# Projection based transfer learning

Christian Poelitz

Dortmund Technical University
Artificial Intelligence Group
44227 Dortmund, Germany

**Abstract.** We propose greedy selections strategies to identify a small subset of data samples that are most suited for transfer learning. On these samples the transfer learning is done on a subspace in a kernel defined feature space. The sampling strategies make the kernel methods applicable even in large scale scenarios. We validate our proposed method on a benchmark data set and compare to state of the art transfer learning methods.

## 1 Introduction

The usual assumption for most of the Data Mining and Machine Learning tasks is that the training data used to learn a model has the same distribution as the test data on which the model is applied. On the other hand, there are many situation where this is not true. For instance, in a case when new data gets available but no additional information is present to learn a new model.

We assume to have two data sets with (possible large) difference in distribution. On the one hand, we have data from a source domain $S$ that is distributed via $p_s$ together with label information $y$ distributed via $p_s(y|x)$. On the other hand, we also have data from a target domain $T$ that is distributed via $p_t$ with no label information. The transfer learning task now is to use the source domain together with its label information to find a classifier that labels the target domain best.

From bounds on the expected error on a target domain $T$ using only training data from a given source domain $S$, we learn that for transfer learning to be successful, we need at least two things. First, the probability distributions of the two domains must be similar. Hence, $D(p_s, p_t)$ is small, for $D$ any measure of discrepancy of distributions. And second, the difference of the hypothesis from both domain must be small. This can be directly read from the following bound by Ben-David et al. [BDBC+10] from Theorem 1 page 155:

$$\epsilon_t(h) \leq \epsilon_s(h) + D(p_s, p_t) \tag{1}$$

$$+ \min \left( \int |h_s(x) - h_t(x)| p_s(x) dx, \int |h_s(x) - h_t(x)| p_t(x) dx \right) \tag{2}$$

The bound tells, that the expected error $\epsilon_t$ of any hypothesis $h$ on the target domain can be bounded by the expected error $\epsilon_s$ on the same hypothesis trained

on the source domain, the difference in distribution of target and source domain, and the expected difference of any two hypothesis $h_s$ and $h_t$ from the source and target domain. Our goal for transfer learning now will be to minimize this bound by finding a suitable data representation.

Based on several observations many approaches have been proposed to find a proper representation of the data from both domains to account for a good domain adaptation or transfer of knowledge.

One line of research tries to find such feature representations of different data sets that are invariant for both data set distributions. Low dimensional feature representations are used to capture this invariances. These representations are for instance extracted via dimension reduction methods like (kernel) PCA.

To sum up, we want to find a low dimensional representation of the data from source and target domain. This representation shall keep enough structure from the data that a classifier trained on the source domain still perform well on the source and target domain. By this, we want to account for that part of the bound above that integrates the generalization error on the source domain. On the other hand, a low dimensional representation shall make the two data sets more similar and possible hypothesis from the source and target domain closer. This accounts for that part of the bound above that integrates the distance of the distributions of the two data sets and the expected distance in hypothesis from the domains..

Formally, we have the following generic loss for transfer learning:

$$L(p_s(x), p_t(x)) = D(p_s, p_t) + \lambda \cdot d(h_s, h_t)$$

Where $D(p_s, p_t)$ is a discrepancy measure between distributions and $d(h_s, h_t)$ estimates the distance between any two hypothesis trained in the source respectively in the target domain. The parameter $\lambda$ trades off the influence of the discrepancy and the generalization error on the source domain. The task now is to find a low dimensional subspace such that when projection onto this subspace the loss above gets minimized.

We concentrate on kernel methods, since they provide us with high-dimensional and non-linear data representations. Further, via kernel methods we can also integrate structural information and even information from probabilistic models.

A critical issue with kernel methods is that - in the simplest implementation - they scale quadratically or cubically in the number of training samples. To account for the issue different approximation strategies have been proposed. For instance, Nystrom sampling [WS01], Incomplete Cholesky decomposition [STC04] or Kernel Matching pursuit [VB02].

In order to reduce memory and computational effort, we propose a greedy selection strategy that finds the most useful data samples in the source domain for transfer learning. By this, we reduce the data size and concentrate on those samples that are potentially best suited for transfer knowledge. This idea is based on the assumption that not all source samples might by equally important for adaptability. This has been investigated for instance by Gong et al. in [GGS13].

