Supplementary Material: A Deeper Look at Dataset Bias

Tatiana Tommasi[1], Novi Patricia[2], Barbara Caputo[3], Tinne Tuytelaars[4]

[1]Department of Computer Science, University of North Carolina, Chapel Hill
[2]Idiap Research Institute, Martigny; EPFL, Lausanne, Switzerland
[3]University of Rome, La Sapienza
[4]KU Leuven, ESAT-PSI, iMinds, Belgium

Throughout the main text, we omitted some details on the used methods and on the experimental setups, both to have a better readability and to observe the limit on the number of pages. We provide here those extra information.

1 Undoing the Dataset Bias

We summarize below the method proposed in [9] (section 3 of the main submission, last paragraph).

Let us assume to have $i = 1, \dots, n$ datasets $D_1, \dots, D_n$ each consisting of s_i training samples $D_i = \{(\mathbf{x}_1^i, y_1^i), \dots, (\mathbf{x}_{s_i}^i, y_{s_i}^i)\}$. Here $\mathbf{x}_j^i \in \mathbb{R}^m$ represents the m-dimensional feature vector and $y_j^i \in \{-1, 1\}$, the j-th example of D_i. The *Unbias* algorithm [9] consists in learning a binary model per dataset $\mathbf{w}_i = \mathbf{w}_{vw} + \Delta_i$, where $\mathbf{w}_{vw}$ is a model for the visual world, while Δ_i is the bias for each dataset. These two parts are obtained by solving the following optimization problem:

$$\min_{\substack{\mathbf{w}_{vw}, \Delta_i \\ \xi, \rho}} \frac{1}{2}\|\mathbf{w}_{vw}\|^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\|\Delta_i\|^2 + C_1 \sum_{i=1}^{n}\sum_{j=1}^{s_i}\xi_j^i + C_2 \sum_{i=1}^{n}\sum_{j=1}^{s_i}\rho_j^i \tag{1}$$

$$\text{subject to } \mathbf{w}_i = \mathbf{w}_{vw} + \Delta_i \tag{2}$$

$$y_j^i \mathbf{w}_{vw} \cdot \mathbf{x_j^i} \geq 1 - \xi_j^i \tag{3}$$

$$\xi_j^i \geq 0,\ \rho_j^i \geq 0 \tag{4}$$

$$i = 1, \dots, n\ \ j = 1, \dots, s_i\ . \tag{5}$$

We ran a preliminary cross-validation to choose the parameters. The cross validation is executed on the remaining source collections with the following parameter ranges: $\lambda = [0.5, 1, 5, 10]$, $C_1 = [10^2, 10^3, 10^4]$, $C_2 = [10, 20, 40, 60, 80, 100]$. The best parameter combination is chosen as the one for which $\mathbf{w}_{vw}$ produces the best result on the source collections and it is then used to obtain the results on the target dataset.

We showed results on four object classes, *car*, *dog*, *chair* and *cow*. To explain the setup we focus here on the first class and the experiment involving three datasets: SUN, Caltech101 and Pascal VOC07. For each of the collections extracted from the sparse setup we identified and separated the positive images belonging to class *car* and the negative images belonging to all the other classes. The setup is analogous for all the other experiments on different classes and collections.

(a) (b)

Fig. 1: (a) Imagenet images annotated as Caltech256 data with BOWsift but correctly recognized with DeCAF7. (b) Caltech256 images annotated as Imagenet by BOWsift but correctly recognized with DeCAF7.

2 Name the Dataset on the Dense Set

We can get a more concrete idea of the DeCAF performance in the name the dataset experiment executed on the dense setup (see section 4 of the main submission, first paragraph) by looking at Figure 1.

3 Noisy Source Data and Domain Adaptation

We briefly review here the methods used for the Bing-Caltech and Bing-Sun experiments (section 4 of the main submission, last paragraph).

