Discrepancy-Based Active Learning for Domain Adaptation

Antoine de Mathelin, François Deheeger, Mathilde Mougeot, Nicolas Vayatis

▶ **To cite this version:**

Antoine de Mathelin, François Deheeger, Mathilde Mougeot, Nicolas Vayatis. Discrepancy-Based Active Learning for Domain Adaptation. International Conference on Learning Representations, 2022, Online, United States. hal-03161091v2

HAL Id: hal-03161091
https://hal.science/hal-03161091v2

Submitted on 17 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discrepancy-Based Active Learning for Domain Adaptation

Antoine de Mathelin[1,2] **Francois Deheeger**[1] **Mathilde Mougeot**[2] **Nicolas Vayatis**[2]
[1]Manufacture Française des Pneumatiques Michelin, Clermont-Ferrand, France
[2]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, Gif-sur-Yvette, France
antoine.de-mathelin-de-papigny@michelin.com

Abstract

The goal of the paper is to design active learning strategies which lead to domain adaptation under an assumption of covariate shift in the case of Lipschitz labeling function. Building on previous work by Mansour et al. (2009) we adapt the concept of discrepancy distance between source and target distributions to restrict the maximization over the hypothesis class to a localized class of functions which are performing accurate labeling on the source domain. We derive generalization error bounds for such active learning strategies in terms of Rademacher average and localized discrepancy for general loss functions which satisfy a regularity condition. A practical K-medoids algorithm that can address the case of large data set is inferred from the theoretical bounds. Our numerical experiments show that the proposed algorithm is competitive against other state-of-the-art active learning techniques in the context of domain adaptation, in particular on large data sets of around one hundred thousand images.

1 Introduction

Machine learning models trained on a labeled data set from a *source* domain may fail to generalize on new *target* domains of interest [55]. This issue, which can be caused by *domain shift*, can be handled when no target labels are available through unsupervised domain adaptation methods [18]. Using a small sample of labeled target data can, besides, greatly improve the model performances [45]. Acquiring such new labels is often expensive [58] and one seeks to query as few labels as possible. This explains why strategies of optimal labels acquisition, referred as *active learning* [8] seem very promising for domain adaptation [62].

Active learning is a challenging task and a broad literature exists. On the one hand, some active learning methods introduce heuristic approaches which provide the benefit of using practical algorithm based on simple criteria. For instance, the spatial coverage of the target domain [27, 7] or the minimization of target uncertainties [54, 17] are considered, as well as combination of these heuristics [70, 34]. However finding the proper heuristics is not straightforward and previous methods do not link their query strategy with the target risk [68]. On the other hand, active learning methods based on distribution matching aim at minimizing a distribution distance between the labeled set and the unlabeled one [3, 69, 68]. These methods provide theoretical guarantees on the target risk through generalization bounds. However the computation of the distances is either not scalable to large-scale data sets [3, 69, 68] or based on adversarial training [62, 59] which involves complex hyper-parameter calibration [37].

In this work, we propose to address the issue of active learning for general loss functions under domain shift through a distribution matching approach based on discrepancy minimization [42]. In Section 2, we derive theoretical results by adopting a localized discrepancy distance [73] between the labeled and unlabeled empirical distributions. This localized discrepancy is defined as the plain discrepancy

Preprint. Under review.

considered on a hypothesis space restricted to hypotheses close to the labeling function on the labeled data set. This distance has the benefit to focus only on relevant candidates for approximating the labeling function and thus provides tighter bound of the target risk under some given assumptions [73]. Based on this distance and assuming that the labeled function belongs to the hypothesis space, we provide a generalization bound of the target risk involving pairwise distances between sample points (Theorem 1). Inspired by this generalization error bound, we propose in Section 3 an accelerated version of the K-medoids query algorithm with a better scalability. In Section 4, we present the related works and analytically show that our proposed approach displays tighter theoretical control of the target risk than the one provided by recent active learning methods. We finally present in Section 5 the benefit of the proposed approach on several empirical regression and classification active learning problems in the context of domain adaptation.

2 Discrepancy Based Active Learning

Setup and theory develop in this section mainly focus on regression tasks. Section 5 presents how the algorithm derived from the theoretical results can be extended to classification tasks.

2.1 Setup and definitions

Given two subsets $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}^q$ and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ a distance on $\mathcal{X}$, we denote the source data set $\mathcal{S} = \{x_1, ..., x_m\} \in \mathcal{X}^m$ and the target data set $\mathcal{T} = \{x'_1, ..., x'_n\} \in \mathcal{X}^n$. We consider the *domain shift* setting where the respective data sets $\mathcal{S}$ and $\mathcal{T}$ are drawn according to two different distributions Q and P on $\mathcal{X}$. We consider a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ and a hypothesis space H of k-Lipschitz functions from $\mathcal{X}$ to $\mathcal{Y}$. We denote by $\mathcal{L}_D(h, h') = \mathrm{E}_{x\sim D}[L(h(x), h'(x))]$ the average loss (or risk) over any distribution D on $\mathcal{X}$ between two hypotheses $h, h' \in H$. We also define the expected Rademacher complexity of H for the distribution P as:

$$\mathfrak{R}_n(H) = \mathop{\mathrm{E}}_{\{x'_i\}_i\sim P}\left[\mathop{\mathrm{E}}_{\{\sigma_i\}_i\sim U}\left[\sup_{h\in H}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(x'_i)\right]\right],$$

with σ_i drawn according to U the uniform distribution on $\{-1, 1\}$.

We make the *covariate-shift* assumption [63, 6] stating that the labeling function $f : \mathcal{X} \to \mathcal{Y}$ is the same on the source and target domains, i.e. f returns the true output label for any $x \sim Q$ or $x' \sim P$. We further consider the *noisy setting* adapted from [21] where the labels on $\mathcal{Y}$ are recorded through an oracle function $g : \mathcal{X} \to \mathcal{Y}$ with bounded noise of maximal level $\epsilon \geq 0$, i.e. for any $x \in \mathcal{S} \cup \mathcal{T}$, $L(g(x), f(x)) \leq \epsilon$. We finally consider the single-shot batch active learning framework [68] where all queried data are picked in one single batch of fixed budget of $K > 0$ queries. In this framework, an active learning algorithm takes as inputs the source data set $\mathcal{S}$ along with its corresponding recorded labels $\{g(x); x \in \mathcal{S}\}$ as well as the target data set $\mathcal{T}$. The algorithm then returns a batch of K queried target data denoted $\mathcal{T}_K \subset \mathcal{T}$. The corresponding labels for $\mathcal{T}_K$ are then recorded through the oracle g and used along with the source labeled data to fit an hypothesis $h \in H$. The goal is to select the K target data to label in order to minimize the target risk of $h : \mathcal{L}_P(h, f)$.

2.2 Localized discrepancy

To formulate the problem of active learning under domain shift as a distribution matching problem, one need to consider a measure of divergence between distributions. Recent interest focuses on the discrepancy [42] which proves to be useful for domain adaptation [12, 72] and is recently used on the active learning setting [68]. However this metric, defined as a maximal difference between domain losses over the whole hypothesis space, is relatively conservative as it includes hypotheses that the learner might not ever consider as candidates for the labeling function [13, 73]. Based on this consideration, we introduce a localized discrepancy [73] to restrict the measure of divergence between domains on a set of relevant candidate hypotheses for approximating the labeling function:

Definition 1. ***Localized Discrepancy.*** *Let $K > 0$ be the number of queries and $\epsilon \geq 0$ the maximal noise level of the oracle function g. Let $\mathcal{T}_K$ be a queried batch of size K, the empirical distributions of $\mathcal{S} \cup \mathcal{T}_K$ and $\mathcal{T}$ are respectively denoted $\widehat{Q}_K$ and $\widehat{P}$. Let H be a hypothesis space and L a loss function. The localized discrepancy is defined as:*

$$\mathit{disc}_{H^K_\epsilon}(\widehat{Q}_K, \widehat{P}) = \max_{h,h'\in H^K_\epsilon} \left|\mathcal{L}_{\widehat{Q}_K}(h, h') - \mathcal{L}_{\widehat{P}}(h, h')\right|, \tag{1}$$

with $H_\epsilon^K = \{h \in H;\, L(h(x), g(x)) \leq \epsilon \;\; \forall x \sim \widehat{Q}_K\}$

The localized space H_ϵ^K includes hypotheses "consistent" with the recorded labels, i.e. hypotheses fitting the labeled data with an error below the maximal noise level ϵ and thus approaching the labeling function f of at most 2ϵ on the labeled set. Notice that if $f \in H$ then f is included in H_ϵ^K. In this case, H_ϵ^K is a reduced hypothesis set of potential candidates for f. Under this assumption, a preliminary result is an empirical target risk bound for the localized discrepancy adapted from the one of [42]:

Proposition 1. *Let $K > 0$ be the number of queries, H a hypothesis space and $\epsilon \geq 0$. Let $\widehat{P}$ and $\widehat{Q}_K$ be the empirical distributions of the respective sets $\mathcal{T}$ and $\mathcal{S} \cup \mathcal{T}_K$ of respective size n and $m + K$. We assume that $f \in H$ and that L is a symmetric, μ-Lipschitz and bounded loss function verifying the triangular inequality. We denote by M the bound of L. For any hypothesis $h \in H_\epsilon^K$ and any $\delta > 0$, the following generalization bound holds with at least probability 1-δ:*

$$\mathcal{L}_P(h, f) \leq 2\epsilon + \text{disc}_{H_\epsilon^K}(\widehat{Q}_K, \widehat{P}) + 2\mu\mathfrak{R}_n(H) + M\left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}\right). \tag{2}$$

2.3 Main results: generalization bounds for active learning

Considering the previous bound (Proposition 1) it appears that a natural way of choosing the K queries in an active learning perspective is to pick the target data minimizing the localized discrepancy. Unfortunately this is a difficult problem for an arbitrary functional space H, since it leads to compute a maximum over the set space H_ϵ^K. Our main idea is then to further bound the localized discrepancy with a computable criterion. This can be done by considering the assumption $f \in H$ which leads to the following theorem:

Theorem 1. *Let $K > 0$ be the number of queries, H a hypothesis space of k-Lipschitz functions and $\epsilon \geq 0$. Let $\mathcal{S} \cup \mathcal{T}_K$ be the labeled set and $\mathcal{T}$ the target set drawn according to P with empirical distribution $\widehat{P}$. We assume that $f \in H$ and that L is a symmetric, μ-Lipschitz and bounded loss function verifying the triangular inequality. We denote by M the bound of L. For any hypothesis $h \in H_\epsilon^K$ and any $\delta > 0$, the following generalization bound holds with at least probability 1-δ:*

$$\mathcal{L}_P(h, f) \leq 4\,\epsilon + \frac{2k\mu}{n}\sum_{x' \in \mathcal{T}} d(x', \mathcal{S} \cup \mathcal{T}_K) + 2\mu\mathfrak{R}_n(H) + M\left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}\right) \tag{3}$$

With $d(x', \mathcal{S} \cup \mathcal{T}_K) = \min_{x \in \mathcal{S} \cup \mathcal{T}_K} d(x', x)$

Visual insights to understand Theorem 1 are presented in Figure 1 : the main idea is to approximate the maximal hypotheses $h, h' \in H_\epsilon^K$ returning the localized discrepancy by the k-Lipschitz envelope of the labeling function f, consistent with f on the labeled points, i.e. at most 2ϵ close to f on these points. Indeed, for any target point $x' \in \mathcal{T}$, the gap between h, h' on $x' : L(h'(x'), h(x'))$ is upper bounded by twice the distance from x' to its closest labeled point times the Lipschitz constants of L and h, h' plus the maximal error on the labeled point : $L(h'(x'), h(x')) \leq 2k\mu\, d(x', \mathcal{S} \cup \mathcal{T}_K) + 2\epsilon$.

