This ICCV2013 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Frustratingly Easy NBNN Domain Adaptation

Tatiana Tommasi
ESAT-PSI & iMinds
KU Leuven, Belgium
ttommasi@esat.kuleuven.be

Barbara Caputo
University of Rome La Sapienza, Italy
caputo@dis.uniroma1.it

Abstract

Over the last years, several authors have signaled that state of the art categorization methods fail to perform well when trained and tested on data from different databases. The general consensus in the literature is that this issue, known as domain adaptation and/or dataset bias, is due to a distribution mismatch between data collections. Methods addressing it go from max-margin classifiers to learning how to modify the features and obtain a more robust representation. The large majority of these works use BOW feature descriptors, and learning methods based on image-to-image distance functions.

Following the seminal work of [6], in this paper we challenge these two assumptions. We experimentally show that using the NBNN classifier over existing domain adaptation databases achieves always very strong performances. We build on this result, and present an NBNN-based domain adaptation algorithm that learns iteratively a class metric while inducing, for each sample, a large margin separation among classes. To the best of our knowledge, this is the first work casting the domain adaptation problem within the NBNN framework. Experiments show that our method achieves the state of the art, both in the unsupervised and semi-supervised settings.

1. Introduction

The amount of freely available, curated image databases has dramatically increased over the last years, thanks to the diffusion of high-quality cameras, and also to the introduction of new and cheap annotation tools such as Mechanical Turk. Attempts to leverage over and across such large data sources has proved challenging. Several authors showed that, for a given task, training on a dataset (e.g. Pascal VOC 07 [11]) and testing on another (e.g. ImageNet [9]) produces very poor results, although the set of depicted object categories is the same [22, 25, 12]. In other words, existing object categorization methods do not generalize well across databases.

This fact has been interpreted so far mainly in two different ways. The *first* has adopted the notion of *domain*, already used in machine learning for speech and language processing [5, 8]. A source domain (S) usually contains a large amount of labeled images, while a target domain (T) refers broadly to a dataset that is assumed to have different characteristics from the source, and few or no labeled samples. Within this context, the across dataset generalization problem stems from an intrinsic difference between the marginal distributions of the data $P_S(x) \neq P_T(x)$ under the covariate shift assumption $P_S(y|x) = P_T(y|x)$ where $x \in \mathcal{X}$ indicates the generic image sample and $y \in \mathcal{Y}$ the corresponding class label. The *second* way argues that specific annotator tendencies, as well as changes in the acquisition device, procedure and in the post-processing create a *dataset bias* [25] among corresponding classes, which leads to $P_S(x|y) \neq P_T(x|y)$. In both cases, the techniques proposed so far to rectify the distribution mismatch, regardless of its underlying reason, range from max-margin classifiers able to adapt their learning parameters [15] to methods attempting to learn how to project the data in a new intermediate space, where the features lose the specific bias [13].

But what if broadly adopted features and classifiers would be part of the problem, rather than good ingredients for its solution? Since the seminal work of Boiman *et al* [6], the Naive Bayes Nearest Neighbor (NBNN) method has challenged (i) the vector quantization step in Bag of Words (BOW, [23]) descriptors, that allows to have a compact feature representation to the expenses of its informative content, and (ii) the computation of image-to-image distances, that enables to use kernel based classification methods, but that does not generalize much beyond the labelled images. Even though the domain adaptation/dataset bias problem is clearly at its core a generalization problem, the almost totality of approaches presented so far use image-to-image learning algorithms on top of BOW representations.

Here we turn the table around: instead of considering a descriptor and trying to amend the issues that it generates with image-to-image distance based learning methods, we show that the NBNN method is *a priori* more robust to visual domain shift. Experiments on existing domain adaptation databases confirm our intuition: on all of them, the NBNN classifier obtains strong results, often achieving the state of the art. Armed with this knowledge, we build a NBNN domain adaptation algorithm that iteratively learns a class metric while inducing, for each sample, a large margin separation among classes. Our algorithm is the first NBNN-based domain adaptation method proposed so far and performs consistently better than the original NBNN classifier, obtaining the state of the art in all experiments .

The rest of the paper is organized as follows: after reviewing previous work (section 2) we set the notation and show with a proof of concept experiment that the mere plug-and-play of the NBNN classifier leads to remarkable results on the domain adaptation problem (section 3). Section 4 describes our NBNN-based domain adaptation algorithm, and section 5 reports the experiments showing the strength of the original NBNN classifier on two different settings, as well as the added value brought by our algorithm. We conclude with an overall discussion of our findings, and of possible new research directions stemming from our results.

2. Related Work

Domain adaptation is a widely researched problem. It deals with data sampled from different distributions and on how to compensate this mismatch. The topic has a long history in machine learning [3, 21] and natural language processing [8, 5]. It recently emerged also in the vision community [22, 13, 4]. The field is infact becoming increasingly aware of the dataset bias issue [25]: existing image collections used for object categorization present specific characteristics which prevent cross-dataset generalization. As a result, any supervised learning method trained on a particular dataset achieves a significant decrease in accuracy when tested on a different one.

