifier. Obviously, a practical implementation of (7) would be based on the empirical risk.

6.1. Large-margin IBPI

We use (7) to develop a large-margin classifier. We focus on the binary case to prove the validity of the framework by comparing it with the state-of-the-art, which also focussed on the binary case. In particular, we restrict the search space for $q(T|X)$ by assuming $T = \phi(X;A)$, where A is a suitable set of parameters. Moreover, $\tilde{f}(T)$ is a binary decision function given by $Y = \text{sign}(\langle w, T \rangle + b)$, where $\langle \cdot, \cdot \rangle$ identifies a dot product, w defines the margin, and b is an offset. Therefore, by using the hinge loss function, from (7) we derive the following classifier learning formulation, which we refer to as the *large-margin IBPI (LMIBPI)*

$$\begin{aligned} \min_{A,w,b,\xi_i} \ & I(T;X) - \gamma I(X^*;T) + \frac{\beta}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \ & y_i(\langle w, \phi(x_i, A)\rangle + b) \geq 1 - \xi_i \ , \\ & \xi_i \geq 0 \ , \quad \forall i \in \{1, \cdots, N\} \ . \end{aligned} \tag{8}$$

where C is the usual parameter to control the slackness.

Kernels. We set $T = \phi(X, A) = A\phi(X)$, where we require $\phi(X)$ to have positive components and be normalized to 1, and A to be a stochastic matrix, made of conditional probabilities between components of $\phi(X)$ and T.

Algorithm 2 FALM for LMIBPI

Chose $\mu_f > 0$ and $\mu_g > 0$ and $A^0 = B^0 = E^1$, set $t_1 = 1$
for $k = 1, 2, \cdots$ **do**
  $A^k = \arg\min_{0 \le A \le 1} Q_g(A, E^k)$
  $B^k = \arg\min_{0 \le B \le 1} Q_f(B, A^k)$
  $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
  $E^{k+1} = B^k + \frac{t_k - 1}{t_{k+1}}(B^k - B^{k-1})$
end for

This assumption greatly simplifies computing mutual informations. X can be mapped to a feature space with $\psi(X)$. In this case we set $\phi(X) = \rho(\Psi\psi(X))^\top$, where $\Psi = [\psi(x_1), \cdots, \psi(x_N)]$, and $\rho(\cdot)$ is the additive logistic transformation that maps $u \in \mathbb{R}^N$ to the $N+1$ dimensional simplex $v = \left[\frac{e^{u_1}}{1+\sum_i e^{u_i}}, \cdots, \frac{e^{u_N}}{1+\sum_i e^{u_i}}, \frac{1}{1+\sum_i e^{u_i}}\right]$, with positive components and normalized to 1. Thus, without loss of generality, in the sequel we set $T = AX$. X^* can be mapped to a feature space $\varphi(X^*)$ with the same strategy.

Mutual informations. $I(T; X)$ is given by

$$I(T; X) = E\left[\sum_{i,j} A(i,j)X(j) \log \frac{A(i,j)}{T(i)}\right] \quad (9)$$

where $A(i,j)$ is the entry of A in position i, j, whereas $T(i)$ and $X(j)$ are the components in position i and j of T and X respectively. Obviously, during training the expectation is replaced by the empirical average.

To compute $I(T; X^*)$, let $t(i)$, $x^*(j)$, and $x(h)$ be histogram realizations for T, X^*, and X, where i, j, and h index the histogram bins. The mutual information $I(t, x^*)$ is given by $\sum_{i,j} p(i,j) \log \frac{p(i,j)}{t(i)x^*(j)}$. By the low of total probability, $p(i,j)$ is rewritten as $\sum_h p(i|j,h)p(h|j)x^*(j)$, where $p(i|j,h) = p(i|h) = A(i,h)$ because $t(i)$ is completely defined by $x(h)$. In addition, we call $F(h,j) = p(h|j)$, from which it follows that $X = FX^*$, where F is also a stochastic matrix. Therefore, $I(T; X^*)$ is given by

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

Learning F. F is learned from training data. Specifically, let's indicate with $\bar{X} = [x_1, \cdots, x_N]$ and $\bar{X}^* = [x_1^*, \cdots, x_N^*]$ the training data points corresponding to the main and privileged domains, then F is learned by solving the following constrained optimization problem: $\min_F D_{KL}(\bar{X} \| F\bar{X}^*)$ s.t. F is a stochastic matrix with normalized columns. We compute F with Algorithm 1, which is a projected gradient method [26] with Armijo's condition

$$\begin{aligned} &D_{KL}(\bar{X} \| F^{k+1}\bar{X}^*) - D_{KL}(\bar{X} \| F^k \bar{X}^*) \quad (11) \\ &\qquad \le \nu \langle \nabla_F D_{KL}(\bar{X} \| F^k \bar{X}^*), F^{k+1} - F^k \rangle \end{aligned}$$

where k is the iteration index. The computation of $\nabla_F D_{KL}(\bar{X} \| F\bar{X}^*)$ is fairly simple, and can be found as a special case in [47].