The generalization bound of Theorem 1 involves the maximal noise level ϵ and the Lipschitz constant k of the hypothesis space H. This last parameter is linked to the complexity of H, complex hypotheses presenting higher Lipschitz constant k. In practice, the considered hypothesis space H should be chosen complex enough to ensure $f \in H$ but with enough regularization to avoid overfitting the noisy labels (see Section 4.3).

It is interesting to notice that the criterion involving the queried batch $\mathcal{T}_K$, i.e. $\sum_{x' \in \mathcal{T}} d(x', \mathcal{S} \cup \mathcal{T}_K)$ is now independent of the hypothesis complexity (characterized by k), it only involves pairwise distances between sample points. As this criterion is computable, we can now propose an active learning strategy.

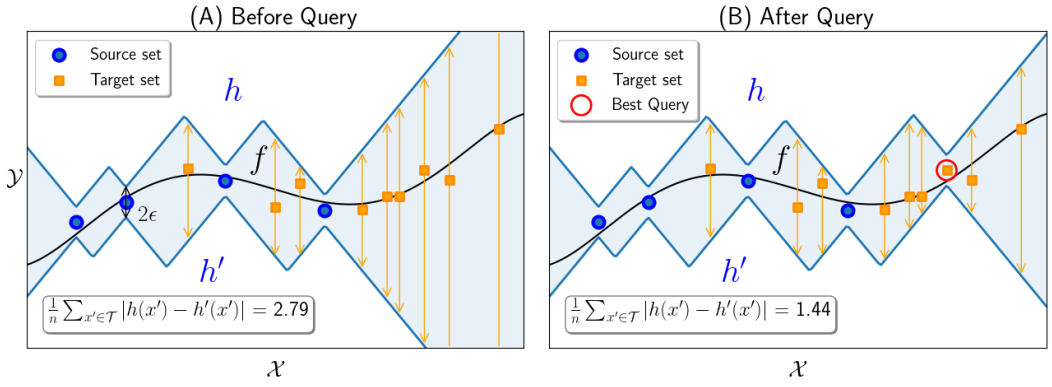

Figure 1: **Visual insights of Theorem 1** : Any potential candidates for f in H^K_ϵ returns values between h and h' : two hypotheses 2ϵ close to f on the labeled points and with slopes of factor k everywhere. Thus, an approximation of the localized discrepancy is given by the mean of gaps between h and h' (mean length of orange arrows) which can be approximated by the distance to the labeled set in $\mathcal{X}$ times k. The best target point to label is chosen in order to minimize this sum.

3 Discrepancy-Based Algorithm

Theorem 1 directly implies that selecting the K queries minimizing $\sum_{x'\in\mathcal{T}} d(x', \mathcal{S}\cup\mathcal{T}_K)$ leads to minimize an upper bound of the target risk. We can thus propose an algorithm dedicated to active learning which provides theoretical guarantees on the target risk.

Seeking the K queries minimizing $\sum_{x'\in\mathcal{T}} d(x', \mathcal{S}\cup\mathcal{T}_K)$ corresponds to solve a K-medoids problem [33]. Notice however that it does not consist of a K-medoids performed directly on the target domain as the source data are already labeled and considered as medoids. The presented algorithm only differs in the initialization process, where the distance between each target and its nearest source neighbour needs to be computed.

Several algorithms exist to solve or approximate the K-medoids [32], [49], [51]. Here we use the greedy version of the algorithm which already provides a $(1-1/e)$-approximation of the optimum [46], and never more than a 0.98-approximation in practice [36].

It is well known that K-medoids algorithms suffer from computational burdens or memory issues on large and moderately large data sets ($\sim$ 100K data) [48]. Indeed they require to compute huge pairwise distance matrix between the source and target samples as well as between targets. Precisely, the greedy K-medoids algorithm presents a complexity of $\mathcal{O}(p(nm+n^2)+Kn^2)$ and a memory usage of $\mathcal{O}(nm+n^2)$ with m, n, K the respective size of $\mathcal{S}$, $\mathcal{T}$ and $\mathcal{T}_K$. p is the dimension of $\mathcal{X}$.

To handle this issue, we propose an adaptation of the K-medoids greedy algorithm with better scalability (Algorithm 1). This algorithm performs the following steps:

1) Computation of the distance to the closest source for each target points via a KD-trees random forest algorithm [60] of complexity $\mathcal{O}(T(m+pn)\log(m)^2)$ with T the number of trees.

2) Medoids initialization using the greedy algorithm on a random target batch of B samples with complexity $\mathcal{O}((K+p)B^2)$.

3) An iterative algorithm on the model of [51], combining assignment of each target point to its closest medoid ($\mathcal{O}(Kpn)$) and medoid update inside each cluster.

4) The medoid update for each cluster is done through an original Branch-and-Bound algorithm [38] which estimates the criterion by iteration over mini-batch, target points for which criterion is bigger than a statistical threshold are left aside. Thus the number of pairwise distances to compute is reduced at each iteration until the maximal iteration number is reached or all target points are left aside. Under some assumptions, we can show that the complexity of the update for all cluster is $\mathcal{O}(n^{3/2}K^{-1/2})$ (the proof is given in the supplementary material).

The overall complexity is then $\mathcal{O}(T(m+pn)\log(m)^2 + (K+p)B^2 + Kpn + pn^{3/2}K^{-1/2})$ which provides reasonable computational time for moderately large data set ($n, m \sim 10^5$ and $p \sim 10^3$). Empirical comparison of computational time is also provided in the supplementary material.

Algorithm 1 Accelerated K-medoids

$\{d(x', \mathcal{S})\}_{x' \in \mathcal{T}} \leftarrow$ **KDT-Forest-Nearest-Neighbour**$(\mathcal{S}, \mathcal{T})$ *# Compute distances to source*
$\mathcal{T}_K = \{x_i^*\}_{i=1..K} \leftarrow$ **Kmedois-Greedy**$(\mathcal{T}, B, K, \{d(x', \mathcal{S})\}_{x' \in \mathcal{T}})$ *# Initialize medoids*
$C_i \leftarrow \{x' \in \mathcal{T} \,|\, d(x', x_i^*) = \min_{j=1..K} d(x', x_j^*) \wedge d(x', \mathcal{S})\}$ *# Assign targets to closest cluster*
while *any medoid* x_i^* *can be updated* **do**
  for i from 1 to K **do**
    $x_i^* = \operatorname{argmin}_{x' \in C_i} \sum_{x'' \in C_i} d(x', x'') \leftarrow$ **Branch-and-Bound**(C_i) *# Update medoid*
  end for
  For all $x' \in \mathcal{T}$, compute $d(x', x_i^*)$ for all updated medoids and reassign x' to the closest C_i
end while

4 Related Work and Discussion

4.1 Related work

Active learning as distribution matching. Active learning methods based on distribution matching aim at reducing the gap between the distributions of the labeled sample and the unlabeled one with a minimal query budget. Several metrics are used to measure the gap between distributions as the Transductive Rademacher Complexity [22], the MMD [69, 64, 68] the Disagreement Coefficient [24, 3, 5, 9, 11, 10], the $\mathcal{H}$-divergence [61, 19, 62] or the Wasserstein distance [59]. To the best of our knowledge, only one paper deals with the discrepancy for active learning [68]. The authors consider the discrepancy on the space of RKHS hypotheses with PSD kernels and provide an explicit way of computing exactly the discrepancy using eigen-value analysis. However the corresponding algorithm encounters computational burden and could hardly be applied on large data sets.

K-medoids for active learning. Many active learning methods use a K-medoids algorithm as an heuristic measure of representativeness [40, 20, 71, 29, 74]. The K-medoids is in general computed on a smaller set of selected targets with the higher uncertainties [70, 34]. In this present work, we provide theoretical insights for this algorithm by highlighting the link with discrepancy minimization.

Active learning for domain adaptation. Our work is related to the recent advances on active learning for domain adaptation as we also consider the domain shift hypothesis [53, 56, 15, 62]. These works use in general the output of a domain classifier to measure the informativeness of target samples. In our work, we consider instead the distance to the source sample to capture informative target data.

Lipschitz consistent functions for active learning. In the context of function optimization, some methods consider the set of Lipschitz or locally Lipschitz functions consistent with the observations [67, 21, 41]. We use similar functions to approximate the maximal hypotheses returning the localized discrepancy. Notice that the goal of the previous papers differ from ours as they aim at finding the maximum of the labeling function.

4.2 Comparison with existing generalization bounds for active learning

The paper by Sener & Savarese [57] proposes the K-centers algorithm for active learning based on an easily computable criterion offering theoretical guarantees. For a regression loss L and $\epsilon = 0$, it controls the target risk as follows:

$$\mathcal{L}_P(h, f) \leq 2k\mu \max_{x' \in \mathcal{T}} d(x', \mathcal{S} \cup \mathcal{T}_K) + 2\mu\mathfrak{R}_n(H) + \mathcal{O}\left(\sqrt{M^2 \log(\frac{1}{\delta})/2n}\right)$$

We directly observe that this bound is looser than the one of equation (3). Indeed, in our case the target risk is controlled with the mean of distances between unlabeled points and the labeled set whereas Sener & Savarese consider the maximum of these distances. This theoretical observation

is also confirmed in our empirical experiments (see Section 5) as our proposed algorithm brings to lower target risks than the greedy algorithm minimizing the criterion proposed by these authors.

A generalization bound for active learning involving the Wasserstein distance W_1 has also been proposed [59]. For $\epsilon = 0$ and under the assumption of Theorem 1 this bound is written:

$$\mathcal{L}_P(h, f) \leq 2k\mu W_1(\widehat{Q}_K, \widehat{P}) + 2\mu\mathfrak{R}_n(H) + \mathcal{O}\left(\sqrt{M^2 \log(\frac{1}{\delta})/2n}\right)$$

Where $W_1(\widehat{Q}_K, \widehat{P}) = \text{argmin}_{\gamma\in\Gamma} \sum_{x\sim\widehat{Q}_K} \sum_{x'\sim\widehat{P}} \gamma_{xx'} d(x, x')$ with $\Gamma = \{\gamma \in \mathbb{R}^{n\times(m+K)}\,;\, \gamma\mathbf{1} = \frac{1}{n}\mathbf{1}\,;\, \gamma^T\mathbf{1} = \frac{1}{m+K}\mathbf{1}\}$.

Thus, in observing that $d(x, x') \geq d(x', \mathcal{S} \cup \mathcal{T}_K)$ for any $x' \in \mathcal{T}$ and $x \in \mathcal{S} \cup \mathcal{T}_K$ we can show that our presented bound of equation (3) is tighter than the one above. Notice however that this comparison is made in the specific case of $\epsilon = 0$ and k-Lipschitz f. For $\epsilon \neq 0$ the two bounds are not comparable. Besides, a more general assumption is considered in [59] as the proposed bound extends to f being "$\phi(\lambda)$-(Q_K, P) Joint Probabilistic Lipschitz" [66].

4.3 Discussion about the assumptions and limitations

In this work, we make the assumption of covariate-shift i.e. we suppose that $Q \neq P$ but that $f_Q = f_P = f$. This hypothesis holds in a variety of scenarios such as sample bias [12] and is considered in several works in the field of domain adaptation [63, 28, 6]. This assumption justifies the usefulness of source data for the selection of target points and for learning the task. If this hypothesis is not verified, the information provided by the source data labels is limited, it may become necessary to label target points close to the sources depending on the level of deviation between f_P and f_Q. In this case, instance-based transfer learning approaches reducing the importance of source data in conflict with target labels can be useful [14, 50].