In the next sections we shortly describe kernel methods. Then, we tell how we can measure discrepancies of distributions and how we can use projections onto subspaces to reduce such a discrepancy. Finally, we propose to greedily select samples that spann the subspace for transfer learning to account for large scale transfer learning tasks.

## 2 Kernel Methods and RKHS

Kernel methods accomplish to apply linear methods on non-linear representations of data. Any kernel methods uses a map $X \to \phi(X)$ from a compact input space $X$, for instance $\Re^n$, into a so called Reproducing Kernel Hilbert Space (RKHS). In this space, linear methods are applied to the mapped elements like Linear Regressions or Support Vector Machines. The RKHS is a space of functions $f(y) = \phi(x)(y)\ \forall x \in X$ that allows point evaluations by an inner product, hence $f(y) = \phi(x)(y) =< \phi(x), \phi(y) >$. $\phi(x)$ is a function and $\phi(x)(y)$ mean the function value at $y$.

For the mapping $\phi$ from above, $K_\phi$ is the integral operator as defined in Equation 3 for a probability distribution $P$ on the input space $X$.

$$K_\phi(f)(t) = \int f(x) \cdot < \phi(x), \phi(t) > \cdot dP(x) \tag{3}$$

For this integral operator, we denote $< \phi(x), \phi(y) >= k(x, y)$ with kernel $k$. By Mercer Theorem [Mer09] there is a one to one correspondence of the above defined RKHS and the integral operator via the kernel $k$. This correspondence is given by the expansion $k(x, y) = \sum_{i=1}^{\infty} \phi_i(x) \cdot \phi_j(y)$ for $\{\phi_i\}$ an orthonormal basis in the RKHS.

The covariance operator $C$ on a Hilbert space $H$ is is defined as $E[Z \times Z^*]$ the outer product of a random elements $Z \in H$ with its adjoint $Z^*$. This is analogue to the covariance of centred random elements in $\Re^n$ where we have $C = E[X \cdot X^T]$. The empirical covariance is estimated via $\hat{C} = \frac{1}{m} \sum \phi(x_i) \cdot \phi(x_j)$ for a centred sample $\{\phi(x_1), \cdots, \phi(x_m)\}$ with $x_i$ drawn from $P$.

Schoelkopf et al. [SSM99] proposed to perform Principal Component Analysis (PCA) [Hot33] in a kernel defined RKHS based on the eigenfunctions and eigenvalues of the covariance operator $C$. In Chapter 3.1, we explain how this can be used to extract subspaces that approximate the subspace where the data is mapped to in the RKHS. We will the distance to this subspace in a selection strategy for samplings data such that they match best a different distribution.

As stated by Steinwart in [Ste05], a classifier is universal consistent if it can asymptotically learn every classification task. A classification task is defined as learning a function $f$ that maps inputs $x$ that are distributed via $p(x)$ to outputs $y$ that are distributed via $p(y|x)$. Given samples $(x, y)$ drawn from $p(x, y)$, $f$ is learned. Concentrating on output spaces $Y = \{-1, 1\}$, the decision function is defined as $sign(f(x))$ the signum of the function $f$. The Bayes risk for a sample $(x, y)$ is defined as the infimum of the probability that the decision function agrees with $y$ over all functions $f$. If for samples $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ and

$n \to \infty$ the decision function reaches the Bayes risk, then the corresponding $f$ is said to be universally consistent. Universal kernels induce functions spaces that contain universally consistent functions. In [MXZ06] Micchelli et al. discuss conditions for a kernel to be universal. For instance translation invariant kernels with compact support are universal. One of the most important universal kernel is the Gaussian kernel $k(x, y) = \exp(-\sigma^2 \cdot \|x - y\|_2^2)$ which is used in this work. As introduced by Smola et al. in [SGSS07], universal kernels can be used to map whole distributions into a Hilbert space. The embedding is defined via the mean map $\mu[p] = E[k(x, .)]$ for all probability distributions $p$ with bounded expectation $E$, respectively its empirical version $\mu[X] = \frac{1}{m} \sum_{i=1}^{m} k(x_i, .)$ for a sample $X = \{x_1, \cdots, x_m\}$ with $x_i$ drawn from $p$. For universal kernels $k$, these mappings are injective and can be used to estimate differences in distributions. Later, we will use this fact in a selection strategy for samplings data such that they match best a different distribution.