Algorithm 3 Projected gradient minimization for Q_f or Q_g

Chose $0 < \eta < 1$, $0 < \nu < 1$.
Initialize A^1 for Q_g (or B^1 for Q_f). Set $\rho = 1$.
for $k = 1, 2, \cdots$ **do**
  if ρ satisfies (16) for Q_g (or (17) for Q_f) **then**
    Repeatedly increase it by $\rho \leftarrow \rho/\eta$ until either ρ does not satisfy (16) (or (17)) or $A(\rho/\eta) = A(\rho)$ (or $B(\rho/\eta) = B(\rho)$)
  else
    Repeatedly decrease ρ by $\rho \leftarrow \rho/\eta$ until ρ satisfies (16) (or (17))
  end if
  Set $A^{k+1} = \max\{0, A^k - \rho\nabla_A Q_g(A^k, B)\}$
  (Set $B^{k+1} = \max\{0, B^k - \rho\nabla_B Q_f(B^k, A)\}$)
  Normalize to 1 the columns of A^{k+1} (or B^{k+1})
end for

Missing auxiliary views. Training samples with missing auxiliary view affect only $I(T; X^*)$. The issue is seamlessly handled by estimating F and the average in (10) by using only the samples that have the auxiliary view.

6.1.1 Optimization

When A is known, (8) is a soft-margin SVM problem. Instead, when the SVM parameters are known, (8) becomes

$$\begin{aligned} \min_A \quad & I(T; X) - \gamma I(X^*; T) + \frac{C}{N}\sum_{i=1}^N \xi_i \quad (12) \\ \text{s.t.} \quad & \xi_i = \max\{0, 1 - y_i(\langle w, \phi(x_i, A)\rangle + b)\} \ . \end{aligned}$$

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 $q(T|X)$ [6]. 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 $f(A) = I(T; X) - \gamma I(X^*; T)$, $g(B) = \frac{C}{N}\sum_{i=1}^N \xi_i$, and then solving $\min_A\{f(A) + g(B) : A - B = 0\}$.

For smoothing the hinge loss we use the Nesterov smoothing technique [27], used also in [51], which requires choosing a proximal function, and then computing the smoothed slack variables in this way $\xi_{i,\sigma} = \max_{0 \le u_i \le 1} u_i(1 - y_i w^\top A x_i) - \frac{\sigma}{2}\|wx_i^\top\|_\infty u_i^2$, which gives

$$\xi_{i,\sigma} = \begin{cases} 0 & y_i w^\top A x_i > 1 \ , \\ (1 - y_i w^\top A x_i) - \frac{\sigma}{2}\|wx_i^\top\|_\infty & y_i w^\top A x_i < 1 \\ & -\sigma\|wx_i^\top\|_\infty, \\ \frac{(1 - y_i w^\top A x_i)^2}{2\sigma\|wx_i^\top\|_\infty} & \text{otherwise.} \end{cases} \quad (13)$$

where σ is a smoothing parameter. In this way, the minimization can be carried out with the Fast Alternating Linearization Method (FALM) [15]. This allows simpler computations, and has performance guarantees when ∇f and ∇g are Lipschitz continuous, which is the case, given the smoothing technique that we used.

	LB-LMIBPI	SVM	SVM-R	RankTr	SVM+	SVM2k-LUPI	KCCA-LUPI	LMIBPI	UB-LMIBPI	SVM2k	KCCA
brush hair	78.49 ± 4.99	78.50 ± 6.68	79.67 ± 4.25	78.16 ± 4.40	79.17 ± 6.40	77.50 ± 5.78	77.00 ± 6.79	$\mathbf{80.66 \pm 4.85}$	84.48 ± 5.03	85.00 ± 5.93	78.00 ± 8.08
dive	78.83 ± 4.23	80.66 ± 3.00	73.63 ± 4.11	79.66 ± 7.97	79.83 ± 3.64	82.50 ± 2.63	83.00 ± 2.33	$\mathbf{83.16 \pm 5.23}$	90.79 ± 5.94	87.16 ± 3.14	85.00 ± 2.07
drink	69.37 ± 6.62	69.16 ± 6.00	68.5 ± 5.43	$\mathbf{75.50 \pm 6.18}$	69.17 ± 5.80	69.50 ± 6.13	68.50 ± 5.41	72.36 ± 5.83	81.98 ± 7.48	74.33 ± 7.16	69.83 ± 6.20
eat	67.04 ± 5.86	67.66 ± 4.00	69.63 ± 5.00	$\mathbf{75.08 \pm 2.10}$	71.00 ± 5.16	71.50 ± 6.35	67.50 ± 6.58	74.85 ± 4.61	81.98 ± 4.95	76.00 ± 6.62	69.00 ± 6.19
golf	78.83 ± 3.68	81.50 ± 4.11	70.43 ± 4.02	74.66 ± 3.01	80.67 ± 7.25	80.00 ± 6.23	78.66 ± 4.76	$\mathbf{85.47 \pm 5.66}$	90.45 ± 3.68	85.33 ± 6.92	78.16 ± 5.29
hug	80.66 ± 3.35	81.83 ± 4.18	80.43 ± 4.66	82.82 ± 3.90	81.50 ± 3.72	83.33 ± 5.03	81.33 ± 4.83	$\mathbf{85.97 \pm 3.44}$	91.11 ± 5.61	87.83 ± 3.93	82.66 ± 4.91
jump	74.16 ± 3.70	74.16 ± 4.30	69.90 ± 2.87	75.66 ± 2.81	76.33 ± 6.42	77.00 ± 6.17	76.16 ± 6.03	$\mathbf{79.83 \pm 2.65}$	84.52 ± 3.62	81.16 ± 4.90	78.33 ± 7.37
pick	65.59 ± 3.88	63.83 ± 5.42	67.53 ± 3.67	78.33 ± 3.56	66.83 ± 4.19	65.16 ± 5.52	63.83 ± 5.15	$\mathbf{79.83 \pm 3.56}$	85.42 ± 3.79	68.50 ± 3.88	64.33 ± 6.09
punch	83.76 ± 3.20	83.66 ± 4.10	81.20 ± 5.72	87.33 ± 4.90	84.33 ± 5.89	84.83 ± 4.47	83.33 ± 4.37	$\mathbf{92.06 \pm 3.54}$	95.38 ± 3.54	86.16 ± 4.30	83.83 ± 4.23
sit	75.33 ± 3.10	75.16 ± 4.83	71.66 ± 9.8	76.33 ± 3.67	74.00 ± 4.60	75.16 ± 5.95	73.50 ± 5.05	$\mathbf{76.99 \pm 3.58}$	85.29 ± 4.05	77.50 ± 5.78	75.33 ± 5.92