The assumption of the k-Lipschitzness of $h \in H$ is rather standard. If H is a set of neural networks, the hypotheses can be made Lipschitz with an arbitrary constant through weight clipping. The assumption that $f \in H$ may be considered strong as the complexity of f is unknown a priori. It is indeed assumed in the experiments that the considered neural network architecture and the weight clipping level are suited to learn the labeling function, i.e. that $f \in H$. This assumption is realistic as a large sample of source labeled data are available to tune through grid-search the architecture of the network and the projecting constant. Thus, the complexity of H can be calibrated in order to be appropriate for the learning of f on the source domain and then on the target domain if the covariate-shift assumption holds.

Regularity assumptions on the loss function L are essentially verified by norms as the L_p which are common losses for regression problems. However, the regularity assumptions are not verified by most classification losses : for instance, the cross-entropy is not symmetric and does not verify triangular inequality. We have observed that applying the basic K-medoids algorithm for classification tasks is not the most appropriate choice. In fact classification loss as the cross-entropy is bounded between 0 and 1 and can not increase linearly with the distance to the closest labeled point. Considering target points far away from sources as interesting points is not the best thing to do. In fact, the most interesting points are the ones in the margin between classes [4]. Thus, in order to focus the K-medoids selection in the margin we propose for classification an improved version of our algorithm weighting each target sample with the BVSB confidence score [31] obtained through a hypothesis fitted on the source labeled set.

The choice of the distance d to consider is an important issue. The common choice is the L_1 or L_2, however they might not be the most meaningful distances depending on the application (for instance with images). Some works have considered to use the L_2 distance in an embedding given by the last layer of a neural network $h \in H$ fitted on the source data [57, 2]. Indeed one can consider that the features obtained in the last layer are more relevant for the task at hand. We have observed that it can provides improvement for classification problems when coupled with uncertainty weighting as suggested in [2]. For the classification case, we propose an improved algorithm referred as **K-medoids Weighted Embedded (W+E)**. An ablation study highlights the influence of weighting and embedding in the supplementary material.

5 Experiments

We choose to compare the performances of our algorithm to classical active learning methods on regression and classification problems in a domain shift context. We consider the single-batch active learning setting [68] where all queries are taken at the same time in one batch. We compare the results obtained on the target domain for different query and training methods. The experiments have been run on a (2.7GHz, 16G RAM) computer. The source code is provided on GitHub [1]. We use the open source code of the corresponding authors for BADGE [2] and WAAL [59].

5.1 Competitors

We compare the proposed approach with the following query methods : **Random Sampling**; **K-means** [27]; **K-centers** [57]; **Diversity** [30]; **QBC** [54]; **BVSB** [31]; **AADA** [62] : hybrid active learning method for domain adaptation using a combination of entropy measure from a classifier and the outputs of a domain discriminator; **WAAL** [59] : active learning method on the model of AADA using the Wasserstein loss for the discriminator instead of the cross-entropy; **BADGE** [2] : hybrid deep active learning method optimizing for both uncertainty and diversity.

We select four different training methods **Uniform Weighting**; **Balanced Weighting** : Assign balanced total weight between source and target instances; **TrAdaBoost** [50] : Transfer learning regression method based on a reverse boosting principle; **Adversarial training** : unsupervised features transformation on the model of DANN [18].

To make a fair comparison between the different query strategies, we use for all experiments, the same set of training hypotheses H. We define H as the set of neural networks composed of two fully connected hidden layers of 100 neurons, with ReLU activations and projection constraints on the layer norms ($< 1.$). We use the Adam optimizer [35]. The network architecture is defined to be complex enough that the network provides a good approximation of the labeling function on the source domain in all experiments. Besides, for each experiment, fine-tuning of the optimization hyper-parameters (epochs, batch sizes...) is performed using only source labeled data. We assume that the architecture and the resulting hyper-parameters will still be appropriate after adding the queried target data to the training set (see Section 4.3). Finally, we consider the L_1 as base distance for K-medoids, Diversity and K-centers algorithms, we use an ensemble of 10 models in QBC.

5.2 Superconductivity data set

As there is very few public data sets for domain adaptation with regression tasks [65], we choose an UCI data set with a reasonable amount of instances and split it in different domains using the setup of [50]. We choose *Superconductivity* [23, 16] which is composed of features extracted from superconductors chemical formula. The task consists in predicting their critical temperature.

Experimental setup: The data set is divided in four domains following [50] : low (l), middle-low (ml), middle-high (mh) and high (h) of around 4000 instances and 166 features. We use a learning rate of 0.001, a number of epochs of 100, a batch size of 128 and the mean squared error as loss function. We conduct an experiment for the 12 pairs of domains. For each experiment the number of queries varies from 5 to 300 and the experiment is repeated 8 times with different random seeds. We report the mean absolute error (MAE) on the target unlabeled data for all experiments when $K = 20$ in Table 1. We also present the evolution of this MAE for the adaptation from mh to h in Figure 2.

Results: We observe on Figure 2 that, for any $K > 0$, the K-medoids algorithm presents the lowest MAE on the target data for the three different training methods. In particular we observe a significant performance gain of K-medoids when using TrAdaBoost which provides the lowest MAE for the majority of fixed budget K compared to other training methods. These observations are confirmed on Table 1 where we observe that K-medoids presents the lowest MAE in 10 experiments over 12. We also observe here that methods based on spatial consideration as K-medoids, K-means and K-centers select more informative target points than the uncertainty based method QBC. This comes from the fact that, in batch mode, QBC is selecting close target points with similar uncertainty level. We notice besides that only K-medoids and K-means perform significantly better than Random. We observe indeed that Diversity and K-centers query extreme target points and thus select a batch which is not

[1] https://github.com/antoinedemathelin/dbal

Table 1: MAE on the critical temperature for the Superconductivity experiments with Balanced Weighting and $K = 20$. Standard deviation are given in the supplementary material.

Experiment	l→ml	l→mh	l→h	ml→l	ml→mh	ml→h	mh→l	mh→ml	mh→h	h→l	h→ml	h→mh
Random	15.33	15.80	17.45	16.53	11.39	14.70	17.65	12.83	10.36	18.75	14.86	10.54
Kmeans	15.22	14.74	**15.39**	**15.30**	9.86	13.65	16.33	14.21	10.23	18.49	14.23	10.19
QBC	20.00	19.03	20.08	15.89	12.24	15.31	20.78	12.87	10.19	31.88	18.86	10.65
Kcenters	19.21	15.73	16.85	15.75	11.62	13.44	22.17	**12.74**	10.24	36.50	19.60	10.39
Diversity	19.52	17.75	24.89	15.81	12.36	17.08	23.66	14.11	10.45	39.65	23.92	11.41
Kmedoids	**13.82**	**13.01**	15.82	15.57	**9.27**	**12.79**	**14.71**	**12.74**	**9.33**	**16.61**	**14.10**	**9.67**

representative of the target distribution. Finally, K-medoids outperforms K-means because it takes into account the distance to source points and then queries less redundant information.

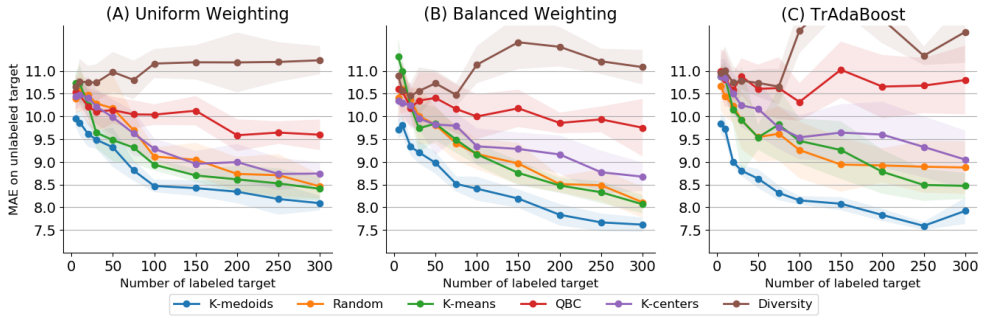

Figure 2: Results for the Superconductivity data set (mh → h experiment). Evolution of the MAE in function of the budget K for three different training methods and six query methods.

5.3 Office data set

The office data set [55] consists in pictures of office items coming from different domains: amazon or webcam. The task is a multi-classification problem with 31 classes (chairs, printers, ...). The goal is to use data from the amazon domain where labels are easily available to learn a good model on the webcam domain where a few labels are chosen using active learning methods.

Experimental setup: We consider the adaptation from the amazon domain with 2817 labeled images to the webcam domain with 795 unlabeled images. We use, as input features, the outputs of the ResNet-50 network [25] pretrained on ImageNet, leading to 2048 features. We vary the number of queries from 5 to 300, repeating each experiment 8 times. The learning rate is set to 0.001, the number of epochs to 60 and the batch size to 128. We use the categorical cross-entropy as loss function and a softmax activation at the last layer.

Results: Figure 3.A presents the results for the different query strategies with the Balanced Weighting training method. We observe that the K-medoids W+E algorithm provides the best performances for almost any K and in particular for $K \leq 100$. We then present the visualization of the two first components of the PCA transform on Figure 4. We observe that the K-medoids algorithm queries points at the center of the target distribution but at a reasonable distance from the sources. The Random and K-means algorithms select a representative subset of the target distribution but without taking into account the sources and therefore query redundant information. K-centers selects data far from the source domain but which are less representative of the the target distribution.

5.4 Digits data set

We consider the experiment proposed in [18] where a synthetic digits data set: SYNTH is used to learn a classification task for a data set of real digits pictures: SVHN (Street-View House Number) [47]. The first data set is composed of 60000 images of size 28×28 and the second of 73257 images.

Experimental setup: To handle the large number of data we use the accelerated K-medoids algorithm (Algorithm 1) with $T = 50$ trees and a initial batch size of $B = 5000$. The algorithm is improved with the (W+E) extension. Besides, we use the KD-trees random forest nearest neighbour algorithm

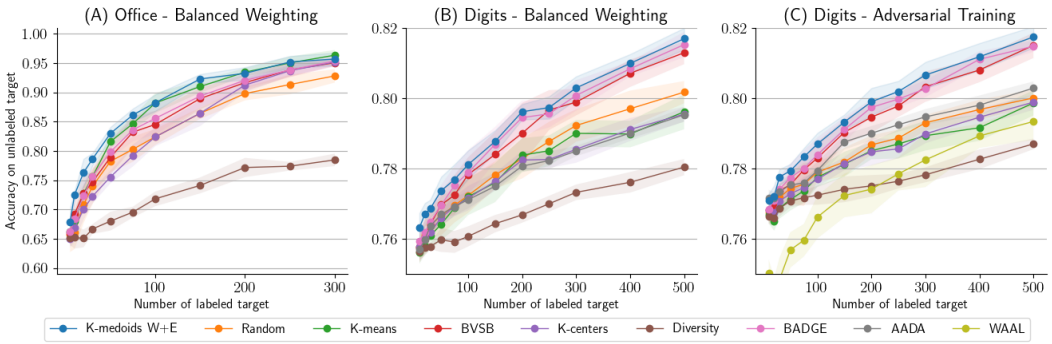

Figure 3: Office and digits results. Evolution of the accuracy in function of the budget K.