## 3 Subspace Methods

A subspace in an RKHS $H$ is a closed subset $H' \subset H$. We identify this subspace by a projection $P$ that maps all elements of $H$ to $H'$. In this work, we concentrate only on subspaces that are spanned by the given data points in the RKHS. This means each element in the subspace can be written as linear combination of all data points in the RKHS, hence $v = \sum_{x \in H} \alpha_i \cdot \phi(x_i)$ for all $v \in H'$. In the next section, we explain how kernel PCA can be used to find an appropriated projection matrix onto such a subspace.

### 3.1 Kernel PCA

Kernel Principal Component Analysis [SSM99] extracts an orthogonal basis, also called principal components, in a kernel induced RKHS. Projecting the data onto the subspace spanned by the first $k$ components captures most of the variance among the data compared to all other possible subspaces where the data lies in.

The $k$ components are exactly the eigenfunctions corresponding to the largest $k$ eigenvalues of the covariance operator of the kernel.

The covariance operator is approximated by the empirical covariance matrix $C = \frac{1}{b} \sum_i \phi(x_i) \cdot \phi(x_i)^T$. An eigenvalue decomposition on $C$ results in a set of eigenvalues $\{\lambda_i\}$ and eigenvectors $\{v_i\}$ such that $\lambda_i \cdot v_i = C \cdot v_i$.

A projection of a sample $x$ in the RKHS onto $U = \{v_i\}$ is done by $P_U(\phi(x)) = (< v_i, \phi(x) >, \cdots, < v_k, \phi(x) >) \in U$. Since, the $v_i$ lie in the span of the $\{\phi(x_i)\}$, each component is given by $v_i = \sum_j \alpha_{j,i} \cdot \phi(x_j)$. This results in the projection $P_U(\phi(x)) = (\sum_j \alpha_{j,1} < \phi(x_i), \phi(x) >, \cdots, \sum_j \alpha_{j,k} < \phi(x_i), \phi(x) >) \in U$. From the eigenvalue decomposition we have $\alpha_{i,j} = (\frac{1}{\sqrt{\lambda_i}} \cdot v_i)_j$.

The steps of kernel PCA can be summarized as [STC04]:

- Center kernel Matrix $K$
- Perform Eigenvalue decomposition: $[V, \Lambda] = eig(K)$

– Calculate kernel matrix $K^P$ of the mapped data samples into the subspace: $K^P$

## 4 Distance Measures

The distance between data sets can be estimated in different ways. We can for instance measure the distance between the elements from the source data set to all samples from the target data set. This can be done in the Euclidean space or a Hilbert space as in our proposed methods. These are metric spaces with norms $||.||$ and we can calculate the distance between two elements $x \in S$ and $y \in T$ by $||x - y||$. In this case, the elements on both data sets must lie in the same metric space. An other approach to estimate the distance between data sets is based on the distance between the data distributions of the data sets. We assume that the elements of set $S$ are drawn from distribution $p_s$ and for $T$ from $p_t$. In this case, we can estimate the distance by $||p_s - p_t||_\infty$ with the supremum norm in $C(X)$ the space of continues functions. Using a universal kernels this distance can be estimated in an RKHS for the empirical distributions. In the next section, we shortly explain two such methods and how how to use them to compare distributions in an RKHS.

### 4.1 Distance between Elements in Hilbert Spaces

First, we use subspace methods to estimate the difference between the source and the target domain. As explained above, kernel PCA extracts a basis in the RKHS such that most of the variance of the data is concentrated in the subspace spanned by a small number of elements from this basis.

The expected distance for a sample from the source domain can be used to estimated whether that sample belongs to the distribution of the target domain for which we extract the principal components or not. From [STC04] for instance, we can use the bound in Equation 4 to estimate this for a data point.

$$E[\|P_U(\phi(x))|^2] \leq min_{i=1,\cdots,k} \frac{1}{l} \sum \lambda_i + 8 \cdot \sqrt{\frac{i+1}{n}} + 3 \cdot \sqrt{\frac{\ln(2 \cdot n \cdot \delta^{-1})}{2 \cdot n}} \quad (4)$$

The method of estimating the distance of the source domain to the target domain consists of the following steps. For the target domain $T$, we perform kernel PCA to approximately represent the target distribution. Using a sample $T = \{x_1, \cdots, x_n\}$ from the target domain and the corresponding kernel $k(x_i, x_j)$, kernel PCA extract a low dimensional representation of the target data $U_t$. By the distance of a sample from the source domain to this subspace, we are able to estimate how useful it might be for transfer knowledge.