Table 1. **HMDB dataset.** Classification accuracies for one-vs-all binary classifications. The HOF features represent the main view, and the HOG features the auxiliary view. Best accuracies are highlighted in boldface.

FALM splits the minimization of the augmented Lagrangian function into two simpler functions to be minimized alternatively, which are given by

$$Q_g(A, B) = f(A) + g(B) + \langle \nabla g(B), A - B \rangle + \tfrac{1}{\mu_g} D_{KL}(A||B) \quad (14)$$

$$Q_f(B, A) = f(A) + g(B) + \langle \nabla f(A), B - A \rangle + \tfrac{1}{\mu_g} D_{KL}(B||A) \quad (15)$$

The FALM iteration is given in Algorithm 2. Since A is a stochastic matrix, the KL-divergence regularization is used in place of the squared Frobenius norm.

Note that lines 3 and 4 of Algorithm 2 are constrained optimizations, requiring A and B to be stochastic matrices with normalized columns. They are implemented by Algorithm 3, a projected gradient method [26] with Armijo's rule that for Q_g and Q_f is given by

$$Q_g(A^{k+1}, B) - Q_g(A^k, B) \leq \nu \langle \nabla_A Q_g(A^k, B), A^{k+1} - A^k \rangle \quad (16)$$

$$Q_f(B^{k+1}, A) - Q_f(B^k, A) \leq \nu \langle \nabla_B Q_f(B^k, A), B^{k+1} - B^k \rangle \quad (17)$$

where k is the iteration index. From (13), (9), (10) it is straightforward to compute $\nabla_A Q_g$, and $\nabla_B Q_f$. We leave those expressions out due to the limited space.

7. Experiments

We have performed experiments with four different datasets. With each dataset we train and test the following binary classifiers.

Single-view classifiers: Using only the main view, we train the SVM-Light [17] (indicated as `SVM`), the SVM-Rank [18] (indicated as `SVM-R`), and LMIBPI where we eliminate the use of auxiliary information by setting $\gamma = 0$ (indicated as `LB-LMIBPI`).

LUPI classifiers: We train the SVM+ [38] (indicated as `SVM+`, implemented by [25]), the Rank Transfer [33] (indicated as `RankTr`, and reimplemented by us), and our LMIBPI approach (indicated as `LMIBPI`). We also train the SVM2k [9] and test only the SVM that uses the main view (indicated as `SVM2k-LUPI`), and we perform kernel CCA (KCCA) [16] between main and auxiliary views, map the main view in feature space and train an SVM (indicated as `KCCA-LUPI`).

Two-view classifiers: Using main and auxiliary views, we train the SVM2k (indicated as `SVM2k`), and we also use KCCA between views to map them in feature space, train two SVMs and average the outputs (indicated as `KCCA`). Finally, we also extend LMIBPI (details are omitted for lack of space) to fuse main and auxiliary views (indicated as `UB-LMIBPI`). Note that for these classifiers main and auxiliary views are used also during testing. So, their performances represent the upper bound for the corresponding LUPI versions.

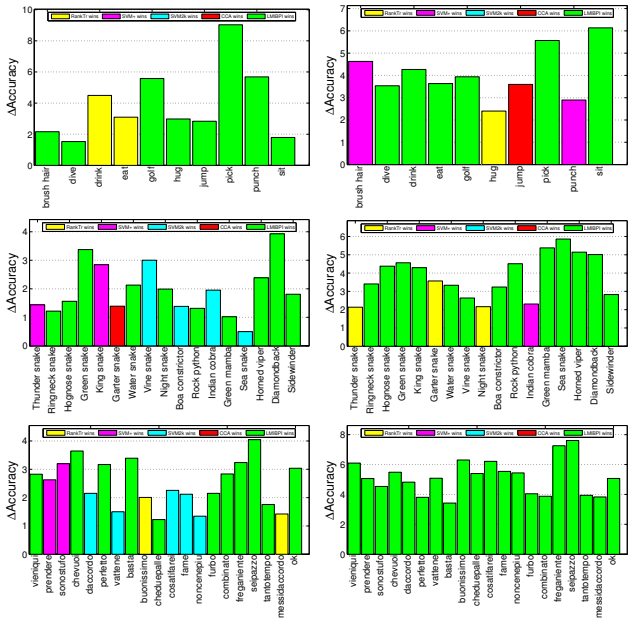

Figure 4. **Linear vs. non-linear kernel.** Plots representing the differences between the classification accuracy of the winner LUPI method against the average accuracy over the following methods: `RankTr` (yellow), `SVM+` (magenta), `SVM2k-LUPI` (cyan), `KCCA-LUPI` (red), and `LMIBPI` (green). The linear kernel was used on the left plots, and the histogram intersection kernel on the right plots. The top row come from the HMDB dataset, the middle row from ImageNet, and the last row from CGD2011.