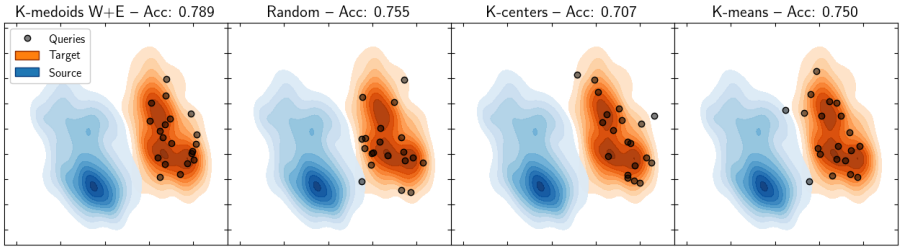

Figure 4: Visualization of the two PCA first components of the Office data set input space. Queries are reported with black points for each query method with $K = 20$.

in Diversity and K-centers to approximate the distance of each target data to the source data set. Finally, we use the mini-batch version of the K-means algorithm. We consider two kinds of input features: the ones obtained with the convulational part of a Lenet [39] trained with the source labeled data and the ones coming from the same network but trained with adversarial training following the model of DANN [18]. In both cases, the network is pre-trained on 30 epochs with a batch size of 128 and a learning rate of 0.001, for the adversarial training the trade-off parameter λ is set to 0.1 following the setup of [62]. After the query process, a Balance Weighting training is performed with the source and target labeled data using the same optimization hyper-parameters than before. Experiments are conducted 8 times for K between 10 and 500.

Results: Figure 3.B and 3.C correspond to the evolution of accuracy with respect to K for the experiments conducted with the features obtained respectively without and with adversarial training. We first observe that adversarial training improves the accuracy by around 1% for all query strategies for small values of K. For $K = 500$, performances are similar between the two training methods. However, the choice of training algorithm does not change the relative performances between query strategies. An exception is made for AADA which better performs with adversarial training. This is expected as AADA is specially designed for adversarial training. We observe that, for any $K > 0$, K-medoids W+E outperforms other query strategies in both cases. The method is followed by BADGE and BVSB. BADGE is indeed closely related to K-medoids W+E as it considers a combination of uncertainty and distance criteria in an embedded space. However its distance criterion, based on the kmeans++ seeding algorithm [1], is close to the K-centers criterion and is then less accurate than the one of K-medoids (cf Section 4.2).

6 Conclusion and Future Work

This work introduces a novel active learning approach based on a localized discrepancy between the labeled and unlabeled distributions. We provide both theoretical guarantees of this approach and an active learning algorithm scaling to large data sets. Several experiments show very competitive results of the proposed approach. Future work will focus on considering a more appropriate distance on the input space, giving more importance to relevant features with respect to the task.

References

[1] David Arthur and Sergei Vassilvitskii. k-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2019.

[3] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[4] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *International Conference on Computational Learning Theory*, pages 35–50. Springer, 2007.

[5] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. *Importance Weighted Active Learning*, page 49–56. Association for Computing Machinery, 2009.

[6] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9), 2009.

[7] Zalán Bodó, Zsolt Minier, and Lehel Csató. Active learning with clustering. In *Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010*, pages 127–139. JMLR Workshop and Conference Proceedings, 2011.

[8] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[9] Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Region-based active learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2801–2809, 2019.

[10] Corinna Cortes, Giulia Desalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Adaptive region-based active learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2144–2153. PMLR, 13–18 Jul 2020.

[11] Corinna Cortes, Giulia DeSalvo, Mehryar Mohri, Ningshan Zhang, and Claudio Gentile. Active learning with disagreement graphs. In *International Conference on Machine Learning*, pages 1379–1387, 2019.

[12] Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519, 2014.

[13] Corinna Cortes, Mehryar Mohri, and Andrés Muñoz Medina. Adaptation based on generalized discrepancy. *J. Mach. Learn. Res.*, 20(1):1–30, January 2019.

[14] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, volume 227, pages 193–200, 01 2007.

[15] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.

[16] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[19] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

[20] Ryan Gomes and Andreas Krause. Budgeted nonparametric learning from data streams. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 391–398, Madison, WI, USA, 2010. Omnipress.

[21] Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Neural Information Processing Systems*, 2015.

[22] Q. Gu and J. Han. Towards active learning on graphs: An error bound minimization approach. In *2012 IEEE 12th International Conference on Data Mining*, pages 882–887, 2012.

[23] Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, 2018.

[24] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pages 353–360, 2007.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[27] Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Off to a good start: Using clustering to select the initial training set in active learning. In *FLAIRS Conference*, 2010.

[28] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J. Smola. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, 2007.

[29] Rishabh K Iyer and Jeff A Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, pages 2436–2444, 2013.

[30] S. D. Jain and K. Grauman. Active image segmentation propagation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2864–2873, 2016.

[31] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379, 2009.

[32] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[33] Leonard Kaufmann and Peter Rousseeuw. Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, pages 405–416, 01 1987.

[34] V. Kaushal, R. Iyer, S. Kothawade, R. Mahadev, K. Doctor, and G. Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1299, 2019.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[36] Andreas Krause. *Optimizing Sensing: Theory and Applications*. PhD thesis, PhD thesis, Carnegie Mellon University.(Section 5.7), USA, 2008.

[37] Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *International Conference on Machine Learning*, pages 3581–3590. PMLR, 2019.

[38] Ailsa H Land and Alison G Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008*, pages 105–132. Springer, 2010.

[39] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[40] Hui Lin, Jeff Bilmes, and Shasha Xie. Graph-based submodular selection for extractive summarization. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 381–386. IEEE, 2009.

[41] Cédric Malherbe and Nicolas Vayatis. Global optimization of lipschitz functions. In *International Conference on Machine Learning*, pages 2314–2323. PMLR, 2017.

[42] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009.

[43] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[44] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[45] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5715–5725, 2017.

[46] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[47] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[48] James Newling and François Fleuret. A sub-quadratic exact medoid algorithm. In *Artificial Intelligence and Statistics*, pages 185–193. PMLR, 2017.

[49] Raymond T. Ng and Jiawei Han. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016, 2002.

[50] David Pardoe and Peter Stone. Boosting for regression transfer. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, June 2010.

[51] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[53] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

[54] T. RayChaudhuri and L. G. C. Hamey. Minimisation of data collection by active learning. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 3, pages 1338–1341 vol.3, 1995.

[55] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, page 213–226, Berlin, Heidelberg, 2010. Springer-Verlag.

[56] Avishek Saha, Piyush Rai, Hal Daumé, Suresh Venkatasubramanian, and Scott L DuVall. Active supervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 97–112. Springer, 2011.

[57] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[58] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52, 07 2010.

[59] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pages 1308–1318. PMLR, 2020.

[60] Chanop Silpa-Anan and Richard Hartley. Optimised kd-trees for fast image descriptor matching. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[61] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5972–5981, 2019.

[62] J. Su, Y. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker. Active adversarial domain adaptation. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 728–737, 2020.

[63] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1433–1440, Red Hook, NY, USA, 2007. Curran Associates Inc.

[64] Ying-Peng Tang and Sheng-Jun Huang. Self-paced active learning: Query the right thing at the right time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5117–5124, 2019.

[65] Takeshi Teshima, Issei Sato, and Masashi Sugiyama. Few-shot domain adaptation by causal mechanism transfer. In *International Conference on Machine Learning*, pages 9458–9469. PMLR, 2020.

[66] Ruth Urner, Sharon Wulff, and Shai Ben-David. Plal: Cluster-based active learning. In *Conference on Learning Theory*, pages 376–397. PMLR, 2013.

[67] Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *International Conference on Machine Learning*, pages 19–27. PMLR, 2013.

[68] Tom Viering, Jesse Krijthe, and Marco Loog. Nuclear discrepancy for single-shot batch active learning. *Machine Learning*, 06 2019.

[69] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Trans. Knowl. Discov. Data*, 9(3), February 2015.

[70] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In Francis Bach and David Blei, editors, *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1954–1963, Lille, France, 07–09 Jul 2015. PMLR.

[71] Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. Using document summarization techniques for speech data subset selection. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 721–726, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[72] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[73] Yuchen Zhang, Mingsheng Long, Jianmin Wang, and Michael I Jordan. On localized discrepancy for domain adaptation. *arXiv preprint arXiv:2008.06242*, 2020.

[74] Jingjing Zheng, Zhuolin Jiang, Rama Chellappa, and Jonathon P Phillips. Submodular attribute selection for action recognition in video. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1341–1349. Curran Associates, Inc., 2014.

Appendix

We recall here the notations and definitions used in the following:

- $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}$ are the respective input and output subsets.
- $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a distance on $\mathcal{X}$.
- P and Q are two distributions on $\mathcal{X}$.
- $\mathcal{T} = \{x'_1, ..., x'_n\} \in \mathcal{X}^n$ is the unlabeled target data set and $\mathcal{S} = \{x_1, ..., x_m\} \in \mathcal{X}^m$ the labeled source data set drawn respectively from P and Q.
- $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function.
- H is a hypothesis set of k-Lipschitz functions from $\mathcal{X}$ to $\mathcal{Y}$.
- $\mathrm{E}_{x\sim D}[L(h(x), h'(x))]$ is the average loss (or risk) over any distribution D on $\mathcal{X}$ between two hypotheses $h, h' \in H$.
- $\mathfrak{R}_n(H) = \underset{\{x'_i\}_i \sim P}{\mathrm{E}} \left[\underset{\{\sigma_i\}_i \sim U}{\mathrm{E}} \left[\sup_{h\in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x'_i) \right] \right]$, is the expected Rademacher complexity of H, with U the uniform distribution on $\{-1, 1\}$.
- $f : \mathcal{X} \to \mathcal{Y}$ is the true labeling function on the source and target domains such that f returns the true output label for any $x' \sim P$ or $x \sim Q$.
- $g : \mathcal{X} \to \mathcal{Y}$ is the oracle function such that g returns the recorded output label for any $x \in \mathcal{S} \cup \mathcal{T}$.
- ϵ is the maximal noise level of the function g such that for any $x \in \mathcal{S} \cup \mathcal{T}$, $L(g(x), f(x)) \leq \epsilon$.
- $K > 0$ is the number of queries
- $\mathcal{T}_K \subset \mathcal{T}$ with $|\mathcal{T}_K| = K$ is a queried batch or subset.
- $\mathscr{L}_K = \mathcal{S} \cup \mathcal{T}_K$ is the labeled data set.
- $\widehat{Q}$, $\widehat{Q}_K$ and $\widehat{P}$ are the respective empirical distributions on $\mathcal{X}$ of $\mathcal{S}$, $\mathcal{S} \cup \mathcal{T}_K$ and $\mathcal{T}$.
- $H_\epsilon^K = \{h \in H; L(h(x), f(x)) < \epsilon \; \forall x \in \mathscr{L}_K\}$ is the localized hypothesis space .
- $\mathrm{disc}_{H_\epsilon^K}(\widehat{Q}_K, \widehat{P}) = \max_{h,h' \in H_\epsilon^K} |\mathcal{L}_{\widehat{Q}_K}(h, h') - \mathcal{L}_{\widehat{P}}(h, h')|$ is the localized discrepancy between $\widehat{Q}_K$ and $\widehat{P}$.