Many of the adaptive methods recently introduced for object recognition focus on modifying the image descriptors. They define several procedures to transform the input features such that different domains become similar and any classifier can be proficiently applied. In [22] the key idea is to learn a regularized transformation using information-theoretic metric learning that maps source data to the target domain. Gopalan *et al.* [13] proposed to project both the source and target domain samples onto a set of intermediate subspaces, while Gong *et al.* [12] considered an infinite number of subspaces through a kernel-based method. Another stream of works propose classifier adaptation approaches. They are mainly based on max-margin methods associated with strategies to adapt the learning parameters to novel problems [10, 29, 7]. The most recent paper which follows this line is [15]. Here the authors learn a a model composed of a general and a specific part, taking care of the dataset bias at training time.

In spite of their variety, most of the cited techniques are evaluated by using BOW descriptors. This representation allows to reach state of the art results (even combined with SPM [16]) over in-domain problems, but its use for cross-domain tasks may not be beneficial. As pointed out in [20, 19], a visual word dictionary built on the source set may be a bad reference for local descriptor extracted from the target. This worsen the domain distribution mismatch. Thus, any adaptive method might end up solving this problem, instead of focusing on the real variation of the image content.

The weaknesses of BOW have been fully exposed in [6]. There, the authors did shed light on the issues of local descriptor quantization, and on the limits of classifiers based on image-to-image distance. The NBNN classifier was introduced to overcome both these problems and it has shown top performances when applied on visual object categorization tasks. This seminal work has been followed by several other publications which proposed to add a learning component to the non-parametric NBNN method [27, 2]. Our work fits in this context: we focus on the suitability of NBNN for domain adaptation and we propose an algorithm that further exploits the NBNN specific features.

3. NBNN and Domain Adaptation

In this section, after defining the setting and notation of the paper (section 3.1), we briefly review the NBNN method (section 3.2) and we describe a proof of concept experiment illustrating the benefits of NBNN for the domain adaptation problem (section 3.3).

3.1. Problem Setting and Notation

Let us consider an image i represented by a set of descriptors $F_i = \{f_{i1}, f_{i2}, \ldots, f_{iM_i}\}$ each extracted at one of the $m = \{1, \ldots, M_i\}$ detected interest points. Here every local feature is denoted as $f_{im} \in R^d$ and we neglect all the information regarding the point coordinates. In this framework a widely used procedure consists in reducing the descriptor set to a BOW representation. Each f_{im} is quantized to a pre-defined vocabulary of w visual words and substituted by the index of the closest codebook element. Thus the image i is described by a single vector $BOW_i \in R^w$ containing the normalized histogram of index frequencies. With this representation, the Euclidean distance among two images $\{i, j\}$ is simply $D_{EUC}(i, j) = \|BOW_i - BOW_j\|^2$ and it can be used directly in a 1-Nearest Neighbor (NN) classifier.

Given two images and the corresponding sets F_i, F_j, we can also measure their similarity by matching all their local features with the kernel function [18]

$$K(F_i, F_j) = \frac{1}{M_i}\frac{1}{M_j}\sum_{g=1}^{M_i}\sum_{q=1}^{M_j}(\tilde{K}_{ij}(g,q))^a, \qquad (1)$$

where we simply choose $\tilde{K}_{ij}(g,q) = f_{ig}^{\top} f_{jq}$. The parameter a automatically assigns more relative weight to the terms in the sum corresponding to similar local descriptors. Thus we can define a *match distance* among two images $\{i, j\}$ as $D_{MAT}(i,j) = K(F_i, F_j) + K(F_i, F_j) - 2K(F_i, F_j)$.

3.2. The NBNN algorithm

The local descriptors of any image i can be considered as independently sampled from a class-specific feature distribution. Hence, for a maximum a posteriori classifier, each descriptor m votes for the most probable class in $c = \{1, \ldots, C\}$ and the collection of votes is used to label the image. In this setting the NBNN algorithm [6] defines the votes in terms of the local distance $D_{LOC}(im, c) = \|f_{im} - f_{im}^{c}\|^2$ between each feature and its NN in class c . Finally the image-to-class distance is written as $D_{I2C}(F_i, c) = \sum_{m=1}^{M_i} D_{LOC}(im, c)$ and any image i is labeled according to

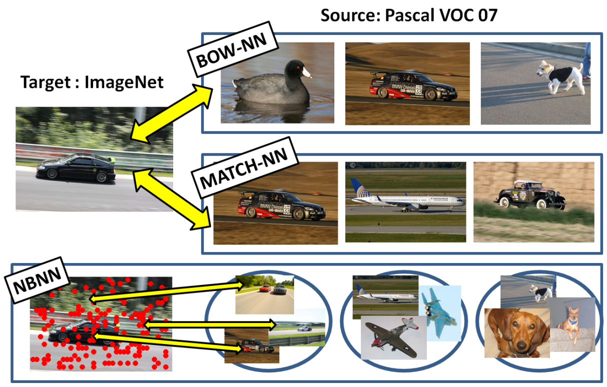
$$p = \underset{c}{\operatorname{argmin}} \; D_{I2C}(F_i, c) \; . \qquad (2)$$

We indicate all the remaining classes with $n : \{c \neq p\}$ and we name $D_{I2C}(F_i, p)$, $D_{I2C}(F_i, n)$ respectively as positive and negative distances.