Model selection: We use the same joint cross validation and model selection procedure described in [33], 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 m, 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 $\{0.1, 0.3, 0.5\}$, and the m's in the range $\{50, 70, 90\}$.

Performance: For each binary classification experiment we

	LB-LMIBPI	SVM	SVM-R	RankTr	SVM+	SVM2k-LUPI	KCCA-LUPI	LMIBPI	UB-LMIBPI	SVM2k	KCCA
Thunder snake	56.84 ± 3.21	57.09 ± 3.14	52.42 ± 2.53	57.88 ± 3.57	**60.17 ± 2.29**	59.12 ± 2.00	56.79 ± 2.33	59.70 ± 2.62	63.31 ± 3.23	61.70 ± 1.35	57.05 ± 2.39
Ringneck snake	62.03 ± 2.62	63.31 ± 2.76	53.55 ± 3.78	62.25 ± 1.46	63.14 ± 2.45	63.93 ± 2.82	63.47 ± 2.03	**64.72 ± 2.36**	68.36 ± 2.65	67.49± 2.43	64.46 ± 2.55
Hognose snake	57.71 ± 1.56	60.11 ± 1.34	55.74 ± 2.36	55.33 ± 2.77	59.10 ± 1.81	59.73 ± 2.10	60.55 ± 0.96	**60.63 ± 1.58**	65.32 ± 4.25	61.53± 1.36	60.03 ± 1.32
Green snake	68.11 ± 2.85	71.46 ± 1.39	55.43 ± 6.54	62.20 ± 2.99	70.66 ± 1.83	71.12 ± 1.40	70.03 ± 2.20	**72.72 ± 2.17**	77.82 ± 5.36	72.64± 1.45	68.80 ± 2.59
King snake	62.75 ± 1.57	60.20 ± 1.83	61.14 ± 3.53	59.70 ± 3.74	**63.84 ± 2.07**	59.81 ± 1.54	59.92 ± 2.01	61.70 ± 4.96	65.38 ± 2.36	64.89± 1.41	60.70 ± 1.78
Garter snake	66.72 ± 5.23	69.02 ± 3.25	57.07 ± 4.79	66.47 ± 2.58	66.02 ± 2.34	69.17 ± 2.90	**69.21 ± 2.86**	68.23 ± 1.79	70.23 ± 5.36	72.97± 2.37	69.65 ± 3.31
Water snake	70.26 ± 1.47	71.94 ± 1.91	64.80 ± 9.72	67.86 ± 3.40	68.82 ± 2.86	72.21 ± 1.83	71.34 ± 1.95	**72.72 ± 4.96**	73.21 ± 3.42	72.50± 2.13	70.11 ± 1.54
Vine snake	67.85 ± 3.52	78.92 ± 2.04	73.04 ± 5.00	69.97 ± 4.87	74.86 ± 2.09	**79.05 ± 2.14**	78.45 ± 1.95	77.91 ± 1.77	78.92 ± 4.02	80.15± 2.58	78.45 ± 1.72
Night snake	52.42 ± 6.32	53.97 ± 3.62	54.01 ± 2.25	52.96 ± 3.23	55.19 ± 1.76	55.26 ± 3.39	55.00 ± 3.44	**57.09 ± 2.14**	60.17 ± 4.23	55.48± 3.35	54.51 ± 3.32
Boa constrictor	61.90 ± 2.41	61.76 ± 1.87	59.03 ± 4.80	59.64 ± 3.40	62.66 ± 1.23	**62.92 ± 1.65**	61.60 ± 2.08	60.86 ± 1.72	63.76 ± 6.03	63.46± 2.10	61.19 ± 1.48
Rock python	57.88 ± 6.85	60.39 ± 2.36	56.84 ± 2.83	57.71 ± 2.59	58.43 ± 2.92	60.14 ± 2.46	59.50 ± 1.44	**60.59 ± 1.54**	61.14 ± 2.36	60.92± 2.16	58.78 ± 1.81
Indian cobra	61.90 ± 3.56	65.21 ± 2.84	59.04 ± 6.87	63.76 ± 3.92	65.88 ± 2.55	**66.88 ± 2.83**	64.92 ± 3.55	63.19 ± 2.97	64.53 ± 4.02	68.80± 1.52	65.42 ± 3.39
Green mamba	65.24 ± 1.69	68.50 ± 2.33	62.71 ± 9.11	66.77 ± 5.23	67.56 ± 2.39	68.36 ± 1.95	67.09 ± 2.47	**68.72 ± 3.50**	71.46 ± 5.36	70.14± 1.67	67.27 ± 2.16
Sea snake	72.72 ± 2.56	77.42 ± 1.64	68.36 ± 4.46	77.22 ± 2.03	76.17 ± 1.61	**77.55 ± 1.84**	77.10 ± 1.54	77.22 ± 1.70	78.53 ± 2.39	78.51± 1.41	73.06 ± 1.66
Horned viper	67.90 ± 1.49	69.92 ± 1.25	63.95 ± 5.17	67.74 ± 2.75	67.41 ± 1.94	69.49 ± 2.39	69.54 ± 1.01	**71.46 ± 3.21**	74.86 ± 6.45	71.80± 2.72	68.98 ± 1.31
Diamondback	64.27 ± 2.53	66.18 ± 2.56	61.90 ± 3.09	64.07 ± 2.60	63.48 ± 3.12	66.60 ± 1.81	65.89 ± 2.12	**69.92 ± 2.42**	71.94 ± 5.01	69.32± 1.59	65.69 ± 2.60
Sidewinder	67.41 ± 3.92	69.55 ± 2.34	60.28 ± 4.82	68.11 ± 3.21	67.85 ± 2.64	68.93 ± 5.54	68.70 ± 2.50	**70.66 ± 3.19**	72.77 ± 4.03	69.88± 7.25	66.46 ± 1.84

