Sampling Methods for Random Subspace Domain Adaptation

Christian Pölitz

TU Dortmund University, Otto Hahn Str. 12, 44227 Dortmund

Abstract. Supervised classification tasks like Sentiment Analysis or text classification need labelled training data. These labels can be difficult to obtain, especially for complicated and ambiguous data like texts. Instead of labelling new data, domain adaptation tries to reuse already labelled data from related tasks as training data. We propose a greedy selection strategy to identify a small subset of data samples that are most suited for domain adaptation. Using these samples the adaptation is done on a subspace in a kernel defined feature space. To make this kernel approach applicable for large scale data sets, we use random Fourier features to approximate kernels by expectations.

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 that the model is applied. On the other hand, there are many situation where this is not true. Imagine as Data Mining task Sentiment Analysis on product reviews from Amazon. In case of a new product or product type, producers might be interested in how their products catches on. Sentiment Analysis now tries to label the reviews of the corresponding products as being positive or negative. To assign such labels, classification models are trained on some labelled reviews and than applied on the unlabelled reviews. For new product types it is reasonable to assume that we have no labelled training data at hand. Labelling the new reviews can be quite expensive. Especially identifying the sentiment in texts can be hard - even for experts. Ambiguous words or sarcasm for instance make this task difficult. Instead of starting to label the new reviews, another possibility is to reuse already labelled reviews from different products. There might be for instance already labelled reviews about books and now we get new reviews about DVDs. The idea is to leverage the reviews about books to train a classifier that is applied on reviews about DVDs. To accomplish this, we need to find a way to safely transfer the information from one domain to another.

We solve this problem by domain adaptation with the following assumptions: We have two data sets with (possible large) difference in distribution. 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 domain adaptation task now is to use the source domain together with its label information to find a classifier that labels the target domain best.

We expect that many data sets share similarities on latent subspaces. On product reviews for instance, a book might be described as *tedious* while a toaster might be described as *malfunctioning*. Both words have negative meaning and very likely appear together with other negative words like *bad*, *poor* or *poorly*. In a latent subspace in the space spanned by the words, we expect that these words span together a whole dimension. When we map texts of reviews from books and electronic articles onto such a subspace the words *tedious* and *malfunctioning* can be replaced by their common meaning. This will make the texts from the different domains more similar. Further, only terms alone might not be able to find such subspaces. For instance, bi-grams like *little helpful* or *hardly improving* can also span a latent subspace that is helpful for domain adaptation. Generally, n-grams should also be considered.

In order to integrate information of multiple combinations of words, kernels like polynomial kernels can be used. Kernel methods can also integrate structural information and even information from probabilistic models. Consequently, we find low dimensional representations of the data from a source and a target domain in a Reproducing Kernel Hilbert Space. These representations shall keep enough structure from the data that a classifier trained on the source domain still performs well. On the other hand, the low dimensional representation shall make the two data sets more similar. This justifies a safe application of a classifier trained on the source domain, to the target domain.

To find the subspace for the domain adaptation we propose a greedy selection strategy that finds the most useful data samples in the source domain for the domain adaptation. By this, we reduce the data size and concentrate on those samples that are potentially best suited to 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 in [10]. Further, we approximate kernels by random Fourier features as proposed by [19]. This tackles the quadratically or cubically scaling behaviour of kernel methods in the number of data samples.

Related Work

We distinguish two main directions in 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 [8] propose density estimators that incorporate sample selection bias to adapt two distribution, [13] do this by matching the distributions in an RKHS, [14] find the optimal weights by solving least squares problem and [23] minimize the Kullback-Leibler divergence of the target distributions and the weighted source distribution, to name only a few. A theoretical analysis of this adaptation can be found in [6] and [5].