3.3. Proof of Concept Experiment

A key feature of NBNN is its ability to generalize over categorization problems. This is due to the combined effect of (i) avoiding the BOW representation, where the quantization procedure might significantly degrade the the discriminative power the original local descriptors, and (ii) considering all descriptors as belonging to their class label, ignoring from which image they have been originally computed (the image-to-class paradigm). We believe that these two components may be highly effective for generalizing across domains and datasets.

Figure 1 illustrates this point when training on the whole Pascal VOC07 dataset (20 classes), and testing on an image of class car extracted from ImageNet. We used SIFT descriptors defining a BOW codebook over a random selection of the Pascal data. By using NN over the BOW representation the query image on the left is assigned to the class bird. The second nearest neighbor is actually a car image, while the third is a dog image. A NN classifier using the D_{MAT} distance (that we call from now on MATCH) labels the query image correctly. The same happens for the NBNN algorithm: car is the class with the minimal image-to-class distance, followed by airplane and dog. The table shows the accuracy result over 30 samples of class car. We repeated the experiment also for 30 samples of class chair and bird. In all cases NBNN performs better than BOW. MATCH can be better or worse than BOW, but it is always outperformed by NBNN.



	BOW	MATCH	NBNN
car	20.0	23.3	46.7
chair	13.3	13.3	33.3
bird	6.7	3.3	13.3

Figure 1. Analysis of the NBNN performance in comparison to NN using BOW and MATCH (1-Nearest Neighbor with match distance and $a = 10$). The table contains the recognition rate (%) results obtained separately over three classes, training on Pascal VOC07 and testing on ImageNet.

4. NBNN-based Domain Adaptation

Our main intuition is that the distribution of local features per class may be similar across two domains despite the variation between the respective image distributions. This similarity can also be enhanced by a domain adaptation approach in the NBNN setting. With this goal in mind, we start from the metric learning method proposed in [27] and we extend it to deal with two domains. Inspired by [7] we propose a greedy algorithm which progressively selects an increasing number of target instances and combines it with a subset of the source data while learning iteratively a Mahalanobis metric per class. We give a schematic representation of our Domain Adaptive NBNN (DA-NBNN) method in Figure 2 and we formalize it in the following.

Metric Learning. When facing an unsupervised cross-domain problem several labeled samples of a source set $S : \{F_l, y_l\}_{l=1}^{L}$ are available together with unlabeled samples of the target set $T : \{F_u\}_{u=1}^{U}$. By applying the NBNN algorithm, with the source local features as training[1], we can estimate the label for each target sample according to (2). The data subsets S_k and T_k extracted from the two domains can then be used to learn a specific Mahalanobis metric per class which induces for each sample a large margin separation between the image-to-class distance of the correct (or estimated) class (p) and all the other classes (n). The metrics are coded in the matrices $W^c \in R^{d \times d}$ for

[1] For NBNN (and NN) there is no real training phase, but we refer to the available labeled samples as training set, generalizing from the standard statistical learning terminology.

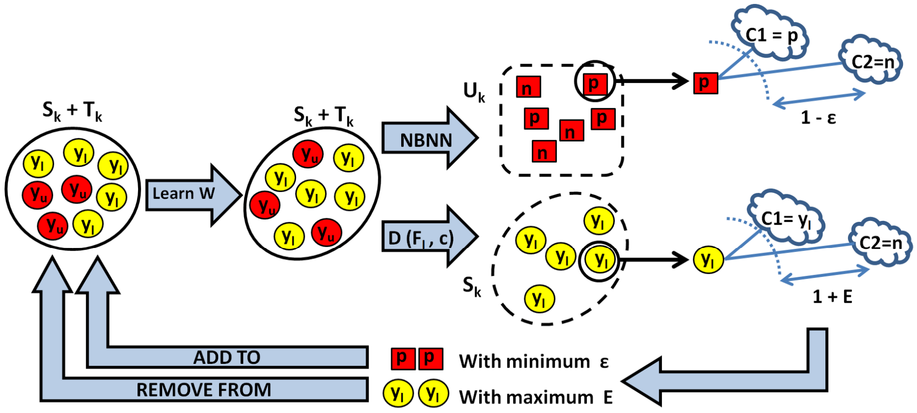


Figure 2. Schematic representation of our DA-NBNN algorithm in a simple binary case. The source samples are depicted as yellow circles and marked with their class label y_l, while the red circles are the target samples with their assigned label y_u. At each of the k iterations, NBNN predicts on the unlabeled target samples in U_k. Two samples from U_k and two samples from S_k are then respectively added to, and removed from the training set, depending on the difference between the negative and positive distances.