Table 2. **ImageNet dataset.** Classification accuracies for one-vs-all binary classifications. The BoW from the whole image is the main view, and the BoW from the bounding box region is the auxiliary view. Best accuracies are highlighted in boldface.

randomly select the same number of positive and negative samples for training, and the same for testing. Each experiment is repeated 10 times and average classification accuracy and standard deviation are reported.

Kernels: We use the linear and the histogram intersection (HIK). Due to space constraints we report table results for the linear case, and include figures for the HIK. Tables for more non-linear kernels are omitted for lack of space.

HMDB dataset: The HMDB dataset [19] is a video dataset for action recognition, composed of 51 classes. Each class has approximately 100 videos. We have randomly selected 10 classes, and we have considered the binary classification between one class versus the rest. With this experiment we test whether computing an auxiliary feature only during training, can be used to improve the recognition during testing. This would mean a performance improvement while saving computing power. For every video we extracted two bag-of-word (BoW) representations, one given by HOF descriptors, and one by HOG descriptors. We used dictionaries of size 400, learned with VLFeat [39]. We used 70 samples per class for training and 30 for testing. The HOF descriptors were set to the main view, and the HOG's represented auxiliary information. Table 1 collects the classification accuracies for the linear kernel. As expected, LUPI classifiers improve upon single-view, and `LMIBPI` outperforms the others 8 out of 10 times in the linear case, and 6 times with HIK. See Figure 4 (top row).

Time complexity: `LMIBPI` estimates F only once, and then iterates between optimizing A and a SVM. Both components are fast, also thanks to the derivation in the primal space. In addition, Figure 5 shows the accuracy convergence for the `drink` class of the HMDB dataset for different m's. We observed that less than 10 iterations were enough to reach convergence most of the time.

ImageNet dataset: We use the ImageNet [31] object categories of the 2012 challenge, also used in [33]. This subset has bounding box annotations, and we test whether they can improve recognition when used as auxiliary information. We use the group of snakes, which has 17 classes, for a total of 7746 images (some bounding boxes did not have images). For each sample we extracted a BoW from the entire image to be used as main view, and a BoW from the image portion in the bounding box to be used as auxiliary view. The descriptor used was dense SIFT [39] with a vocabulary size of 400. The classification task is between one snake class versus all the others. We use 200 samples per class for training and the rest for testing. Table 2 summarizes the classification accuracy results. Even here LUPI classifiers improve upon single-view, and `LMIBPI` outperforms the others 10 out of 17 times in the linear case, and 13 times with HIK. See Figure 4 (middle row).

CGD2011 dataset: The CGD2011 dataset [2] contains 20 gesture classes, each of which has about 400 RGB-D videos, along with skeleton tracking data. Since skeleton tracking is typically more expensive to obtain, we test whether by using it as auxiliary data it can boost performance. We perform one-vs-all classification with 100 samples per class for training and 90 for testing. We used a BoW with dictionary size 100 based on HOF features as main view. For the auxiliary view, from a video we extract a histogram of the joint positions, accumulated over all the frames of the sequence. Specifically, at every frame we place a spatial grid aligned with the head position of an individual and bin the position of each of the joints with respect to the grid. The resulting count is normalized and produces a histogram with 100 bins Table 3 shows the classification accuracies. The LUPI classifiers improve upon single-view, and `LMIBPI` outperforms the others 11 out of 20 times in the linear case, and all the times with HIK. See Figure 4 (bottom row).

AwA dataset: We use the Animals with Attributes (AwA) dataset [20], which contains images of animal categories, and repeat the same experiment performed in [33, 42]. We use the 10 test classes for which the attribute annotations are provided, for a total of 6180 images. The attributes capture 85 properties of the animals. We use the same set of features used in [33]. The main view is given by L_1 normalized 2000 dimensional SURF descriptors, and the attributes are the auxiliary view obtained from the DAP model [20]. We train 45 binary classifiers for each class pair combination.