Landmark. [5] Let $D_s = \{(\mathbf{x}_m, y_m)\}_{m=1}^M$ denote data points and their labels from the source domain and likewise $D_t = \{\mathbf{x}_n\}_{n=1}^N$ for the target domain. An indicator variable $\alpha_m \in \{0, 1\}$ is assigned to each source sample and identified by minimizing the Maximum Mean Discrepancy [6]:

$$\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(\mathbf{x}_m) - \frac{1}{N} \sum_n \phi(\mathbf{x}_n) \right\|^2 \tag{6}$$

$$\text{s.t.} \quad \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m y_{mc} = \frac{1}{M} \sum_m y_{mc} \tag{7}$$

where $\phi(\mathbf{x})$ is a kernel mapping function. Here the constraint is added to have the same class statistics in the selected landmarks as in the original data. The optimization is solved introducing the variables $\beta_m = \frac{\alpha_m}{\sum_m \alpha_m}$ and relaxing them to live in the simplex $\{0 \leq \beta_m \leq 1, \sum_m \beta_m = 1\}$. The binary solution for α_m is recovered by thresholding β_m. The geodesic flow kernel (GFK) [4], computed between the source D_s and the target D_t, is used to compose the kernel mapping function $\phi(\mathbf{x})$:

$$\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

$$= \exp\{-(\mathbf{x}_i - \mathbf{x}_j)^\top G(\mathbf{x}_i - \mathbf{x}_j)/\sigma^2\} \tag{9}$$

For different values of the bandwidth σ_q the corresponding landmarks $\mathcal{L}^q$ are identified and a new domain pair is obtained with source $D_s \setminus \mathcal{L}^q$ and target $D_t \cup \mathcal{L}^q$.

This method has several parameters. First of all GFK needs the definition of a subspace dimensionality. For our experiments we adopted the Subspace Disagreement (SD) measure [4] to choose this dimensionality. Using $\sigma_q = 2^q \sigma_0$ with σ_0 equal to the median distance over all the pairwise data points and $q = [-2, -1, 0, 1, 2]$. The threshold to choose the source samples is fixed as the median of all the β_m values, and the C parameter is chosen in the range $[10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4]$, exploiting the original model selection that automatically considers $D_s \setminus \sum_q \mathcal{L}^q$ as a validation set.

Subspace Alignment (SA). This approach learns an alignment matrix A between the source and the target subspace by minimizing the following Bregman divergence [3]:

$$\|X_s A - X_t\|_F^2 \,. \tag{10}$$

Here X_s and X_t are the subspace bases obtained by applying PCA over the source and target data. Specifically we set d equal to the SD measure [4] and we run the experiments with *SA* using the code provided by the authors. The C parameter is automatically chosen in the range $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3]$.

Domain Adaptation Machine (DAM). The DAM method [2] was originally developed for the transfer learning setup where at least few target training labeled samples are available, but it can also be applied in the unsupervised setting [11]. Specifically, DAM is formulated as a regression task solving the following optimization problem:

$$\min_{\mathbf{f^t},\mathbf{w},b,\xi_i,\xi_i^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_t}(\xi_i + \xi_i^*) + \frac{1}{2}\theta\left(\|\mathbf{f}_l^t - \mathbf{y}_l\|^2 + \sum_s \gamma_s \|\mathbf{f}_u^t - \mathbf{f}_u^s\|^2\right) \tag{11}$$

$$\text{s.t. } \mathbf{w}\cdot\phi(x_i) + b - f_i^t \leq \epsilon + \xi_i, \;\; \xi_i \geq 0, \tag{12}$$

$$f_i^t - \mathbf{w}\cdot\phi(x_i) + b \leq \epsilon + \xi_i^*, \;\; \xi_i^* \geq 0 \tag{13}$$

where ξ_i, ξ_i^* are slack variables for the ϵ-insensitive loss. With $\mathbf{f}_l^t$ we indicate the predictions of the learned model over the labeled target samples and $\mathbf{y}_l$ are the corresponding ground truth labels. $\mathbf{f}_u^t$ and $\mathbf{f}_u^s$ are respectively the predictions of the target model and of the source model s on the unlabeled part of the target. In the unsupervised case the first term in the parentheses of equation (11) is absent.