Further, among a large amount source samples we are able to identify the most similar samples to our target domain. These samples are potentially most appropriated for transfer knowledge.

For the next method, we directly estimate the difference of the source and the target domain by comparing distributions mapped into an RKHS. As proposed by Gretton et al. [GBR+08] the maximum mean discrepancy (MMD) can be used to estimate the difference of two distributions $p_s$ and $p_t$. For the unit ball $F$ in an RKHS induced by a universal kernel, the MMD and its empirical estimate are defined as:

$$MMD(F, p_s, p_t) = sup_{f \in F}(\int f(x) \cdot p_t(x) \cdot dx - \int f(x) \cdot p_s(x) \cdot dx) \quad (5)$$

$$MMD(F, S, T) = sup_{f \in F}(\frac{1}{|S|} \sum_{x \in S} f(x) - \frac{1}{|T|} \sum_{x \in T} f(x)) \quad (6)$$

The MMD can be effectively calculated as norm in the RKHS of the difference of the expectation functionals $\mu[p_s] = E_{x \sim p_s}[\phi(x)]$ and $\mu[p_t] = E_{x \sim p_t}[\phi(x)]$, respectively their empirical versions.

As for the subspace method above, this measure is used to select samples from the source domain such that their empirical distribution is most similar to the target distribution.

## 5 Related Work

We distinguish two main directions in transfer learning and domain adaptation. On the hand, many of the existing approaches try to find weights for the samples that account for an mismatch in distribution of a target and a source domain. This is especially useful under the so call covariate shift assume. Here, we assume that the distribution of the labels given a sample is the same for both target and source domain. Via the weights, a sample selection bias shall be corrected. This means, we assume that the source domain is sampled from the target distribution applied a certain weighting mechanism.

Many previous approaches learn such weights such that the weighted source distributions is most similar to the target distribution.

For instance, [HSG+06] Huang et al. do this by matching the distributions in an RKHS, [KHS09] find the optimal weights by solving least squares problem and [SNK+07] minimize the Kullback-Leibler divergence of the target distributions and the weighted source distribution, to name only a few. A theoretical analysis this adaptation can be found in Cortes et al. [CMRR08] and Cortes et al. [CMM10].

In contrast to these approaches on the other hand, several other works try to extract a subspace in the data space that covers invariant parts across the target and the source distribution. Within such a subspace, transferring knowledge between the source and target domain is expected to be more effective than in the whole ambient space.

In [PTKY11], Pan et al. introduce Transfer Component Analysis to find low dimensional representations in a kernel defined Hilbert space. In this representation the target and source domain a more similar than before. Si et al., learn

in [STG10] a liner subspace that is suitable for transfer learning by minimizing Bregman divergence of the target and source distribution in this subspace. Zhang et al. [ZZW+13] propose to transfer knowledge in a Hilbert space by aligning a kernel with the target domain. Further, Muandet et al. [MBS13] learn domain invariant data transformation to minimize differences in source and target domain distributions while preserving functional relations of the data with possible label information.

These are only some of the vast amount of related work on transfer learning and domain adaptation. For a more general and deeper introducing we refer the reader to [PY10].

## 6 Transfer Learning

In our transfer learning task, we try to use information about a data set $S$ for a classification task on data from set $T$. For instance, in online reviews about products we might have reviews and information about the sentiment of the reviews about lots of electronic products. Now, the people also start reviewing books. A company might for instance broaden their offers. Now, the new reviews of books shall also be classified by their sentiment. Instead of starting from scratch and labelling all book reviews, we want to leverage the information from all the reviews about electronics that have already been classified by their sentiment. Using this information, a classifier can be learned on a transformed representation of the electronic reviews and be applied to transformed book reviews.