$c = \{1, \ldots, C\}$ and the image-to-class distance becomes

$$D(F_i, c) = \sum_{m=1}^{M_i} (f_{im} - f^c_{im})^\top W^c (f_{im} - f^c_{im}) \;, \qquad (3)$$

where we dropped the subscript $I2C$ to simplify the notation.

We would like to have $D(F_i, n) - D(F_i, p) > 1$ for each $\{i, p, n\}$. Hence, we define an optimization problem for all the W^c which imposes this constraint, and we regularize it by minimizing over the positive distances with a trade-off parameter λ. We follow the formulation in [27], but we keep the source and target samples separated with different parameters λ_s,λ_t and relative weights $\Gamma_s(k)$,$\Gamma_t(k)$ such that the full objective reads like this

$$\begin{aligned} &O(W^1, W^2, \ldots, W^C)_k = \\ &\Gamma_s(k) \left\{ (1 - \lambda_s) \sum_{l,p\to l} D(F_l, p) + \lambda_s \sum_{l,p\to l,n\to l} \xi_{lpn} \right\}_{S_k} + \\ &\Gamma_t(k) \left\{ (1 - \lambda_t) \sum_{u,p\to u} D(F_u, p) + \lambda_t \sum_{u,p\to u,n\to u} \xi_{upn} \right\}_{T_k} , \end{aligned} \qquad (4)$$

and the optimization problem is

$$\begin{aligned} &\min_{\{W^1, W^2, \ldots, W^C\}_k} O(W^1, W^2, \ldots, W^C)_k \\ &\text{such that } \;\forall\; \{l, p, n\}_{S_k} : \\ &D(F_l, n) - D(F_l, p) > 1 - \xi_{lpn} \;,\;\; \xi_{lpn} > 0 \\ &\text{and } \;\forall\; \{u, p, n\}_{T_k} : \\ &D(F_u, n) - D(F_u, p) > 1 - \xi_{upn} \;,\;\; \xi_{upn} > 0 \\ &\text{with } \;W^c_k \succeq 0 \;\;\forall\;\; c \in \{1, \ldots, C\} \end{aligned} \qquad (5)$$

where the slack variables ξ in the error terms allow soft-margins. The full problem can still be easily solved with the gradient descent method presented in [27].

The process is repeated several times over progressively refined selections of the data, with the index k referring to the subsequent iterations. At the first round the labels predicted for the target samples may not be fully reliable. The penalty assigned to large positive distances $D(F_u, p)$ and to active triplets (*i.e.* the subset of triplets $\{u, p, n\}$ for which $\xi_{upn} > 0$) should be small at the beginning and increase in the following steps. On the other hand, we want the source sample importance to decrease in time. For this reason we adopted a weighting strategy based on a temporal criterion with $\Gamma_s(k) = 1 - \Gamma_t(k)$ and $\Gamma_t(k) \propto k$.

Training sample selection. We clarify here how the samples are extracted from the two domains to define the sets S_k and T_k . For completeness we also introduce a third set $U_k = T - T_k$. At each iteration a prediction is performed over U_k by using NBNN with the image-to-class distance in (3), the matrices W^c_k, and the combination $(S_k + T_k)$ as training set. The distances $D(F_l, p)$ and $D(F_l, n)$ are calculated at each round by using $f^c_{lm} \in (S_k + T_k - l)$.

We initialize the method with $S_0 = S$, $T_0 = \emptyset$, $U_0 = T$, $W^c_0 = I \;\; \forall \;c$. For each assigned class $y_u = p$ we sort the samples in U_k by calculating $z_{p\to u} = \min_n [1 - D(F_u, n) + D(F_u, p)]_+$ and we add to T_{k+1} the instances u_1,u_2 with $z_{p\to u_1} < z_{p\to u_2} < z_{p\to u}, \forall \; u \in U_k$. Here $[x]_+ = max\{x, 0\}$. In the considered max-margin framework, this corresponds to identifying a couple of images that fall into the margin band and which are closest to the margin bound. This procedure is repeated at each round k and the described samples are progressively moved from U_k to T_{k+1}, thus helping to tune the metric at the following iteration and adapting it to solve the target problem.