We used the implementation provided by the authors fixing the parameter for a single source as in the original code: $\theta = 1$ and $\gamma_1 = 0.5$. The source models are trained with a linear SVM choosing the C parameter by two-fold cross validation on the sources in the range $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3]$. For DAM we adopted the C value that was producing the best average result on the sources.

Discovering Latent Domains (reshape). Each dataset may contain multiple domains. To discover them, the method proposed in [8] considers two main steps. If there are K semantic categories and S domains the data points are grouped into $J = K \times S$ local clusters. Intuitively each local cluster will contain the data points from one domain and one semantic category. In the second stage the means of the local clusters are grouped

The optimization problem is formulated as an EM iteration algorithm with the two stages repeated until convergence.

This method have already been used to separate the Bing images in sub-domains [8]. In our work we applied it on the $K = 40$ categories and searched for $S = 2$ domains. We considered the combination with the *SA* and *DAM* methods.

reshape + SA: each of the two obtained domain is used separately as source set and the SA method is applied to obtain the alignment with the target set. This setup is chosen to analyze the ideal condition of an oracle providing the target labels. Our goal is to evaluate if, in this best-case scenario, it would be possible see an improvement over the case where all the source samples are considered at once.

reshape + DAM: we followed the same logic even when combining the discovered sub-domains with DAM. In this case the sources are not used separately but together trying all the possible combinations with $\gamma_1 = [0.1 : 0.1 : 1]$ and $\gamma_2 = 1 - \gamma_1$.

Self-Labeling. A simple strategy for domain adaptation consists in subselecting and using the target samples while learning the source model. This technique is known as *self-labeling* [1,7] and starts by annotating the target with a classifier trained on the source. The target samples for which the source model presents the highest confidence are then used together with the source samples in the following iteration. In our experiments we followed the same naïve multiclass self-labeling approach described in [10]. A one-vs-all SVM model is trained on the source data with the C parameter chosen by cross validation. At each iteration the model is used to classify on the target data and the images assigned to every class are ranked on the basis of their output margin. Only the images with a margin higher than the average are selected and sorted by the difference between the first and the second higher margin over the classes. The top samples in the obtained list per class are then used in training with the pseudo-labels assigned to them in the previous iteration. We set the number of iterations to 10 and the number of selected target samples per class to 2.

References

1. Bruzzone, L., Marconcini, M.: Domain adaptation problems: A DASVM classification technique and a circular validation strategy. IEEE T-PAMI 32(5), 770–787 (2010)
2. Duan, L., Tsang, I.W., Xu, D., Chua, T.S.: Domain adaptation from multiple sources via auxiliary classifiers. In: ICML (2009)
3. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV (2013)
4. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR (2012)
5. Gong, B., Sha, F., Grauman, K.: Overcoming dataset bias: An unsupervised domain adaptation approach. In: NIPS workshop on Big Data Meets Computer Vision (2012)
6. Gretton, A., Borgwardt, K., Rasch, M., Schlkopf, B., Smola, A.: A kernel method for the two sample problem. In: NIPS. pp. 513–520 (2007)
7. Habrard, A., Peyrache, J.P., Sebban, M.: Iterative Self-labeling Domain Adaptation for Linear Structured Image Classification. International Journal on Artificial Intelligence Tools (Dec 2013)

8. Hoffman, J., Kulis, B., Darrell, T., Saenko, K.: Discovering latent domains for multisource domain adaptation. In: ECCV (2012)
9. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A., Torralba, A.: Undoing the damage of dataset bias. In: ECCV (2012)
10. Tommasi, T., Tuytelaars, T.: A testbed for cross-dataset analysis. In: ECCV workshop on Task-CV (2014), https://sites.google.com/site/crossdataset/
11. Visual Computing Research Group, School of Computer Engineering, N.T.U.: Domain adaptation datasets and source codes. http://vc.sce.ntu.edu.sg/transfer_learning_domain_adaptation/domain_adaptation_home.html