A Proof of Proposition 1

Proposition 1. Let $K > 0$ be the number of queries, H a hypothesis space and $\epsilon \geq 0$. Let $\widehat{P}$ and $\widehat{Q}_K$ be the empirical distributions of the respective sets $\mathcal{T}$ and $\mathcal{S} \cup \mathcal{T}_K$ of respective size n and $m + K$. We assume that $f \in H$ and that L is a symmetric, μ-Lipschitz and bounded loss function verifying the triangular inequality. We denote by M the bound of L. For any hypothesis $h \in H_\epsilon^K$ and any $\delta > 0$, the following generalization bound holds with at least probability 1-δ:

$$\mathcal{L}_P(h, f) \leq 2\epsilon + \mathrm{disc}_{H_\epsilon^K}(\widehat{Q}_K, \widehat{P}) + 2\mu\mathfrak{R}_n(H) + M\left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}\right) . \tag{4}$$

Proof. Let's consider $h \in H$. According to [44] we have for any $\delta > 0$, with probability at least $1 - \delta$:

$$\mathcal{L}_P(h, f) \leq \mathcal{L}_{\widehat{P}}(h, f) + 2\mu\mathfrak{R}_n(H) + M\sqrt{\frac{\log(\frac{1}{\delta})}{2n}} . \tag{5}$$

Besides, in noticing that $f \in H_\epsilon^K$, we have for any $h \in H_\epsilon^K$:

$$\begin{aligned}\mathcal{L}_{\widehat{P}}(h,f) &= \mathcal{L}_{\widehat{Q}_K}(h,f) + \mathcal{L}_{\widehat{P}}(h,f) - \mathcal{L}_{\widehat{Q}_K}(h,f) \\ &\leq \mathcal{L}_{\widehat{Q}_K}(h,g) + \mathcal{L}_{\widehat{Q}_K}(g,f) + \max_{h \in H_\epsilon^K} |\mathcal{L}_{\widehat{P}}(h,f) - \mathcal{L}_{\widehat{Q}_K}(h,f)| \\ &\leq 2\epsilon + \max_{h,h' \in H_\epsilon^K} |\mathcal{L}_{\widehat{P}}(h,h') - \mathcal{L}_{\widehat{Q}_K}(h,h')|\end{aligned} \tag{6}$$

Thus we conclude that for any $h \in H_\epsilon^K$:

$$\mathcal{L}_P(h,f) \leq 2\epsilon + \text{disc}_{H_\epsilon^K}(\widehat{Q}_K, \widehat{P}) + 2\mu\mathfrak{R}_n(H) + M\left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}\right). \tag{7}$$

□

B Proof of Theorem 1

Theorem 1. Let $K > 0$ be the number of queries, H a hypothesis space of k-Lipschitz functions and $\epsilon \geq 0$. Let $\mathcal{S} \cup \mathcal{T}_K$ be the labeled set and $\mathcal{T}$ the target set drawn according to P with empirical distribution $\widehat{P}$. We assume that $f \in H$ and that L is a symmetric, μ-Lipschitz and bounded loss function verifying the triangular inequality. We denote by M the bound of L. For any hypothesis $h \in H_\epsilon^K$ and any $\delta > 0$, the following generalization bound holds with at least probability 1-δ:

$$\mathcal{L}_P(h,f) \leq 4\,\epsilon + \frac{2k\mu}{n}\sum_{x' \in \mathcal{T}} d(x', \mathcal{S} \cup \mathcal{T}_K) + 2\mu\mathfrak{R}_n(H) + M\left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}\right) \tag{8}$$

With $d(x', \mathcal{S} \cup \mathcal{T}_K) = \min_{x \in \mathcal{S} \cup \mathcal{T}_K} d(x', x)$

Proof. Let $\epsilon \leq 0$ and $K > 0$. We define the labeled data set $\mathscr{L}_K = \mathcal{S} \cup \mathcal{T}_K$ with $\mathcal{T}_K$ one queried subset such that $|\mathcal{T}_K| = K$.

For all $h, h' \in H_\epsilon^K$, for all $x' \in \mathcal{T}$ and for all $x \in \mathscr{L}_K$ we have:

$$\begin{aligned}L(h(x'), h'(x')) &\leq L(h(x'), h(x)) + L(h(x), h'(x)) + L(h'(x), h'(x')) \\ &\leq L(h(x'), h(x)) + L(h'(x), h'(x')) + L(h(x), g(x)) + L(g(x), h'(x)) \\ &\leq \mu\left(|h(x) - h(x')| + |h'(x) - h'(x')|\right) + 2\epsilon \\ &\leq 2k\mu\, d(x', x) + 2\epsilon\end{aligned} \tag{9}$$

The two first inequalities come from the triangular inequality, the others from the lipschitzness of h, h' and definition of H_ϵ^K.

As the above inequality is true for any $x \in \mathscr{L}_K$, we have in particular for any $x' \in \mathcal{T}$:

$$\begin{aligned}L(h(x'), h'(x')) &\leq 2k\mu \min_{x \in \mathscr{L}_K} d(x, x') + 2\epsilon \\ &= 2k\mu\, d(x', \mathscr{L}_K) + 2\epsilon\end{aligned} \tag{10}$$

Leading to:

$$\mathcal{L}_{\widehat{P}}(h(x'), h'(x')) \leq \frac{2k\mu}{n}\sum_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) + 2\epsilon \tag{11}$$

We then deduce the following, for all $h, h' \in H_\epsilon^K$:

$$\begin{aligned}
\text{disc}_{H_\epsilon^K}(\widehat{Q}_K, \widehat{P}) &= \max_{h,h' \in H_\epsilon^K} |\mathcal{L}_{\widehat{P}}(h, h') - \mathcal{L}_{\widehat{Q}_K}(h, h')| \\
&\leq \max \left[\max_{h,h' \in H_\epsilon^K} \mathcal{L}_{\widehat{P}}(h, h'), \max_{h,h' \in H_\epsilon^K} \mathcal{L}_{\widehat{Q}_K}(h, h') \right] \\
&\leq \max \left[\frac{2k\mu}{n} \sum_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) + 2\epsilon, 2\epsilon \right] \\
&\leq \frac{2k\mu}{n} \sum_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) + 2\epsilon
\end{aligned} \tag{12}$$

Finally, according to Proposition 1, we have for all $h, h' \in H_\epsilon^K$:

$$\mathcal{L}_P(h, f) \leq 4\,\epsilon + \frac{2k\mu}{n} \sum_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) + 2\mu \mathfrak{R}_n(H) + M \left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}} \right) \tag{13}$$

□

C Comparison with other active learning bounds (cf section 4.2)

C.1 K-center bounds

Sener and Savarese [57] propose the following generalization bounds for $\epsilon = 0$ and $h \in H_\epsilon^K$:

$$\mathcal{L}_{\widehat{P}}(h, f) \leq \delta(\lambda^l + MC\lambda^\mu) + M \left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}} \right) \tag{14}$$

which leads to:

$$\mathcal{L}_P(h, f) \leq \delta(\lambda^l + MC\lambda^\mu) + 2\mu \mathfrak{R}_n(H) + 2M \left(\sqrt{\frac{\log(\frac{1}{\delta})}{2n}} \right) \tag{15}$$

With $\delta = \max_{x' \in \mathcal{T}} d(x', \mathscr{L}_K)$, C the class number and λ^μ the Lipschitz constant of a class-specific regression function. λ^l is the Lipschitz constant of the loss function l verifying $l : (x, h) \to l(x, f(x), h) = L(h(x), f(x))$.

If we consider a regression problem, we can drop the term corresponding to the class-specific function and we have:

$$\mathcal{L}_P(h, f) \leq \delta\lambda^l + 2\mu \mathfrak{R}_n(H) + \mathcal{O} \left(\sqrt{\frac{M^2 \log(\frac{1}{\delta})}{2n}} \right) \tag{16}$$

Let's now consider $h \in H$ and $x, x' \in \mathcal{X}$, we have for k-Lipschitz f :

$$\begin{aligned}
|l(x, f(x), h) - l(x', f(x'), h)| &= |L(h(x), f(x)) - L(h(x'), f(x'))| \\
&\leq |L(h(x), f(x)) - L(h(x'), f(x))| + |L(h(x'), f(x)) - L(h(x'), f(x'))| \\
&\leq L(h(x), h(x')) + L(f(x), f(x')) \\
&\leq \mu|h(x) - h(x')| + \mu|f(x) - f(x')| \\
&\leq 2\mu k|x - x'|
\end{aligned} \tag{17}$$

The two first inequalities are obtained with triangular inequalities, then we use Lipschitz assumptions on h, f and L.

Thus, we have $\lambda^l = 2\mu k$ from which we deduce:

$$\mathcal{L}_P(h,f) \leq 2k\mu \max_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) + 2\mu\mathfrak{R}_n(H) + \mathcal{O}\left(\sqrt{\frac{M^2 \log(\frac{1}{\delta})}{2n}}\right) \tag{18}$$

C.2 Wasserstein bounds

Shui et al. [59] write the following generalization bounds:

$$\mathcal{L}_P(h,f) \leq \ \mathcal{L}_{\widehat{Q}_K}(h,f) + \mu(k+\lambda) W_1(\widehat{Q}_K, \widehat{P}) + \mu\phi(\lambda) + 2\mu\mathfrak{R}_n(H) + \mathcal{O}\left(\sqrt{\frac{M^2 \log(\frac{1}{\delta})}{2n}}\right) \tag{19}$$

With λ, ϕ verifying:

$$P_{(x,x') \sim \gamma} \left(|h(x) - h(x')| > \lambda d(x,x') \right) \leq \phi(\lambda) \tag{20}$$

We remark that for $\lambda = k$ we have $\phi(\lambda) = 0$ as h is k-Lipschitz.

Thus for $\epsilon = 0$ and for any $h \in H_\epsilon^K$, we have:

$$\begin{aligned} \mathcal{L}_P(h,f) &\leq 2k\mu W_1(\widehat{Q}_K, \widehat{P}) + 2\mu\mathfrak{R}_n(H) + \mathcal{O}\left(\sqrt{\frac{M^2 \log(\frac{1}{\delta})}{2n}}\right) \\ &= 2k\mu \sum_{x' \in \mathcal{T}} \sum_{x \in \mathscr{L}_K} \gamma^*_{x'x}\, d(x', x) + 2\mu\mathfrak{R}_n(H) + \mathcal{O}\left(\sqrt{\frac{M^2 \log(\frac{1}{\delta})}{2n}}\right) \end{aligned} \tag{21}$$

With $\gamma^* = \operatorname{argmin}_{\gamma \in \Gamma} \sum_{x' \in \mathcal{T}} \sum_{x \in \mathscr{L}_K} \gamma_{x'x} d(x', x)$ and $\Gamma = \{\gamma \in \mathbb{R}^{n \times (m+K)}\,;\, \gamma \mathbf{1} = \frac{1}{n}\mathbf{1}\,;\, \gamma^T \mathbf{1} = \frac{1}{m+K}\mathbf{1}\}$.

Thus, in observing that:

$$\begin{aligned} \sum_{x' \in \mathcal{T}} \sum_{x \in \mathscr{L}_K} \gamma_{x'x} d(x', x) &\geq \sum_{x' \in \mathcal{T}} \sum_{x \in \mathscr{L}_K} \gamma_{x'x} \min_{x \in \mathscr{L}_K} d(x', x) \\ &\geq \sum_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) \sum_{x \in \mathscr{L}_K} \gamma_{x'x} \\ &\geq \frac{1}{n} \sum_{x' \in \mathcal{T}} d(x', \mathscr{L}_K) \end{aligned}$$

We deduce that our presented bound of Theorem 1 is also tighter than the one proposed in [59] in the case of $\epsilon = 0$ and k-Lipschitz f and h.