	LB-LMIBPI	SVM	SVM-R	RankTr	SVM+	SVM2k-LUPI	KCCA-LUPI	LMIBPI	UB-LMIBPI	SVM2k	KCCA
vieniqui	52.89 ± 1.98	49.72 ± 4.96	52.83 ± 6.24	52.11 ± 4.83	50.27 ± 4.20	51.16 ± 4.50	48.33 ± 4.84	**54.00 ± 1.32**	56.50 ± 5.06	50.72 ± 2.99	52.66 ± 3.38
prendere	53.95 ± 4.23	52.39 ± 3.00	56.38 ± 4.10	54.50 ± 3.87	**58.05 ± 2.34**	54.83 ± 3.16	52.44 ± 3.34	57.28 ± 4.18	61.61 ± 8.32	56.50± 3.54	57.50 ± 2.72
sonostufo	55.95 ± 2.56	52.27 ± 3.97	57.00 ± 4.61	57.11 ± 4.19	**59.44 ± 3.74**	54.05 ± 4.29	51.33 ± 3.56	59.28 ± 2.19	66.33 ± 7.25	58.88± 4.47	57.00 ± 4.55
chevuoi	60.00 ± 5.62	57.55 ± 4.16	57.72 ± 5.12	55.22 ± 3.16	54.77 ± 4.24	59.22 ± 4.22	57.00 ± 3.45	**61.11 ± 2.84**	65.83 ± 6.10	67.05± 2.12	61.11 ± 2.27
daccordo	61.53 ± 5.25	65.83 ± 3.34	**67.00 ± 3.87**	63.61 ± 2.34	65.50 ± 5.28	**67.00 ± 3.59**	63.27 ± 3.40	64.86 ± 3.94	67.33 ± 8.25	74.83± 3.54	65.66 ± 4.69
perfetto	66.61 ± 4.05	64.55 ± 4.54	62.05 ± 3.46	60.11 ± 4.60	64.16 ± 2.40	64.94 ± 4.79	65.83 ± 4.26	**67.72 ± 7.71**	68.11 ± 7.23	64.05± 3.78	66.16 ± 5.14
vattene	61.83 ± 7.02	65.66 ± 3.19	62.27 ± 2.16	61.83 ± 5.43	64.55 ± 4.07	**65.72 ± 1.88**	63.88 ± 3.24	65.11 ± 5.07	68.70 ± 4.21	67.44± 3.47	66.83 ± 2.74
basta	67.00 ± 6.56	65.11 ± 5.18	65.44 ± 3.35	63.38 ± 4.37	64.11 ± 2.55	65.27 ± 3.91	62.72 ± 5.42	**68.11 ± 4.28**	69.22 ± 6.32	74.94± 6.21	72.11 ± 4.87
buonissimo	56.56 ± 8.02	52.44 ± 12.1	**58.64 ± 6.57**	58.55 ± 5.18	55.94 ± 5.17	56.05 ± 5.82	54.50 ± 4.52	57.67 ± 5.71	61.50 ± 2.35	65.38± 6.76	55.11 ± 5.00
cheduepalle	63.89 ± 2.01	66.27 ± 2.29	66.44 ± 2.82	65.83 ± 2.87	67.33 ± 3.33	66.66 ± 1.81	64.94 ± 2.47	**67.72 ± 2.01**	68.11 ± 3.05	76.05± 2.67	70.72 ± 2.85
cosatifarei	58.78 ± 6.20	61.99 ± 3.29	62.33 ± 4.03	61.50 ± 4.17	61.61 ± 4.40	**64.50 ± 3.55**	61.50 ± 5.25	62.11 ± 4.98	67.17 ± 6.24	64.88± 4.40	63.94 ± 5.75
fame	61.11 ± 5.23	59.55 ± 2.98	60.66 ± 2.87	61.38 ± 3.34	62.66 ± 3.90	**63.38 ± 3.47**	58.33 ± 1.50	60.55 ± 2.35	66.61 ± 7.22	65.94± 3.52	61.44 ± 4.40
noncenepiu	53.83 ± 1.99	52.61 ± 4.39	53.11 ± 3.55	53.83 ± 2.70	52.94 ± 3.21	**54.94 ± 4.71**	51.33 ± 3.73	**54.94 ± 3.01**	58.55 ± 5.23	55.83± 5.57	56.44 ± 4.21
furbo	63.39 ± 5.06	67.27 ± 3.56	65.22 ± 3.65	63.00 ± 3.10	66.33 ± 1.53	68.66 ± 3.30	66.05 ± 2.93	**68.70 ± 4.65**	72.22 ± 4.31	73.05± 1.87	70.22 ± 4.71
combinato	56.67 ± 7.26	56.33 ± 2.41	59.83 ± 3.73	58.55 ± 4.55	61.05 ± 3.38	58.83 ± 2.61	55.83 ± 2.43	**62.11 ± 2.26**	65.83 ± 6.32	75.00± 2.83	64.05 ± 3.66
freganiente	55.00 ± 4.25	52.38 ± 3.21	58.77 ± 3.28	56.94 ± 4.56	54.05 ± 6.20	56.94 ± 3.74	53.00 ± 3.12	**59.28 ± 4.89**	64.16 ± 3.95	58.05± 4.77	54.44 ± 4.59
seipazzo	60.61 ± 6.52	57.16 ± 4.58	55.50 ± 4.89	55.00 ± 3.92	53.55 ± 3.90	60.05 ± 3.23	58.05 ± 4.37	**61.72 ± 5.02**	65.44 ± 6.03	70.77± 2.97	61.94 ± 5.92
tantotempo	59.89 ± 5.15	61.50 ± 2.95	60.75 ± 3.75	59.27 ± 3.63	63.66 ± 1.96	62.22 ± 2.35	61.27 ± 2.75	**63.80 ± 3.72**	67.27 ± 3.25	70.83± 3.22	65.33 ± 2.74
messidaccordo	54.83 ± 1.34	53.49 ± 8.88	57.15 ± 4.47	**59.05 ± 4.67**	**59.05 ± 2.98**	58.44 ± 2.39	55.66 ± 3.43	55.94 ± 3.17	59.05 ± 5.23	54.50± 4.76	58.50 ± 4.92
ok	53.06 ± 2.98	51.83 ± 2.95	**56.50 ± 10.2**	53.44 ± 3.62	52.50 ± 2.78	53.88 ± 3.39	50.22 ± 2.79	56.39 ± 2.19	60.75 ± 6.35	51.27± 3.43	52.77 ± 3.16

Table 3. **CGD2011 dataset.** Classification accuracies for one-vs-all binary classifications. The HOF features are used as main view, and histograms of joint positions are used as auxiliary view. Best accuracies are highlighted in boldface.