### 6.1 Transfer Learning via Subspaces

We assume that both data sets lie in the same Hilbert space $H$ and that their distributions have the same support. Further, we have for each element a probability distribution over a label $l$ that is the same for both data sets. This is the so called Covariate Shift assumption. This means, given an element from $H$ the probability of label $l$ depends not on the set the elements is in, but only on the element.

To transfer knowledge, we project all data onto a low dimensional subspace that captures most of the structure of the source data. This is important since otherwise we might not be able to train a good classifier or even project all data points onto a single point. In this case the distributions are the same but we can not train a classifier.

Having found a suitable subspace for transfer knowledge we project all data orthogonally onto this space. Note that an orthogonal projection onto a low dimensional subspace retracts all data points and hence makes the distributions of the two data sets already more similar. This is true since $\|P \cdot \mu_t - P \cdot \mu_s\| \leq \|P\| \cdot \|\mu_t - \mu_s\|$ and $\|P\| = 1$ for an orthogonal projection $P$ and the mean functionals $\mu_t$ of the target distribution and $\mu_s$ of the source distribution.

## 7 Greedy Selection

We propose strategies to reduce the amount of computation and storage by greedily selecting data samples from the source. This enables transfer learning strategies that use kernels to be applied on large data sets.

The proposed strategies are based in the distance of the data points to the target domain. Above, we explained how the can be calculated.

On way to estimate the distance of a sample to the target domain is to calculate the distance in the RKHS to the target domain. The target domain can be represented by the subspace $U$ extracted by kernel PCA via the first $k$ principal components. The distance from a data sample $x$ to this subspace can be simple calculated by length of the orthogonal projection $P_U$ on it, hence $\|P_U(\phi(x))\|^2$. The closer a sample is to subspace the more like it is similar the target domain and potentially helpful for transfer knowledge. This leads to the selection a samples from the source domain as described in Equation 7. We iteratively choose the sample from the source domain that has smallest distance to the subspace from the principal components of the target domain.

$$x_{t+1} = argmin_{x \in S - \{x_1, \cdots, x_t\}} \|P_{U_t}(\phi(x))\|^2 \tag{7}$$

The next selection strategy is based on kernel herding to iteratively select samples from the source domain that are most similar to the target distribution. For $\mu_t$ the expectation functional for the target domain in the kernel induced RKHS, the difference $\|\mu_t - \frac{1}{n} \sum_{x \in S' \subset S} \phi(x)\|_H^2$ estimate the difference of the target distribution and a subset of the source distributions. Chen et al. [CWS12] showed that the sampling strategy herding [Wel09] can be used to empirical and true distributions in an RKHS. Although their approach guarantees only that an optimal subset of points from a distributions can be greedily found to approximate the true distribution, we use this to approximate the target distribution by samples from the source domain. Equation 8 shows the selection strategy based on herding.

$$x_{t+1} = argmax_{x \in S - \{x_1, \cdots, x_t\}} < w_t, \phi(x) > \tag{8}$$

$$w_{t+1} = w_t + E_{p_t}[\phi(x)] - \phi(x_{t+1}) \tag{9}$$

## 8 Integration of the Greedy Selection

The proposed greed selection strategy minimizes the above loss function **??**. We after the greedy selection phase we have reduced the data size or respectively matrix size. Now we use this reduced set to define the subspace used for transfer learning. We project the data from the source domain onto this subspace and train an SVM to get a classifier $f(x)$. Next, the data from the target domain is also projected into this subspace and the classifier is applied to them.

The transfer learning process can be summarized by the following steps:

1. Perform greedy selection to get $S' \subset S$
2. Perform kernel PCA on the $K(S' \cup T, S' \cup T)$ to get projection $P_U$
3. Train SVM on $\{P_U(\phi(x_i))\}$ for all $x_i \in S'$ to get classifier $f$
4. Apply $f$ on $\{P_U(\phi(x_i))\}$ for all $x_i \in T$

This strategy directly tries to minimize the above bound in Equation 1.