D Algorithms (cf Section 3)

D.1 K-medoids

Algorithm 2 K-Medoids Greedy

Input: $\mathcal{T}$, B, K, $\mathcal{S}$, $\{d(x', \mathcal{S})\}_{x' \in \mathcal{T}}$
Output: $\mathcal{T}_K = \{x_j\}_{j \leq K} \subset \mathcal{T}$
$\mathcal{T} \leftarrow \{x'_{s_1}, ..., x'_{s_B}\}$ with $\{x'_{s_1}, ..., x'_{s_B}\}$ picked randomly in $\mathcal{T}$ without replacement.
Initialize query subset: $\mathcal{T}_0 = \{\}$
For all $x' \in \mathcal{T}$, $d^{x'} = d(x', \mathcal{S}) = \min_{x \in \mathcal{S}} d(x', x)$
For all $x, x' \in \mathcal{T}$ compute $d^{xx'} = d(x, x')$
for i from 1 to K **do**
    $x_i \leftarrow \text{argmin}_{x \in \mathcal{T}} \sum_{x' \in \mathcal{T}} \min\left(d^{xx'}, d^{x'}\right)$
    $\mathcal{S}_i \leftarrow \mathcal{S}_{i-1} \cup \{x_i\}$
    For all $x' \in \mathcal{T}$, update $d^{x'} = \min(d^{x'}, d^{x_i x'})$
end for

Algorithm 3 Branch & Bound Medoid (B & B)

Input: Cluster $C \in \mathbb{R}^{n_c \times p}$, previous medoid criterion $\mathcal{C}^*$, batch size B
Output: New medoid x^*
Initialize candidates $\tilde{C} = C$
Initialize computed distance set $D_x = \{\}$ for all $x \in \tilde{C}$.
Initialize criterion $\mathcal{C}_x = 0$ and standard deviation $\sigma_x = 0$ for all $x \in \tilde{C}$.
Initialize threshold $t = \mathcal{C}^*$
for i from 1 to n_c/B **do**
    $C_i = \{x_j \in C \ ; \ (B-1)i \leq j \leq Bi\}$
    for $x \in \tilde{C}$ **do**
        $\mathcal{D}_x \leftarrow \mathcal{D}_x \cup \{d(x, x') \ ; \ x' \in C_i\}$
        $\mathcal{C}_x \leftarrow \frac{1}{Bi} \sum_{d \in \mathcal{D}_x} d$
        $\sigma_x \leftarrow \sqrt{\frac{1}{Bi} \sum_{d \in \mathcal{D}_x} (d - \mathcal{C}_x)^2}$
    end for
    $x^* \leftarrow \text{argmin}_{x \in \tilde{C}} \mathcal{C}_x$
    $t \leftarrow \min(t, \mathcal{C}_{x^*} + \frac{2\sigma_{x^*}}{\sqrt{Bi}})$
    $\tilde{C} \leftarrow \left\{x \in \tilde{C} \ ; \ \mathcal{C}_x - \frac{2\sigma_x}{\sqrt{Bi}} < t\right\}$
end for
$x^* \leftarrow \text{argmin}_{x \in \tilde{C}} \mathcal{C}_x$

D.2 Complexity Computation

In the following, a distance computation is considered to be done in $\mathcal{O}(p)$.

1. **KD-Trees Random Forest**: Each of the T trees is built by splitting one sample, at each root, at the median of a random feature until the leaf-sizes are $\sim \log(m)$. The median computation for each root with m_r data is in $\mathcal{O}(m_r \log(m_r))$. Thus the overall complexity to build one tree is $\mathcal{O}(\sum_{i=0}^{M} 2^{-i} m \log(2^{-i} m) 2^i) = \mathcal{O}(m \log(m) \sum_{i=0}^{M} 1)$ with $M \sim \log(m/\log(m))$ which becomes $\mathcal{O}(m \log(m)^2)$. Then, each target is assigned to a leaf by performing $\mathcal{O}(\log(m))$ computations, inside the assigned leaf all distance computations are done in $\mathcal{O}(p \log(m))$. Thus an approximate nearest neighbour is given for all targets with a complexity $\mathcal{O}(T(m + pn) \log(m)^2)$.

2. **Medoids Initialization**: Using the greedy algorithm, $\mathcal{O}(pB^2)$ distance computations are first done, then, for all targets, a sum over the target data is computed at each of the K steps. Thus the complexity is $\mathcal{O}((K+p)B^2)$.

3. **Assignation to the Closest Medoid**: The distance between all target and the medoids is computed in $\mathcal{O}(Kpn)$.

4. **Branch & Bound Medoid Computation**

   B & B algorithm (Algorithm 3) takes as input one cluster C of n_c unlabeled data from $\mathcal{T}$. It also takes a batch size B and the previous cluster medoid criterion $\mathcal{C}^*$ which is used as an initial threshold. The use of the initialization $\mathcal{C}^*$ may accelerate the algorithm, in the following we do not take into account this initialization, i.e we consider that $\mathcal{C}^* = +\infty$.

   Besides, to use notations consistent with the common notations in statistics, we will denote the batch size B by n ($B \equiv n$). Notice that it is redundant with the size of the unlabeled data set $\mathcal{T}$. An explicit mention will be made, if n does not refer to the batch size.

   Definitions and notations: Let's consider one cluster $C \subset \mathcal{T}$ with n_c data. We consider the uniform norm as underlying distance d, defined for all $x_i, x_j \in C$ as $d(x_i, x_j) = \max\left(|x_i^{(1)} - x_j^{(1)}|, ..., |x_i^{(p)} - x_j^{(p)}|\right)$ with $x_i = (x_i^{(1)}, ..., x_i^{(p)}) \in \mathbb{R}^p$.

   Computing the complexity of the B & B algorithm for any distribution of the x_i would be too difficult. We make here the simplifying assumption that the x_i in C are uniformly distributed on the hyper-cube C of edge size 2 and centered on $x^* = (0, ..., 0) \in \mathbb{R}^p$

   We define for any $i \in [|1, n_c|]$ and any $j \in [|1, n_c|]$, the variables $Z_j^i = d(x_i, x_j)$. We also define $Z_j^* = d(x^*, x_j)$ for any $j \in [|1, n_c|]$. We suppose that for any $i \in [|1, n_c|]$, $Z_1^i, ..., Z_{n_c}^i$ are iid and that for any $j \in [|1, n_c|]$ the Z_j^i are independents. We define, for any $i \in [|1, n_c|]$, the mean $\mu_i = \mathrm{E}[Z_0^i] = \frac{1}{2^p}\int_{x \in C} d(x_i, x)$ and the variance $\sigma_i^2 = \mathrm{Var}[Z_0^i] = \frac{1}{2^p}\int_{x \in C}(d(x_i, x) - \mu_i)^2$. We consider a first batch of distance computations of size $n < n_C$. We define for any $i \in [|1, n_c|]$ the empirical mean $\widehat{\mu}_i = \frac{1}{n}\sum_{j=1}^n Z_j^i$ and the empirical variance $\widehat{\sigma}_i^2 = \frac{1}{n-1}\sum_{j=1}^n (Z_j^i - \widehat{\mu}_i)^2$. We denote by μ^* and σ^{*2} the mean and variance of Z_0^* and $\widehat{\mu}^*$ and $\widehat{\sigma}^{*^2}$ their respective empirical estimator.

   We first observe that μ_i and σ_i are finite for any $i \in [|1, n_c|]$ as the x_i are uniformly distributed on the hyper-cube centered in $x^* \in C$. We also notice that $Z_j^i \in [0, 2]$ for any $i, j \in [|1, n_c|]$.

   Preliminary results: We make the assumption that $p >> 1$. We aim at giving bounds for any μ_i and σ_i. We admit the intuitive results that for any $x_i = (x_i^{(1)}, ..., x_i^{(p)}) \in C$ and for any $j \in [|1, n_c|]$, $\mu_i \geq \mu_i^{j \to 0}$ and $\sigma_i \geq \sigma_i^{j \to 0}$ with $\mu_i^{j \to 0}$ and $\sigma_i^{j \to 0}$ the mean and variance of the variable $d(x_i^{j \to 0}, .)$ with $x_i^{j \to 0} = (x_i^{(1)}, ..., x_i^{(j-1)}, 0, x_i^{(j+1)}, ..., x_i^{(p)})$. We consider indeed that the more x_i is close to the center of the hyper-cube smaller is μ_i and σ_i. Considering this fact, we have, for any $i \in [|1, n_c|]$:

   $$\mu^* \leq \mu_i \tag{22}$$
   $$\sigma^* \leq \sigma_i \tag{23}$$

   Besides, for any $0 \leq r \leq 1$ the sample density in the elementary surface between the balls centered on x^* and of respective radius $r + \mathrm{d}r$ and r is $pr^{p-1}\mathrm{d}r$. Thus, we can notice that Z_0^* follows a beta distribution of parameters $\alpha = p$ and $\beta = 1$, from which we deduce that:

   $$\mu^* = \frac{p}{p+1} \tag{24}$$
   $$\sigma^{*2} = \frac{p}{(p+1)^2(p+2)} \tag{25}$$

   We further notice that for any $0 \leq a \leq 1$ and for any $i \in [|1, n_c|]$ such that $d(x^*, x_i) = a$ we have $\mu_i \geq \mu_a$ with μ_a the criterion of the sample $x_a = (a, 0, ..., 0) \in \mathbb{R}^p$.

To compute $\mu_a = \frac{1}{2^p}\int_{x\in C} d(x_a, x)$, we split the integral on three parts: $d(x_a, x) \leq 1-a$, $1-a \leq d(x_a, x) \leq 1$ and $1 \leq d(x_a, x) \leq 1+a$:

$$\begin{aligned}
\mu_a &= \int_{r=0}^{1-a} pr^{p-1}\mathrm{d}r + \frac{1}{2}\int_{r=1-a}^{1}\left(pr^{p-1} + (p-1)(1-a)r^{p-2}\right)\mathrm{d}r + \frac{1}{2}\int_{1}^{1+a} r\mathrm{d}r \\
&= \frac{p}{p+1}(1-a)^{p+1} + \frac{1}{2}\left[\frac{p}{p+1} + (1-a)\frac{p-1}{p} - (1-a)^{p+1}\left(\frac{p}{p+1} + \frac{p-1}{p}\right)\right] + \frac{a}{2}\left(1+\frac{a}{2}\right) \\
&\simeq 1 - \frac{a}{2} + \frac{a}{2}\left(1+\frac{a}{2}\right) \\
&\simeq 1 + \frac{a^2}{4}
\end{aligned} \tag{26}$$

Using the simplifying approximation $\frac{p}{p+1} \simeq \frac{p-1}{p} \simeq 1$ for $p >> 1$. Thus for any $0 \leq a \leq 1$ and for any $i \in [|1, n_c|]$ such that $d(x^*, x_i) = a$ we have:

$$\mu_i \geq 1 + \frac{a^2}{4} \tag{27}$$

An upper bound of the σ_i^2 is given by the variance σ_c^2 of the variable $d(x_c, .)$ with $x_c = (1, ..., 1)$ which corresponds to one corner of the hyper-cube C:

$$\begin{aligned}
\sigma_c^2 &= \frac{1}{2^p}\int_0^2 pr^{p+1}\mathrm{d}r - \left(\frac{1}{2^p}\int_0^2 pr^p\mathrm{d}r\right)^2 \\
&= 4\frac{p}{p+2} + 4\left(\frac{p}{p+1}\right)^2 \\
&= 4\frac{p}{(p+1)^2(p+2)} \\
&= 4{\sigma^*}^2
\end{aligned} \tag{28}$$

From which we deduce that for any $i \in [|1, n_c|]$:

$$\sigma^* \leq \sigma_i \leq 2\sigma^* \tag{29}$$

To simplify the calculations, we make the approximation $\widehat{\sigma}_i \simeq \sigma_i$ for any $i \in [|1, n_c|]$, which is relevant for sufficiently large n as $\mathrm{E}[{Z_0^i}^4] < +\infty$.

Probability of rejecting all optimal medoid candidates: Let $\epsilon > 0$ be an approximation factor of μ^*. The goal of the Branch & Bound algorithm is to identify one sample $x_i \in C$ such that $\mu_i \leq \mu^*(1+\epsilon)$ with less distance computations as possible. The process consists in removing all candidates x_i such that $\widehat{\mu}_i - \frac{2\widehat{\sigma}_i}{\sqrt{n}} > \widehat{\mu}_{i^*} + \frac{2\widehat{\sigma}_{i^*}}{\sqrt{n}}$ with $\widehat{\mu}_{i^*}$ the current minimal empirical mean.