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

In [18], Transfer Component Analysis is introduced to find low dimensional representations in a kernel defined Hilbert space. In this representation the target and source domain are more similar than before. The authors in [22] learn a linear subspace that is suitable for transfer learning by minimizing Bregman divergence of the target and source distribution in this subspace, [21] transform the target points such that they are a linear combination of a basis in the source domain, [25] propose to transfer knowledge in a Hilbert space by aligning a kernel with the target domain, [17] 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. Further, in [9] the authors propose to create subspaces that aligns to the eigenspaces of the target and source domain.

Background

In this section, we introduce the background on kernel methods, subspaces in Hilbert Spaces and distances of distributions. The presented information are crucial for our proposed strategy in the next sections.

Kernel Methods and RKHS

Kernel methods accomplish to apply linear methods on non-linear representations of data. Any kernel method uses a map $X \rightarrow \phi(X)$ 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 $f(y) = \phi(x)(y) = \langle \phi(x), \phi(y) \rangle$. $\phi(x)$ is a function and $\phi(x)(y)$ means the function value at y.

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 into 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 $v \in H'$. This is important since we only need to consider kernel evaluations and not infinite dimensional elements of the RKHS. Kernel PCA for instance can be used to find an appropriated projection matrix onto such a subspace. See [20] for further details.

Distance Measures

As proposed by [12] the maximum mean discrepancy (MMD) can be used to estimate the difference of two distributions p_s and p_t. For the unit ball H in an RKHS induced by a universal kernel k, the MMD and its empirical estimate are defined as:

$$MMD(H, p_s, p_t)^2 = \|\mu[p_s] - \mu[p_t]\|_H^2$$

respectively

$$MMD(H, S, T)^2 = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} k(x_i, x_j) - \frac{1}{|S||T|} \sum_{x_i \in S, x_j \in T} k(x_i, x_j) + \frac{1}{|T|^2} \sum_{x_i, x_j \in T} k(x_i, x_j). \tag{1}$$

Random Features

To avoid large computational and storage complexity of kernel methods, approximations of the kernel can be used. Random features for instance approximate the feature maps in Hilbert spaces by low dimensional random projections. The expectation of the inner products of these random features evaluate to corresponding kernel values. Any shift-invariant kernel (as for example the Gaussian kernel) can be represented as expectation of random features $cos(\omega x + b)$ for an appropriate distribution $p(\omega)$ and b uniformly drawn from $[0, 2\pi]$, see [19]. For Gaussian kernels, ω is drawn from the distribution: $p(\omega) = (2\pi)^{-k/2} e^{-\|\omega\|^2/2}$. An unbiased estimate of the expectation is $z_\omega(x_i)' z_\omega(x_j)$ for $z_\omega(x) = \frac{\sqrt{2}}{k}[cos(\omega_1 x), \cdots cos(\omega_k x), sin(\omega_1 x), \cdots sin(\omega_k x)]$.

The deviation of the inner product of the random features of dimension k to the true kernel value is bounded by a tail bound using Hoeffding's inequality. Since $z_\omega \in [-\sqrt{2}, \sqrt{2}]$, we have $z_\omega(x_i)' z_\omega(x_j) \in [-2, 2]$. This and $E_\omega[z_\omega(x_i)' z_\omega(x_j)] = k(x_i, x_j)$ justifies the following bound:

$$P(|z_\omega(x_i)' z_\omega(x_j) - k(x_i, x_j)| \geq \epsilon) \leq 2e^{-k\epsilon^2/8}$$

Domain Adaptation

In a domain adaptation 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.

Domain Adaptation via Subspaces

We assume that both data sets lie in the same Hilbert space H by using the same kernel 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 the structure of the source data and the target 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 good classifier.

The simplest way to find a projection onto a subspaces that captures most of the structure is using kernel PCA. We have two data sets that should not loose too much of its structure after projection. The structure of the source domain must be kept to train a good classifier, but the target domain is the actual data we are interested in. Further, we expect that not all information from the source is useful. The idea is now to keep the structure of the target data completely, but for the source data only those parts such that the source and target distributions are close on the subspace that covers only this structure.