The minimization of the distance of the distributions is clear since we select only samples from the source domain, that are most similar to the target domain. The second part is also straight forwards. Since, we concentrated on linear classifiers in an RKHS, we write any two hypothesis as from the source respectively the target domain as: $h_s(.) = \sum \alpha_i \cdot < \phi(x_i^s), . >$ and $h_t(.) = \sum \beta_i \cdot < \phi(x_i^t), . >$. Hence, we identify the hypothesis by weight vectors $w_s = \sum \alpha_i \cdot \phi(x_i^s)$ respectively $w_t = \sum \beta_i \cdot \phi(x_i^t)$. In the subspace after projecting all elements via $P$, the corresponding weight vectors are $w_s^P = \sum \alpha_i^{'} \cdot P \cdot \phi(x_i^s)$ respectively $w_t^P = \sum \beta_i^{'} \cdot P \cdot \phi(x_i^t)$ The distance of any of these hypothesis can be bounded in the following way:

$$\begin{aligned}
\int |w_s^P - w_t^P| p_t(x) dx &= \int |\sum \alpha_i \cdot P \cdot \phi(x_i^s) - \sum \beta_i \cdot P \cdot \phi(x_i^t)| p_t(x) dx && (10)\\
&\leq \|P\| \cdot \int |\sum \alpha_i \cdot \phi(x_i^s) - \sum \beta_i \cdot \phi(x_i^t)| p_t(x) dx && (11)\\
&= \int |\sum \alpha_i \cdot \phi(x_i^t) - \sum \beta_i \cdot \phi(x_i^t)| p_t(x) dx && (12)\\
&= \int |w_s(x) - w_t(x)| p_t(x) dx && (13)
\end{aligned}$$

Here, we use the fact that the norm of the orthogonal projection is 1, hence $\|P\| = 1$. The bound shows that the expected distance of the linear hypothesis in the subspace is less than in the whole space. The inequality cannot become an equality since we project always on lower dimensional subspace.

## 9 Experiments

We used the Amazon reviews ([BDP07]) about products from the categories books (B), DVDs (D), electronics (E) and kitchen (K). The classification task is to predict a given document as being written in a positive or negative context. The documents from one category will be used as target domain while the documents of the remaining categories are used as source domain. We use stop word removal and keep only the words that appear less than 95% and more often than 5% of the time on all documents. This results in 1993 words.

In the first experiment, we investigate how well kernel PCA can be used to extract a subspace among all data points, that is suitable for transfer learning. We simply extract the first 100 principal components from the kernel matrix $K$ for all samples from source and target domain. This means, for each $x_i, x_j \in \{T \cup S\}$ we have $K = (k(x_i, x_j)_{i,j}$. We project all data samples onto the subspace spanned by the extracted components and train a classifier for the source domain in this subspace. Next, we apply this classifier on the target domain in the

| Method | E→D | E→B | E→K | D→E | D→B | D→K | B→E | B→D | B→K | K→E | K→D | K→B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| kPCA | 75.9 | 73.9 | 81.3 | 74 | 77.7 | 75 | 71.9 | 77.5 | 72.7 | 84.4 | 79.8 | 76 |
| KMM | 68.7 | 70.7 | 81.8 | 70.7 | 74.3 | 74.1 | 68 | 71.2 | 69.6 | 83.9 | 73.5 | 74.6 |
| TCA | 64.7 | 65.2 | 80.3 | 73.7 | 69.5 | 77.2 | 73 | 69 | 73.8 | 76.7 | 67.8 | 63.7 |

**Table 1.** This table shows the accuracies on target domains using training data from different source domains, $Source \rightarrow Target$. Here, we compare covariate shift correction by Kernel Mean Matching (KMM), transfer learning by Transfer Component Analysis and projections on the principal components by kernel PCA.

| Method | E→D | E→B | E→K | D→E | D→B | D→K | B→E | B→D | B→K | K→E | K→D | K→B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| kPCA+ | 74,2 | 72.1 | 80.6 | 73.2 | 76 | 74.4 | 71.7 | 75.1 | 70.2 | 82.9 | 79 | 76.5 |
| kPCA$\mu$ | 74.9 | 68.4 | 81.2 | 70.6 | 76.2 | 72.5 | 67.5 | 76.1 | 70.6 | 82.1 | 78 | 77.3 |

**Table 2.** This table shows the accuracies on target domains using training data from different source domains. Here, we projections on the principal components by kernel PCA on samples from the source domains selected by the greedy selection strategies:Distance Based (kPCA+) and Kernel Herding Based (kPCA$\mu$).

subspace. In order to compare this simple approach we state of the art other approaches, we compare the TCA [PTKY11] and KMM [HSG$^+$06]. For TCA we use the same number of components, hence 100.