While focusing on the target, the source information should become less and less relevant. At each iteration we evaluate $z_{p\to l} = \min_n [D(F_l, n) - D(F_l, p) - 1]_+$ and we choose two instances per class l_1,l_2 with $z_{p\to l_1} >$

Algorithm 1 DA-NBNN

```
Input  $S$ , $T$
Initialize $S_0 = S$ , $T_0 = \emptyset$ , $U_0 = T$ , $W_0^c = I$
for $k = 0$ to $K$ do
    Solve $\forall\, u \in U_k$
            $y_u = \mathrm{argmin}_c \; D(F_u, c)$
            with $f^c_{um} \in (S_k + T_k) \; \forall\, m \in \{1, \ldots, M_u\}$
    Calculate $D(F_l, c) \; \forall\, l \in S_k$
            with $f^c_{lm} \in (S_k + T_k - l) \; \forall\, m \in \{1, \ldots, M_l\}$
    Define $S_{k+1}, T_{k+1}, U_{k+1}$
    Solve the optimization problem in (5): get $W^c_{k+1}$
end for
Output  $y_u \;\; \forall\, u \in T$
```

$z_{p \to l_2} > z_{p \to l}, \forall \;\; l \; \in S_k$. They are iteratively removed from the training set and do not appear in S_{k+1}. In practice we identify and delete the source sample lying far from the margin bound and which have low probability to affect its position.

Discussion. The combination of metric learning and sample selection define our DA-NBNN algorithm. Each process helps the other: by tuning W^c we adjust the image-to-class distance, while by changing the training data we redefine the local feature set for each class, making it progressively more suitable for the target domain. The whole workflow of DA-NBNN[2] is summarized in Algorithm 1.

The computational load of our method depends mainly on the number of active triplets $\{lpn\}$,$\{upn\}$. At the beginning the first set may be large for large-scale source datasets, while in the following steps the dimension of both sets is regulated by the rate with which the samples are added/removed from the training set. The tricks adopted in [28, 27] to speed up the triplets identification can be directly applied on DA-NBNN reducing its complexity. Moreover, since the full method is iterative, it is also possible to use an early stopping strategy on the metrics update at each round k. Finally, from a theoretical point of view, DA-NBNN reaches convergence when $S_k = \emptyset$ and $\xi_{upn} < 0$ for each u. Nevertheless empirical evidence show that the algorithm produces significant improvements over standard NBNN (and I2CDML [27]) already after few ($k < 10$) iterations.

5. Experiments

We focus on the problem of object categorization, presenting results on the *Office* dataset [22]. This is the standard testbed used for visual domain adaptation methods. It provides three distinct domains: Amazon (images downloaded from online merchants), Webcam (low resolution images by a web camera) and Dslr (high-resolution images by a SLR camera). In the first set the images contain a single centered object usually on white background, while for the others the images are collected in real settings with lighting variation and background changes. Each domain consists of 31 classes of office-related objects (*e.g.* keyboard, monitor). We also consider the dataset *Office+Caltech*, i.e. a modified version of the described collection proposed in [12]. Here, the number of classes is reduced to 10 and a new domain is added with images extracted from Caltech-256 [14].

We adopted an experimental protocol similar to what used in previous work [22, 12]. SURF [1] features were extracted at interest point locations over all images resized at the same width and converted to grayscale. The obtained 64-dimensional descriptors are then used with a 1-Nearest Neighbor classifier in two different ways, without including any spatial information:

BOW: for each domain we constructed a 800-visual-word vocabulary by k-means on a random subset of the data. All the images were then represented by histograms over the codebooks. We choose the reference domain to use, and the associated histogram descriptor, depending on the specific experiment (more details in the following sections).

MATCH: avoiding any vector quantization, the SURF descriptors are kept in their original format and used to represent separately each patch surrounding the detected interest points. The D_{MAT} distance is used here with $a = 10$.

We ran experiments always considering a couple of domains, one regarded as source and the other as target. In the unsupervised setting the domains are disjoint, while in the semi-supervised setting three images per class of the target domain are added to the source. A preliminary analysis showed that DA-NBNN does not depend sensitively on the specific choices of its parameters; for our experiments we used $\lambda_s = \lambda_t = 0.5$ and $\Gamma_t(k) = 0.1k$ and we report the results for $k = 5$.

We consider as baseline several state-of-the-art domain adaptation methods:

PCA$_T$: all the original features are projected to the PCA subspace learned from the target domain [12].

SGF [13]: the Sampling Geodesic Flow approach considers a set of intermediate subspaces between the source and target domains to model their shift.

GFK [12]: it extends the previous technique by considering an infinite number of intermediate subspaces integrated by the Geodesic Flow Kernel.

Metric [22]: this approach is limited to the semi-supervised setting. It uses the correspondence between source and target label data to learn a metric which maps the samples into a new feature space.

[2] The algorithm was implemented in MATLAB and a demo is available online at `http://homes.esat.kuleuven.be/~ttommasi/source_code_ICCV13.html`.