Having found a suitable subspace for domain adaptation we project all data orthogonally onto this space. An orthogonal projection onto a low dimensional subspace retracts all data points and makes the distributions of the two data sets 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 functional μ_t of the target distribution and μ_s of the source distribution.

Further, the expected distance between classification models on source and target domain decreases. Since we concentrated on linear classifiers in an RKHS, we write any classifier from the source, respectively the target domain as: $h_s(.) = \sum \alpha_i \cdot \langle \phi(x_i^s), . \rangle$ and $h_t(.) = \sum \beta_j \cdot \langle \phi(x_j^t), . \rangle$. Hence, we identify the classifier by weight vectors $w_s = \sum \alpha_i \cdot \phi(x_i^s)$ respectively $w_t = \sum \beta_j \cdot \phi(x_j^t)$. After projecting all elements onto the subspace via P, the corresponding weight vectors are $w_s^P = \sum \alpha_i \cdot P \cdot \phi(x_i^s)$ respectively $w_t^P = \sum \beta_j \cdot P \cdot \phi(x_j^t)$ The distance of any of these classifiers can be bounded in the following way:

$$
\begin{aligned}
& \int |w_s^P(x) - w_t^P(x)| p_t(x) dx \qquad (2) \\
= & \int |\sum \alpha_i \cdot P \cdot \phi(x_i^s) - \sum \beta_j \cdot P \cdot \phi(x_j^t)| p_t(x) dx \\
\leq & \|P\| \cdot \int |\sum \alpha_i \cdot \phi(x_i^s) - \sum \beta_j \cdot \phi(x_j^t)| p_t(x) dx \\
= & \int |\sum \alpha_i \cdot \phi(x_i^t) - \sum \beta_j \cdot \phi(x_j^t)| p_t(x) dx \\
= & \int |w_s(x) - w_t(x)| p_t(x) dx
\end{aligned}
$$

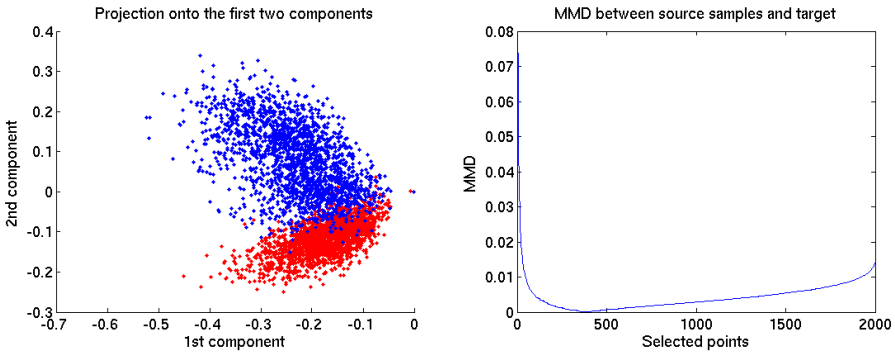

Fig. 1. Illustration of the samplings. Left: Source (electronic reviews in red) and target (DVD reviews in blue) data plotted in the space of the first two components of both of them together. Right: MMD of the selected samples from the source data by Herding based sampling.

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 classifiers in the subspace is less than in the original Hilbert space. The inequality cannot become an equality since we project always on lower dimensional subspace. This shows that these projections decrease the expected error on the target domain of any classifier trained on source domain with a different distribution, see (CF-[2]).

Greedy Selection

To find the most promising data points from the source domain for the domain adaptation, we propose a greedy strategy to efficiently select them. The sampled data points shall be close to the target domain to prevent too much influence of the source domain. On the other hand, the samples must keep enough structure of the source domain such that a good classifier can be trained on the source domain data. The proposed strategy is based on the distance of the the source domain distribution to the target domain distribution. The picture in Figure 1 illustrates our idea on electronic (red) and DVD (blue) reviews. We assume the reviews of electronics as target domain and the reviews about DVDs as source domain. The reviews seem to be more similar on one direct than on the other. The idea now is to prefer points from the source domain that are more prominent in this direction for the domain adaptation.