We aim now at computing the probability of rejecting all optimal medoid candidates x_i verifying $\mu_i \leq \mu^*(1+\epsilon)$ during the B & B process. For this, we define $\mathcal{B}_\epsilon(\mu^*) = \{i \in [|1, n_c|]\,;\, \mu_i \leq \mu^*(1+\epsilon)\}$ the index set of optimal medoid candidates and $\mathcal{B}_\epsilon^c(\mu^*) = \{i \in [|1, n_c|]\,;\, \mu_i > \mu^*(1+\epsilon)\}$ the index set of sub-optimal medoid candidates. We assume that B & B returns an optimal candidate if at least one sample of $\mathcal{B}_\epsilon(\mu^*)$ is kept after the first batch computation. We define the two following probabilities:

$$P_1 = P\left(\left\{\exists i \in \mathcal{B}_\epsilon^c(\mu^*)\,;\, \widehat{\mu}_i + \frac{4\sigma^*}{\sqrt{n}} \leq \mu^*(1+\epsilon/4)\right\}\right) \tag{30}$$

$$P_2 = P\left(\{\exists i \in \mathcal{B}_\epsilon(\mu^*)\,;\, \widehat{\mu}_i \leq \mu^*(1+\epsilon/4)\}\right) \tag{31}$$

We can observe that the probability of rejecting all optimal medoid candidates is upper bounded by: $(1-P_2) + P_2P_1$ considering the approximation $\widehat{\sigma}_i \simeq \sigma_i$ and the fact that $\sigma_i \geq \sigma^*$ for any $i \in [|1, n_c|]$.

We now define, for $i \in \mathcal{B}_\epsilon^c(\mu^*)$:

$$P_i = P\left(\left\{\widehat{\mu}_i + \frac{4\sigma^*}{\sqrt{n}} \leq \mu^*(1+\epsilon/4)\right\}\right) \tag{32}$$

Leading to:

$$P_i = P\left(\left\{\widehat{\mu}_i + \frac{4\sigma^*}{\sqrt{n}} - \frac{\epsilon\mu^*}{4} \leq \mu^*\right\}\right) \tag{33}$$

On the other hand, according to the Bennett's inequality from [43, 26] we have, for any $i \in \mathcal{B}_\epsilon^c(\mu^*)$ and for any $\delta > 0$:

$$\mathrm{P}\left(\widehat{\mu}_i/2 + \sqrt{\frac{\sigma_i^2 \log(1/\delta)}{2n}} + \frac{\log(1/\delta)}{3n} \leq \mu_i/2\right) \leq \delta \tag{34}$$

Notice that we apply the inequality to the $Z_j^i/2$. Then, considering the fact that $\sigma_i \leq 2\sigma^*$ and that $\mu_i > \mu^*(1+\epsilon)$ we have:

$$\mathrm{P}\left(\widehat{\mu}_i + \frac{\sqrt{2}\sigma^*}{\sqrt{n}}\sqrt{\log(1/\delta)} + \frac{2\log(1/\delta)}{3n} \leq \mu^*(1+\epsilon)\right) \leq \delta \tag{35}$$

Leading to:

$$\mathrm{P}\left(\widehat{\mu}_i + \frac{\sqrt{2}\sigma^*}{\sqrt{n}}\sqrt{\log(1/\delta)} - \epsilon\mu^* + \frac{2\log(1/\delta)}{3n} \leq \mu^*\right) \leq \delta \tag{36}$$

Let's consider $\delta > 0$ such that the following equality holds:

$$\frac{\sqrt{2}\sigma^*}{\sqrt{n}}\sqrt{\log(1/\delta)} - \epsilon\mu^* + \frac{2\log(1/\delta)}{3n} = \frac{4\sigma^*}{\sqrt{n}} - \frac{\epsilon\mu^*}{4} \tag{37}$$

Thus:

$$u^2 + Au = B \tag{38}$$

With:

$$u = \sqrt{\log(1/\delta)} \tag{39}$$

$$A = \frac{3}{2}\sqrt{2n}\sigma^* \tag{40}$$

$$B = \frac{3n}{2}\left(\frac{4\sigma^*}{\sqrt{n}} + \frac{3\epsilon\mu^*}{4}\right) \tag{41}$$

$$\tag{42}$$

We then set:

$$\delta = \exp\left(-\left(\frac{\Delta - A}{2}\right)^2\right) \tag{43}$$

$$\Delta = \sqrt{A^2 + 4B} \tag{44}$$

We have for the δ defines above and for any $i \in \mathcal{B}_\epsilon^c(\mu^*)$:

$$P_i \leq \delta \tag{45}$$

Thus, the probability P_1 can be upper bounded as follows:

$$P_1 \leq 1 - (1-\delta)^{n_c} \tag{46}$$

Considering the fact that $|\mathcal{B}_\epsilon^c(\mu^*)| < n_c$. $1-(1-\delta)^{n_c}$ is the probability of getting at least one success for the binomial law of parameters (n_c, δ).

We are now looking for an upper bound of P_2. We observe that, at least the index i such that $x_i = x^*$ is in $\mathcal{B}_\epsilon(\mu^*)$. Besides, as x^* is the center of the hyper-cube, Z_j^* is in $[0,1]$ for any $j \in [|1, n_c|]$. We can then apply the Bennett's inequality to the Z_j^*, and for any $\gamma > 0$ we have:

$$\mathrm{P}\left(\widehat{\mu}^* \leq \mu^* + \sqrt{\frac{2{\sigma^*}^2 \log(1/\delta)}{n}} + \frac{\log(1/\delta)}{3n}\right) \geq 1 - \gamma \tag{47}$$

We set:

$$\gamma = \exp\left(-\left(\frac{\sqrt{C^2+4D}-C}{2}\right)^2\right) \tag{48}$$

$$C = 3\sqrt{2n}\sigma^* \tag{49}$$

$$D = 3n\frac{\epsilon\mu^*}{4} \tag{50}$$

We then have:

$$P_2 \geq \mathrm{P}\left(\widehat{\mu}^* \leq \mu^*(1+\frac{\epsilon}{4})\right) \geq 1-\gamma \tag{51}$$

Leading to:

$$1 - P_2 \leq \gamma \tag{52}$$

Finally the probability of rejecting all optimal candidates is upper bounded by $1-(1-\delta)^{n_c}+\gamma$. To give an order of magnitude of this probability, we consider the scenario where $n_c = 10^5$, $p = 100$, $n = \sqrt{n_c}$ and $\epsilon = 0.05$. Then we have: $\delta \simeq 3.6\,10^{-8}$ and $\gamma \simeq 7.7\,10^{-5}$, which leads to a probability of rejection around $3.6\,10^{-3}$. Thus, in this case, there is at least a probability 0.995 that B & B returns a medoid candidate with a criterion less than 1.05 the optimal.

Complexity computation: We are now looking for the number of distance computations performed by B & B. For this, we need to compute the number of x_i kept at each batch. An upper bound of this number is given by the number of x_i verifying $\widehat{\mu}_i \leq \mu^*(1+\epsilon/4) + \frac{8\sigma^*}{\sqrt{n}}$. We further assume that the previous upper bound can be approximated by the number of x_i verifying $\mu_i \leq \mu^*(1+\epsilon/4) + \frac{8\sigma^*}{\sqrt{n}}$

We have shown in the preliminary results that for any $i \in [|1, n_c|]$, $\mu_i \geq 1 + \frac{a^2}{4}$ with $a = d(x_i, x^*)$. Thus an upper bound of the number of candidates x_i kept after the first batch is given by the number of x_i in the ball of radius a with a verifying:

$$a = 2\sqrt{\epsilon/4 + \frac{8\sigma^*}{\sqrt{n}}} \tag{53}$$

Using the approximation $\mu^* = \frac{p}{p+1} \simeq 1$. Besides, as $\sigma^* = \mathcal{O}(1/p)$ we can suppose that for sufficiently large p and large n, $\frac{8\sigma^*}{\sqrt{n}} \leq \frac{3}{4}\epsilon$ (For instance with n_c, n, p and ϵ considered previously, we have $\frac{8\sigma^*}{\sqrt{n}} \simeq 0.005$ and $\frac{3}{4}\epsilon \simeq 0.03$). Thus:

$$a \leq 2\sqrt{\epsilon} \tag{54}$$

Finally the number of candidates kept after the first batch is in $\mathcal{O}\left(n_c\,\epsilon^{p/2}\right)$ which is very small (for the values of n_c, p and ϵ considered previously, we have $n_c\,\epsilon^{p/2} \simeq 10^{-60}$). If we consider a batch size of $\mathcal{O}\left(\sqrt{n_c}\right)$, the number of distance computations after the first batch is negligible behind $\mathcal{O}\left(n_c\sqrt{n_c}\right)$.
We then conclude that the complexity of a medoid computation in one cluster is in $\mathcal{O}\left(p\,n_c\sqrt{n_c}\right)$ as each of the K cluster has approximately $n_c \simeq n/K$ samples (with n the sample size of $\mathcal{T}$), the overall complexity of B & B is in $\mathcal{O}\left(p\,n^{3/2}K^{-1/2}\right)$.

E Empirical Complexities

This section presents the empirical time computation recorded for **K-medoids Greedy**, **Accelerated K-medoids**, **K-centers** and **K-centers + KD-Trees** which corresponds to the K-centers algorithm with initialization of the nearest source neighbour distances computed through the KD-Trees Random Forest algorithm. The experiments are conducted on the Digits data set.

The experiments are run on a (2.7GHz, 16G RAM) computer using Python 3.8. The scikit-learn [52] implementation of the pairwise euclidean distance is used. No parallel computing is performed.

The parameters are set to $K = 100$, $T = 50$ for the KD-Trees Random Forest algorithm and an initial batch size $B = 5000$ for the Accelerated K-medoids algorithm. The euclidean distance is used as base distance d. The results are reported on Figure 5. The evolution of computational time is a function of the size of the source and target samples (m and n).

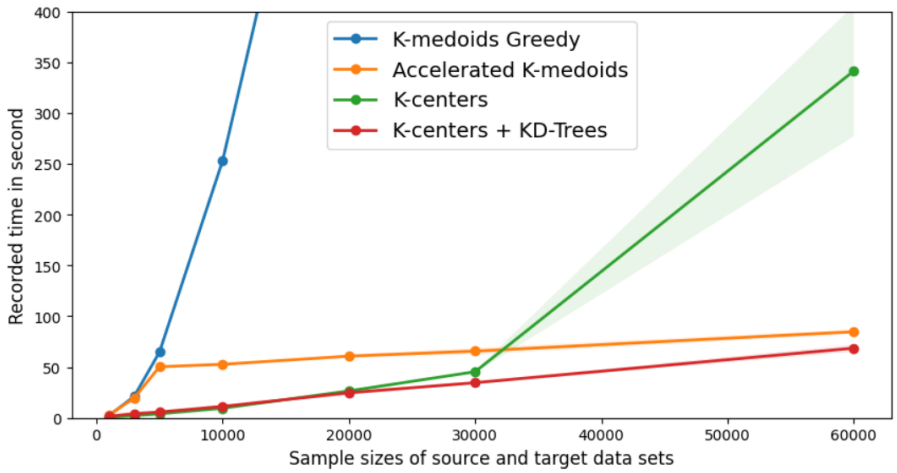

Figure 5: Visualization of empirical computational times in function of the sample sizes.