		LMIBPI
1	*Chimpanzee versus Giant panda*	88.32 ± 0.33
2	*Chimpanzee versus Leopard*	**94.05 ± 0.10**
3	*Chimpanzee versus Persian cat*	90.76 ± 0.19
4	*Chimpanzee versus Pig*	**87.32 ± 0.17**
5	*Chimpanzee versus Hippopotamus*	**90.21 ± 0.12**
6	*Chimpanzee versus Humpback whale*	97.76 ± 0.26
7	*Chimpanzee versus Raccoon*	**88.21 ± 0.27**
8	*Chimpanzee versus Rat*	85.31 ± 0.29
9	*Chimpanzee versus Seal*	**93.11 ± 0.23**
10	*Giant panda versus Leopard*	92.95 ± 0.20
11	*Giant panda versus Persian cat*	92.82 ± 0.32
12	*Giant panda versus Pig*	**86.71 ± 0.40**
13	*Giant panda versus Hippopotamus*	91.12 ± 0.29
14	*Giant panda versus Humpback whale*	**98.82 ± 0.14**
15	*Giant panda versus Raccoon*	89.21 ± 0.30
16	*Giant panda versus Rat*	**89.13 ± 0.25**
17	*Giant panda versus Seal*	**93.81 ± 0.19**
18	*Leopard versus Persian cat*	94.97 ± 0.22
19	*Leopard versus Pig*	87.31 ± 0.21
20	*Leopard versus Hippopotamus*	92.71 ± 0.16
21	*Leopard versus Humpback whale*	98.61 ± 0.26
22	*Leopard versus Raccoon*	80.12 ± 0.22
23	*Leopard versus Rat*	**90.13 ± 0.21**
24	*Leopard versus Seal*	**95.18 ± 0.33**
25	*Persian cat versus Pig*	82.27 ± 0.24
26	*Persian cat versus Hippopotamus*	92.38 ± 0.32
27	*Persian cat versus Humpback whale*	**97.42 ± 0.25**
28	*Persian cat versus Raccoon*	91.24 ± 0.18
29	*Persian cat versus Rat*	**70.49 ± 0.45**
30	*Persian cat versus Seal*	**88.41 ± 0.36**
31	*Pig versus Hippopotamus*	73.42 ± 0.12
32	*Pig versus Humpback whale*	95.93 ± 0.12
33	*Pig versus Raccoon*	**82.19 ± 0.15**
34	*Pig versus Rat*	**73.31 ± 0.25**
35	*Pig versus Seal*	**83.11 ± 0.43**
36	*Hippopotamus versus Humpback whale*	90.11 ± 0.28
37	*Hippopotamus versus Raccoon*	84.46 ± 0.36
38	*Hippopotamus versus Rat*	**86.11 ± 0.26**
39	*Hippopotamus versus Seal*	70.49 ± 0.41
40	*Humpback whale versus Raccoon*	96.97 ± 0.27
41	*Humpback whale versus Rat*	93.89 ± 0.19
42	*Humpback whale versus Seal*	**86.13 ± 0.17**
43	*Raccoon versus Rat*	79.63 ± 0.14
44	*Raccoon versus Seal*	**91.63 ± 0.36**
45	*Rat versus Seal*	79.21 ± 0.28
	Average	**88.38**

Table 4. **AwA dataset.** AP results for one-vs-one classification. Bold numbers represent the cases where LMIBPI improves performance upon SVM, SVM+, RankTr, and LIR [42].

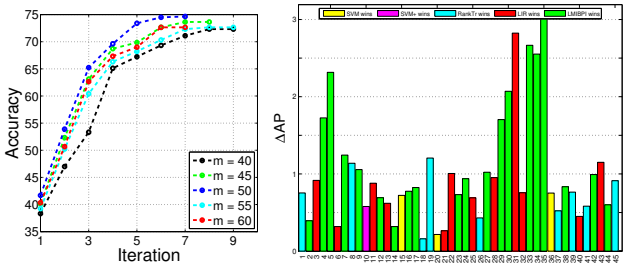

Figure 5. **Rate of convergence and AwA dataset.** Left: Plot showing the convergence rate for different m's for the drink class of the HMDB dataset. Right: Plot showing the differences between the AP of the winner LUPI method against the average accuracy over the following methods: SVM (yellow), SVM+ (magenta), RankTr (cyan), LIR (red), and LMIBPI (green).

We use 50 and 200 samples per class for training and testing, respectively. The train/test split is repeated 20 times. For fair comparison with [33, 42] we use the linear kernel. Due to the limited space Table 4 reports only the average precision (AP) results for our approach, where we have indicated in bold when LMIBPI has improved the AP, which happens 20 times out of 45, and 12 times the improvement is significant according to the z-test. The table including the results of the other approaches can be found in [42]. Figure 5 shows that SVM has the highest AP 3 times, SVM+ 1 time, RankTr 9 times, and LIR [42] 12 times.