The results of the first experiment are shown in Tables 1. The projections onto the components from kernel PCA result in the best performances for most of the categories. The subspace obviously covers the important invariant parts of the data very well.

Next, we show how our greedy selection strategy performs on the same task as in the previous experiment. We use only half the amount of data from the source domains that have been selected from the greedy strategy. In contrast to the previous experiment, where we used 2000 document for the training, now we use only the 1000 documents that have been selected by our proposed greedy selection strategies.

In Table 2 we report the results of the selection strategies. The selection strategy that uses the distance to the subspace from kPCA (kPCA+) performs best on 6 of the transfer learning task. One the other hand, the herding base selection strategy is best on the remaining 6 tasks. Here, we do not see a clear favourite. The results show that our selection strategy choose good samples such that already for half the data size we get only about 2% decrease in accuracy.

Finally, we test our greedy selection strategy on the mixture of all categories as source domains for a certain category as target domain. This means, for target domain $T$ of documents from one category, we use all samples that are not in the this category as source domain. For instance, using electronic reviews as target domain, the reviews about kitchens, dvds and books are all considered as source domain. We compare our sampling strategies to random samples from the source domain. This means, we randomly draw samples from the source and use

them instead of the greedily selected ones. For the random sampling strategy, we report the mean accuracies over 20 runs.

Figure 9 shows the accuracies on the target domains when using a certain number of selected samples as source domain. The herding based sampling perform best and seems to adapt to the target distribution well. The distance based selection strategy results in slightly lower accuracy values after seeing enough data. For a small number of samples herding and even random sampling performs better than the distance based selection.

## 10 Conclusion and Future Work

We propose selection strategies on samples from a source domain that are best suited for Transfer Learning to a target domain with a different distribution. This reduces the number samples and enables kernel methods to be applied even when we have a large number of source samples. The strategies are selected to minimize the distance to the target domain. Projecting onto the subspace of the selected samples results in a subspace that is well suited to transfer knowledge from the source to the target domain.

In the future we will investigate Transfer Learning across Feature Spaces and which samples are best suited for Transfer Learning in the a certain Feature Space. In this context, we want to look at the connections to MLK and Transfer Learning using multiple sources.

## References


[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, May 2010.

[BDP07] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[CMM10] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *NIPS*, pages 442–450. Curran Associates, Inc., 2010.

[CMRR08] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, ALT '08, pages 38–53, Berlin, Heidelberg, 2008. Springer-Verlag.

[CWS12] Yutian Chen, Max Welling, and Alex J. Smola. Super-samples from kernel herding. *CoRR*, abs/1203.3472, 2012.

[GBR+08] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample problem. *CoRR*, abs/0805.2368, 2008.

[GGS13] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML (1)*, volume 28 of *JMLR Proceedings*, pages 222–230. JMLR.org, 2013.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components, 1933.

[HSG+06] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schlkopf. Correcting sample selection bias by unlabeled data. In Bernhard Schlkopf, John Platt, and Thomas Hoffman, editors, *NIPS*, pages 601–608. MIT Press, 2006.

[KHS09] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *J. Mach. Learn. Res.*, 10:1391–1445, December 2009.

[MBS13] Krikamol Muandet, David Balduzzi, and Bernhard Schlkopf. Domain generalization via invariant feature representation. *CoRR*, abs/1301.2115, 2013.

[Mer09] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, 209:415–446, 1909.

[MXZ06] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 6:2651–2667, 2006.

[PTKY11] Sinno Jialin Pan, I.W. Tsang, J.T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on*, 22(2):199–210, Feb 2011.

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.

[SGSS07] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schlkopf. A hilbert space embedding for distributions. In *In Algorithmic Learning Theory: 18th International Conference*, page 1331. Springer-Verlag, 2007.

[SNK+07] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bnau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *NIPS*. Curran Associates, Inc., 2007.

[SSM99] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Advances in kernel methods. chapter Kernel Principal Component Analysis, pages 327–352. MIT Press, Cambridge, MA, USA, 1999.

[STC04] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[Ste05] I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Trans. Inf. Theor.*, 51(1):128–142, January 2005.

[STG10] Si Si, Dacheng Tao, and Bo Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942, 2010.