Method	A → A	C → A	W → A	C → C	A → C	W → C	W → W	A → W	C → W
Unsupervised Setting									
BOW	41.1 ± 2.2	20.9 ± 3.0	16.1 ± 1.5	25.7 ± 1.6	20.3 ± 2.2	16.8 ± 1.1	65.1 ± 3.0	21.0 ± 3.6	18.4 ± 3.8
MATCH	43.5 ± 1.6	26.5 ± 2.3	23.8 ± 1.6	29.8 ± 1.5	24.2 ± 1.6	21.6 ± 1.0	67.3 ± 2.4	22.9 ± 3.7	18.2 ± 3.9
NBNN	64.6 ± 1.4	**41.0 ± 3.0**	**37.4 ± 1.2**	39.3 ± 2.7	31.3 ± 1.3	26.8 ± 1.0	85.9 ± 3.0	31.8 ± 2.2	28.4 ± 3.7
GFK	-	37.3 ± 2.5	31.7 ± 2.6	-	**34.2 ± 1.5**	**27.1 ± 1.2**	-	**37.0 ± 5.1**	**32.5 ± 6.8**
Semi-supervised Setting									
BOW	42.3 ± 2.9	24.1 ± 2.5	24.6 ± 1.8	26.5 ± 1.4	24.0 ± 2.5	22.6 ± 2.8	67.9 ± 4.0	26.0 ± 2.1	24.6 ± 5.2
MATCH	44.5 ± 1.8	24.8 ± 2.7	20.0 ± 1.9	30.4 ± 1.8	21.6 ± 1.5	18.9 ± 1.6	69.9 ± 2.5	18.5 ± 3.2	14.2 ± 3.4
NBNN	66.3 ± 1.9	**50.2 ± 2.3**	**43.5 ± 1.8**	41.0 ± 2.1	34.0 ± 2.2	31.6 ± 1.6	88.5 ± 2.4	**56.9 ± 4.3**	**57.7 ± 6.2**
GFK	-	43.5 ± 2.9	42.2 ± 3.0	-	**35.9 ± 2.3**	**32.1 ± 2.2**	-	56.8 ± 8.1	54.6 ± 6.6

Table 1. Recognition rate (%) obtained when fixing the target and changing the source task over three domains of the Office+Caltech dataset (A: Amazon, C: Caltech, W: Webcam). The results are average values over 20 iterations with randomly extracted samples. We report in bold the best results in each cross-domain column. The underlined values are used to evaluate a measure of domain shift (see detailed description in the text).

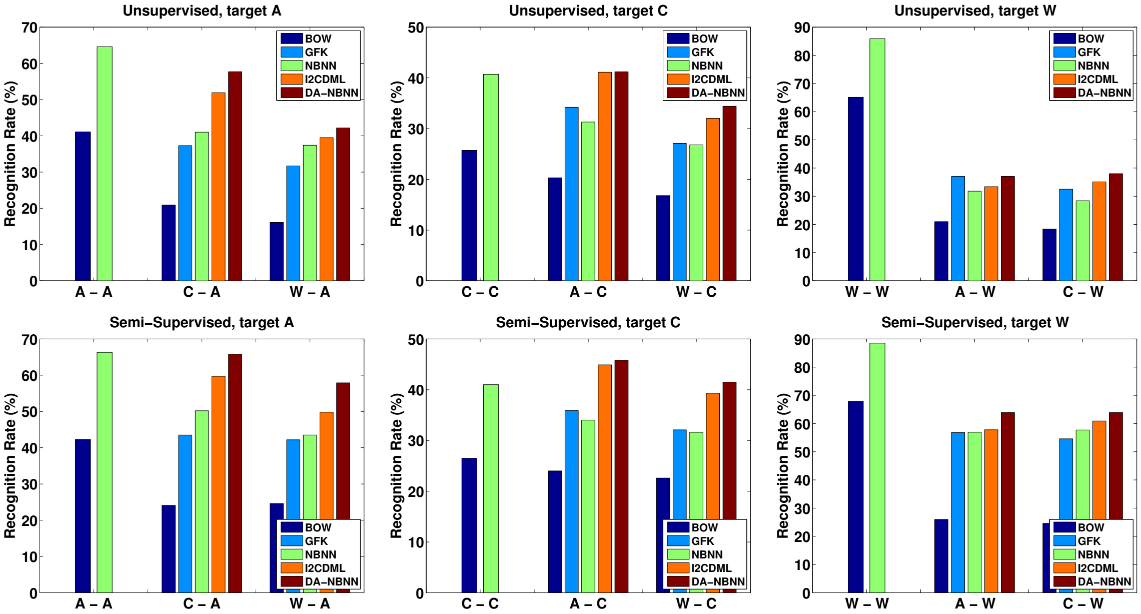


Figure 3. Comparison of our DA-NBNN against NBNN and I2CDLM over the Office+Caltech dataset (A: Amazon, C: Caltech, W: Webcam). The results of BOW and GFK in Table 1 are reported here for completeness.