We first observe on Figure 5 that the K-medoids Greedy algorithm encounters computational burden for samples larger than 10k instances. For $n = m = 60$k, the K-centers algorithm encounters a similar issue due to the computation of the distance matrix between the source and target data sets. Using the KD-Trees Random Forest algorithm decreases in this case the computational time by a factor of 5. We also notice that the accelerated K-medoids algorithm has similar complexity performance to K-medoids for $n, m \leq 5000$ which is the maximum size of the initial batch. Then, the computation time of the algorithm increases slightly from $n, m = 5$k to $n, m = 60$k while remaining at an acceptable level. For $n, m = 60$k the complexities of the accelerated K-medoids and the K-centers + KD-Trees are almost similar but with a level of performance in favor of the accelerated K-medoids (see Section 5.4).

F Ablation Study

This section presents the ablation study of the K-medoids Weighted Embedded algorithm (K-medoids W+E) used for the Office and Digits experiments. We compare the effect of the weighting and the embedding on the relative performances compared to the random strategy (Figure 6). The percentage of accuracy gain over the random strategy is reported on y axis.

The weighting is performed with the BVSB method [31] and the embedding is done with the last layer of the neural network used to learn the task on the source domain [57].

We observe on Figure 6 that the basic K-medoids under-performs the improved versions for most values of labeled target number K. This observation supports the idea that purely distance-based query strategies are not well suited for classification tasks. The weighting improvement provides a significant performance gain in both experiment. The embedding option, however, provides mixed results. For the Digits experiments, using an embedding improves the accuracy gain by 1% in small values of K. In the Office experiment, the use of an embedding seems to decrease the performances as the K-medoids Weighted algorithm provides the best results for any value of K. These mixed results for the "last layer" embedding option may depend on the nature of the data or the architectural priors of the model [2]; however, the influence of these factors is not yet well understood and will be explored in future works.

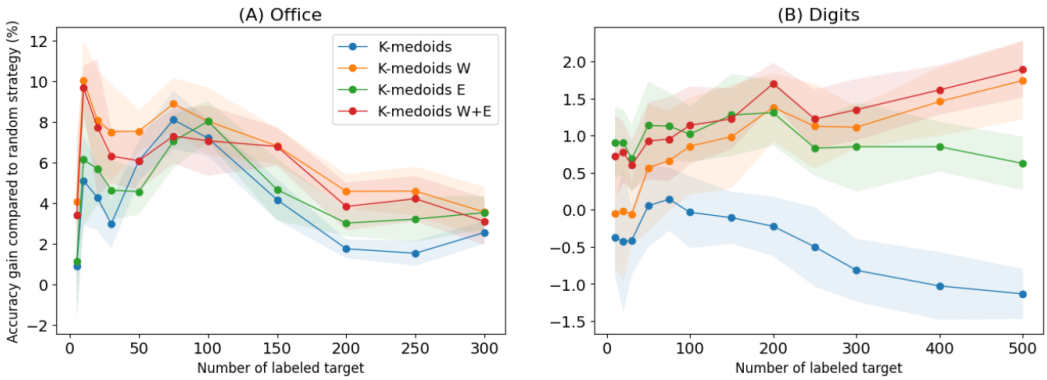

Figure 6: Comparison of the accuracy gain compared to the random query strategy for the K-medoids algorithm and its weighted and embedded extensions. The Balanced Weighting training is considered here.

G Experiments

G.1 Superconductivity

The UCI data set "Superconductivity" [23, 16] is composed of features extracted from the chemical formula of several superconductors along with their critical temperature. There is two kind of features: some features correspond to the chemical element number's in the superconductor chemical formula, others are statistical features derived from the chemical formula as the mean and variance of the atomic mass.

We use the setup of [50] to divide the data set in separate domains. We select an input feature with a moderate correlation factor with the output (~ 0.3). We then sort the set according to this feature and split it in four parts: low (l), middle-low (ml), middle-high (mh), high (h). Each part defining a domain with around 4000 instances and 166 features. The considered feature is then withdrawn from the data set.

A standard scaling preprocessing is performed using the source data on the input statistical features and the output feature. A max scaling is performed on features corresponding to the chemical element number's. A visualization of the first components of the PCA as well as the output distribution is provided in Figure 7).

Table 2 presents the standard deviation of the MAEs obtained on the 8 repetitions of the 12 experiments with the Balanced Weighting training and $K = 20$. We observe that K-medoids provides the smallest standard deviations in 10 of the 12 experiments. K-medoids is indeed a deterministic algorithm and thus selects a determined batch of target points. Besides, K-medoids selects the target batch to label in a distribution matching perspective, i.e. it produces a training set with a distribution close to the one of the testing set (cf Section 2). This can explain why the training is more stable with the training set provided by K-medoids.

Table 2: Standard deviations of the MAE on the unlabeled data for the superconductivity experiments. The deviations are computed using the results of 8 repetitions of each experiment.

Experiment	l→ml	l→mh	l→h	ml→l	ml→mh	ml→h	mh→l	mh→ml	mh→h	h→l	h→ml	h→mh
Random	1.007	2.599	1.597	1.232	1.679	1.291	1.863	0.787	0.266	1.900	1.228	0.595
Kmeans	1.162	1.089	1.040	0.564	0.546	**0.248**	1.045	0.998	0.247	**1.235**	1.414	0.372
QBC	1.469	2.370	1.882	0.571	0.564	0.707	2.151	0.767	0.394	7.240	3.250	0.620
Kcenters	0.807	1.761	1.926	0.626	0.343	0.794	2.188	0.885	0.503	5.403	2.838	0.920
Diversity	1.268	0.961	3.444	0.538	0.900	1.379	1.869	0.628	0.284	2.074	2.741	0.386
Kmedoids	**0.393**	**0.746**	**0.921**	**0.392**	**0.201**	0.459	**0.319**	**0.279**	**0.183**	1.553	**0.327**	**0.351**

G.2 Office

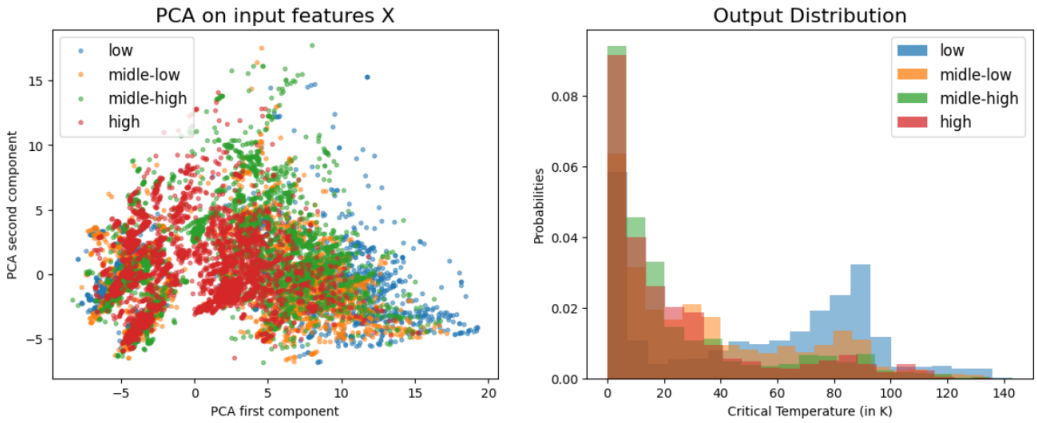

Figure 7: Visualization of the domain shift of the superconductivity data set. The visualization of the two first components of the PCA on the input features is given on the left. The output distribution is given on the right. One domain is represented by one color.

Figure 8: Office data set: examples of images from amazon domain

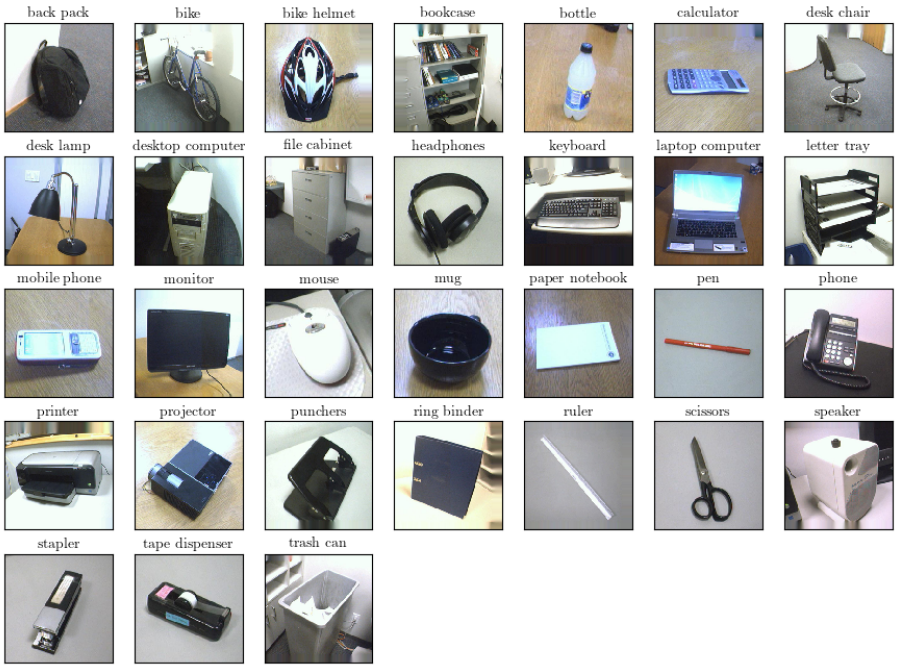

Figure 9: Office data set: examples of images from webcam domain

G.3 Digits

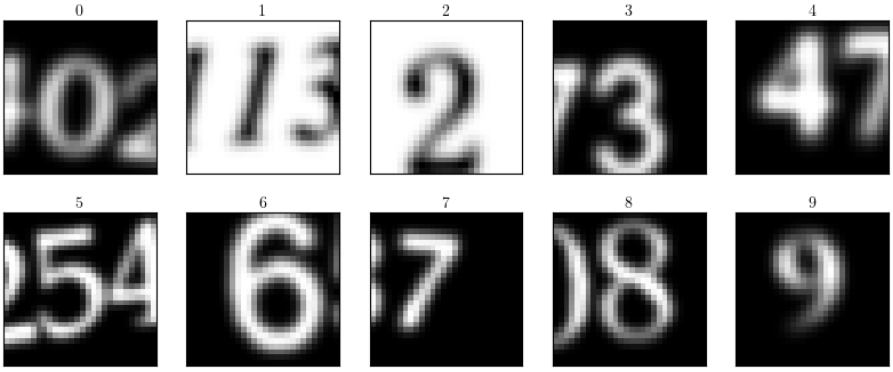

Figure 10: Digits data set: examples of SYNTH images

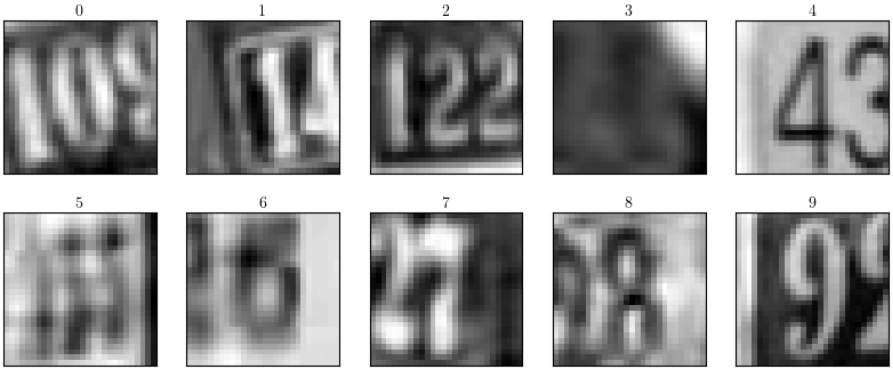

Figure 11: Digits data set: examples of SVHN images