8. Conclusions

In order to develop a general approach for improving visual recognition when auxiliary information is available at training time, we have taken an information theoretic approach, and have extended the IB principle to IBPI. In addition, we have expanded it further for learning any type of classifier based on risk minimization, where training samples with missing auxiliary view can be handled seamlessly. We have applied this new IBPI principle to derive LMIBPI, a large-margin classifier for which we provide an optimization procedure in the primal space (which takes about 10 iterations to converge). The experiments show that the IBPI principle can leverage several types of auxiliary information, like supplemental visual features, bounding box annotations, 3D skeleton tracking data, and animal attributes, and uses them for improving visual recognition, by learning a classifier that is better than the corresponding single-view version. The experiments also show that the proposed approach is more effective than just reducing a multi-view method to work with a missing view. Finally, the proposed LMIBPI outperforms all the state-of-the-art LUPI classifiers on the examined datasets.

Acknowledgements

We thank the Area Chair and the Reviewers for providing constructive feedback. This material is based upon work supported, in part, by the Center for Identification Technology Research and the National Science Foundation under Grant No. 1066197.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1–2):151–175, 2009.

[2] ChaLearn. ChaLearn gesture dataset (CGD2011). California, 2011.

[3] G. Chechik and N. Tishby. Extracting relevant structures with side information. In *NIPS*, 2002.

[4] J. Chen, X. Liu, and S. Lyu. Boosting with side information. In *ACCV*, pages 563–577, 2012.

[5] L. Chen, W. Li, and D. Xu. Recognizing RGB images by learning from RGB-D data. In *CVPR*, pages 1418–1425, 2014.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley and Sons, Inc., 1991.

[7] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *ICCV*, pages 1395–1402, 2011.

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009.

[9] J. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: SVM-2K, theory and practice. In *NIPS*, 2006.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.

[11] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.

[12] J. Feyereisl, S. Kwak, J. Son, and B. Han. Object localization based on structural SVM using privileged information. In *NIPS*, 2014.

[13] S. Fouad, P. Tino, S. Raychaudhury, and P. Schneider. Incorporating privileged information through metric learning. *IEEE Trans. on Neural Networks and Learning Systems*, 24(7):1086–1098, 2013.

[14] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

[15] D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, 141(1–2):349–382, 2013.

[16] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16:26392664, 2004.

[17] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[18] T. Joachims. Training linear svms in linear time. In *KDD*, 2006.

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *IEEE ICCV*, 2011.

[20] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 36(3):453–465, 2014.

[21] M. Lapin, M. Hein, and B. Schiele. Learning using privileged information: SVM+ and weighted SVM. *Neural Networks*, 53:95–108, 2014.

[22] F. Lauer and G. Bloch. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing*, 71(7–9):1578–1594, 2008.

[23] Q. Li, J. Wu, and Z. Tu. Harvesting mid-level visual concepts from large-scale internet images. In *CVPR*, pages 851–858, 2013.

[24] W. Li, L. Niu, and D. Xu. Exploiting privileged information from web data for image categorization. In *ECCV*, pages 437–452, 2014.

[25] L. Liang and V. Cherkassky. Connection between svm+ and multi-task learning. In *IJCNN*, pages 2048 – 2054, 2008.

[26] C. J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

[27] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[28] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.

[29] D. Pechyony and V. Vapnik. On the theory of learning with privileged information. In *NIPS*, 2010.

[30] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE TPAMI*, 29(10):1848–1852, 2007.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[32] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[33] V. Sharmanska, N. Quadrianto, and C. Lampert. Learning to rank using privileged information. In *IEEE ICCV*, pages 825–832, 2013.

[34] N. Slonim, N. Friedman, and N. Tishby. Multivariate information bottleneck. *Neural Computation*, 18(8):1739–1789, 2006.

[35] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS*, 1999.

[36] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

[37] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, pages 776–789, 2010.

[38] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009.

[39] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[40] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, pages 606–613, Sept 2009.

[41] Z. Wang, T. Gao, and Q. Ji. Learning with hidden information using a max-margin latent variable model. In *ICPR*, pages 1389–1394, 2014.

[42] Z. Wang and Q. Ji. Classifier learning with hidden information. In *CVPR*, pages 4969–4977, 2015.

[43] Z. Wang, X. Wang, and Q. Ji. Learning with hidden information. In *ICPR*, pages 238–243, 2014.

[44] L. Wolf and N. Levy. The SVM-Minus similarity score for video face recognition. In *CVPR*, pages 3523–3530, 2013.

[45] C. Xu, D. Tao, and C. Xu. Large-margin multi-view information bottleneck. *IEEE TPAMI*, 36(8):1559–1572, 2014.

[46] X. Xu, W. Li, and D. Xu. Distance metric learning using privileged information for face verification and person re-identification. *IEEE Trans. on Neural Networks and Learning Systems*, 2015.

[47] Z. Yang, H. Zhang, Z. Yuan, and E. Oja. Kullback-Leibler divergence for nonnegative matrix factorization. In *ICANN*, pages 250–257, 2011.

[48] Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *IEEE CVPR*, pages 1855–1862, 2010.

[49] Q. Zhang and G. Hua. Multi-view visual recognition of imperfect testing data. In *ACM MM*, pages 561–570, 2015.

[50] Q. Zhang, G. Hua, W. Liu, Z. Liu, and Z. Zhang. Can visual recognition benefit from auxiliary information in training? In *ACCV*, pages 65–80, 2014.

[51] T. Zhou, D. Tao, and X. Wu. NESVM: A fast gradient method for support vector machines. In *ICDM*, pages 679–688, 2010.