[VB02] Pascal Vincent and Yoshua Bengio. Kernel matching pursuit. *Mach. Learn.*, 48(1-3):165–187, September 2002.

[Wel09] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1121–1128, New York, NY, USA, 2009. ACM.

[WS01] Christopher Williams and Matthias Seeger. Using the nystrm method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[ZZW+13] Kai Zhang, Vincent Zheng, Qiaojun Wang, James Kwok, Qiang Yang, and Ivan Marsic. Covariate shift in hilbert space: A solution via sorrogate kernels. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 388–395. JMLR Workshop and Conference Proceedings, May 2013.

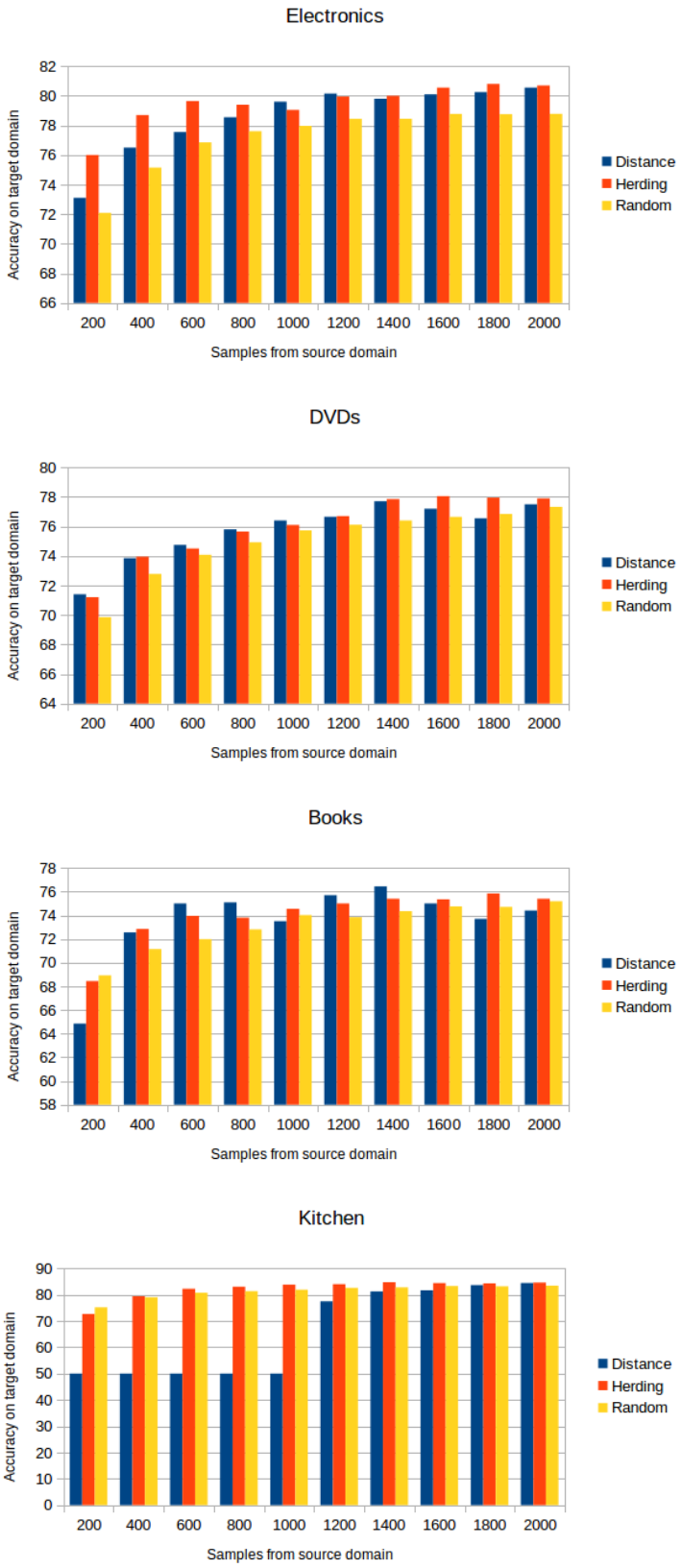


**Fig. 1.** Results on the target data domain for the different categories. We compare random samples with our greedy selection strategy for sampling.