We also benchmark DA-NBNN against the **I2CDML** method [27]. It runs large-margin metric learning over the source domain in the image-to-class setting and corresponds to a non-adaptive reference method. Each experiment ran over 20 random trials. We report the average accuracies as well as the respective standard deviations.

5.1. Results: NBNN

We first analyze the performance of NBNN on in-domain and cross-domain problems over the Office+Caltech dataset. We divided the data of each domain into a training and a test set. When Amazon and Caltech are used as target, all the training sets contain 20 images per class, while the remaining samples define the test set. When Webcam is used as target, all the training sets contain instead 15 samples. The Dslr domain is quite similar to Webcam and contains less images, hence we neglected it for these experiments.

We chose the BOW representation depending on the domain considered as target. For instance, in the $A \rightarrow C$ run the training samples of Amazon are used as source and the testing samples of Caltech are used as target, with all images represented by histograms over the Caltech visual vocabulary. With this setting, the performance of any classifier over the in-domain problem (*e.g.* $C \rightarrow C$) defines the reference upper limit to the accuracy achievable in the cross-domain setting.

Table 1 reports the results over all the possible do-

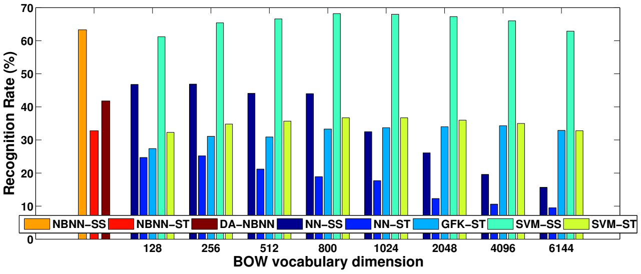


Figure 4. Average Source→Source (SS) and Source→Target (ST) results over the unsupervised setting in Table 1. We show the recognition performance obtained by changing the BOW vocabulary dimension and the classifier. For SVM we used the histogram intersection kernel and C=100.

Method	A → W	D → W	W → D
	Unsupervised Setting		
BOW [12]	10.7 ± 0.4	29.5 ± 0.3	32.7 ± 0.4
BOW	10.8 ± 1.3	27.6 ± 1.6	29.0 ± 1.4
PCA$_T$ [12]	13.8 ± 0.4	46.9 ± 0.4	47.2 ± 0.6
GFK [12]	15.0 ± 0.4	44.6 ± 0.3	49.7 ± 0.5
NBNN	20.0 ± 1.5	61.9 ± 1.4	60.8 ± 1.8
I2CDML	18.1 ± 1.5	59.7 ± 2.1	61.1 ± 1.4
DA-NBNN	23.3 ± 2.7	67.2 ± 1.9	67.4 ± 3.0
	Semi-supervised Setting		
BOW [12]	34.9 ± 0.6	38.4 ± 0.4	48.9 ± 0.5
PCA$_T$ [12]	44.4 ± 0.6	62.9 ± 0.5	63.4 ± 0.4
Metric [22, 12]	34.5 ± 0.7	36.9 ± 0.8	48.1 ± 0.6
SGF [13, 12]	45.1 ± 0.6	61.4 ± 0.4	63.4 ± 0.5
GFK [12]	46.4 ± 0.5	61.3 ± 0.4	66.3 ± 0.4
NBNN	40.0 ± 2.0	70.7 ± 1.2	67.2 ± 2.5
I2CDML	47.9 ± 1.3	73.8 ± 1.6	72.8 ± 2.1
DA-NBNN	52.8 ± 3.7	76.6 ± 1.7	76.2 ± 2.5

Table 2. Recognition rate (%) on the domains of the Office dataset (A: Amazon, D: Dslr, W: Webcam) . The results are average values over 20 iterations with randomly extracted samples. The rows with the citation in the method column contain results reported from the additional material of [12]. We implemented all the other methods.

main couples. We see that for all experiments, both in the unsupervised and semi-supervised settings, the accuracy increases from BOW to MATCH, to NBNN. This confirms the behavior already noticed in the preliminary Pascal/ImageNet experiments. The performance of NBNN can be appreciated even more by comparing the in-domain and cross-domain results. Let us consider for instance the underlined values in the top left part of Table 1. The percentual accuracy drop between $C \rightarrow A$ and $A \rightarrow A$, defined as $1-(C \rightarrow A)/(A \rightarrow A)$, can be seen as a measure of domain shift. We see that it goes from 49% among the BOW results to 39% among the MATCH results, while it is 36% among the NBNN results. This indicates that in the image-to-class setting the domain shift is intrinsically smaller. This behavior replicates over all of the other domain couples.

Finally, let us compare NBNN with the GFK domain adaptation approach[3]. Table 1 shows that NBNN is equal or better (confirmed by the signtest with $p < 0.01$) than GFK over ten of the twelve possible domain couples.

5.2. Results: DA-NBNN

We used the same setup described in the previous section to test our DA-NBNN algorithm, comparing it against both NBNN and I2CDML. To take into account the baseline results already discussed, we present the recognition accuracy as histograms in Figure 3.

First of all, it can be noticed that I2CDML outperforms NBNN (they are equivalent only for the $A \rightarrow W$ couple). This indicates that learning a proper image-to-class metric on the source domain, without even considering any target information provides extremely good results – indeed better than the state of the art established by GFK. Moreover, DA-NBNN improves over I2CDML with a significant gain ($p < 0.01$) in recognition rate over most of the domain couples (they perform equally only over $A \rightarrow C$ and $C \rightarrow W$).

It is worth paying attention also to the comparison between the DA-NBNN results and the corresponding in-domain NBNN results. In all cases, they appear at most statistically equivalent, with DA-NBNN reaching its in-domain upper limit. Differently, GFK may perform much better than the corresponding in-domain BOW (see for example Figure 3, top line, central column). In previous work this kind of behavior has been interpreted as further evidence of the effectiveness of the proposed adaptation method. We suggest another possible interpretation for such behavior. From our point of view, this might be seen as further evidence of the problems induced by the BOW representation, making the in-domain results a bad reference. This would imply that the adaptive methods, apart from the domain shift, ends up solving some of the issues generated by the vector quantization.

The advantage of DA-NBNN remains significant over other Source → Target results regardless of the chosen vocabulary dimension for BOW and the considered image-to-image-based classifier (see Figure 4).

5.3. Results: increasing the number of classes

Lastly, we repeated the experiments on the the original Office dataset, which contains more classes than its Office+Caltech version. We focused on the domain couples analyzed in previous work and we reproduce exactly the experimental setting of [22, 12]. This allows us to compare our results directly with the values reported in [12]. To make sure that this comparison is fair, we show the results

[3] We used the code released by the authors, available at http://www-scf.usc.edu/~boqinggo/domainadaptation.html

obtained in the unsupervised setting by implementing our own BOW method.

The recognition accuracies in both the unsupervised and semi-supervised setting are presented in Table 2. We see that again DA-NBNN outperforms all previously proposed methods, as well as the NBNN and I2CDML baselines. These results further confirm the power of our method.

6. Conclusion

In this paper we tested the generalization ability of the NBNN classifier [6] on the domain adaptation problem, looking into how the vector quantization step in BOW features, and the choice of an image-to-image based classifier, affect performance. The results obtained are competitive, often superior, to the state of the art, achieved by sophisticated image-to-image distance based learning methods. Building on this, we proposed an NBNN-based domain adaptation algorithm that learns iteratively a Mahalanobis class specific metric, while inducing for each sample a large margin separation among classes. We tested our algorithm on two different settings, in the unsupervised and semi-supervised scenarios, obtaining the state of the art.

We believe that these results provide very strong evidence of the importance of casting the domain adaptation problem within the NBNN framework. Our algorithm injects a learning component in it through metric learning, but several other options are possible, such as through the development of NBNN kernel functions [26]. Also, we believe that the results reported in this paper should be considered in the broader view of how to effectively leverage over prior knowledge. Therefore, its implications should be explored also on knowledge transfer [17] and multi-task unaligned learning [24] problems.

Acknowledgements. T.T. acknowledges the support of the EC FP7 project AXES.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *CVIU*, 110:346–359, 2008.

[2] R. Behmo, P. Marcombes, A. Dalalyan, and V. Prinet. Towards optimal Naive Bayes Nearest Neighbor. In *ECCV*, 2010.

[3] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.

[4] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010.

[5] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *EMNLP*, 2006.

[6] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[7] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE PAMI*, 32(5):770–787, 2010.

[8] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[10] L. Duan, I. W.-H. Tsang, D. Xu, and S. J. Maybank. Domain transfer svm for video concept detection. In *CVPR*, 2009.

[11] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2), 2010.

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[13] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[14] G. Griffin, A. Holub, and P. Perona. Caltech 256 object category dataset. Technical Report UCB/CSD-04-1366, California Institue of Technology, 2007.

[15] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

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

[17] J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, 2011.

[18] S. Lyu. Mercer kernels for object recognition with local features. In *CVPR*, 2005.

[19] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *CVPR*, 2013.

[20] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *ECCV*, 2012.

[21] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

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

[23] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[24] T. Tommasi, N. Quadrianto, B. Caputo, and C. Lampert. Beyond dataset bias: Multi-task unaligned shared knowledge transfer. In *ACCV*, 2012.

[25] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[26] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN kernel. In *ICCV*, 2011.

[27] Z. Wang, Y. Hu, and L.-T. Chia. Image-to-class distance metric learning for image classification. In *ECCV*, 2010.

[28] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.

[29] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.