Distribution Based Sampling We propose a sampling strategy that is based on the data distribution. In the Hilbert space we iteratively select mapped samples from the source domain that are most similar to the target distribution. For μ_{p_t} the expectation functional for the target domain in an RKHS, the difference $\|\mu_{p_t} - \frac{1}{n}\sum_{x \in S' \subset S} \phi(x)\|_H^2$ estimates the difference of the target distribution and a subset of samples from the source distribution. Similar approaches are proposed by [4], The authors showed that the sampling strategy introduced by [24] can be used to match empirical and true distributions in an

RKHS. Equation 3 shows the selection strategy based on matching distributions in an RKHS.

$$x_{t+1} = argmax_{x \in S - \{x_1, \cdots, x_t\}} \langle w_t, \phi(x) \rangle \tag{3}$$
$$w_{t+1} = w_t + E_{p_t}[\phi(x)] - \phi(x_{t+1})$$

For deciding when to stop the sampling, we monitor $\max_{x \in S - \{x_1, \cdots, x_t\}} \langle w_t, \phi(x) \rangle$. As soon as we have only data points from the source data set left that make the distance in distribution no longer decreasing, we stop. By this, we sample only those points such that the empirical distributions of samples and the target data are minimal. The picture on the right of Figure 1 shows an example of the course of the MMD of the samples from the source domain (electronic reviews) and the target domain (DVD reviews). We sample as long as the MMD decreases to find all points that make the distribution similar. This bewares us to sample points that make the two distribution dissimilar.

Analysis of Distribution Based Sampling For $\mu_{p_t} = \frac{1}{n_t} \sum_{x_i \in T} \phi(x_i)$, our sampling strategy minimizes:

$$E = \| \mu_{p_t} - \frac{1}{T} \sum_{x_j \in S'} \phi(x_j) \|_H^2.$$

To see this we rewrite

$$E = \langle \mu_{p_t}, \mu_{p_t} \rangle - \frac{2}{T} \sum_{x_j \in S'} \langle \mu_{p_t}, \phi(x_j) \rangle + \frac{1}{T^2} \sum_{x_i, x_j \in S'} \langle \phi(x_i), \phi(x_j) \rangle.$$

Since $\langle \mu_{p_t}, \mu_{p_t} \rangle$ is constant, minimizing E is the same as maximizing

$$\frac{2}{T} \sum_{x_j \in S'} \langle \mu_{p_t}, \phi(x_j) \rangle - \frac{1}{T^2} \sum_{x_i, x_j \in S'} \langle \phi(x_i), \phi(x_j) \rangle.$$

Multiplying the last expression by T results in the greedy sampling as defined above when we set $w_0 = \mu_{p_t}$. This means the strategy matches the empirical distribution of the target samples with the empirical distribution of the subset of the samples from the source distribution.

Random Feature Sampling Our proposed sampling strategy can still result in a large number of points from the source distribution. We further propose to combine the selection strategy and the domain adaptation on a subspace by random features of dimension k. This enables us to perform the domain adaptation task in the linear space spanned by the random Fourier bases of the random features as defined above.

We define MMD_ω similar as MMD in Equation 2 except that the kernel evaluations are replaces by the inner products of the random features. Since $MMD_\omega \in [-8, 8]$, we can apply Hoeffding's inequality to bound the difference to the true MMD by:

$$P(|MMD_\omega^2 - MMD^2| \leq \epsilon) \leq 2e^{-k\epsilon^2/128}.$$

Due to linearity of the expectation we have: $E_\omega {MMD_\omega}^2 = MMD^2$ and from the definition of the random features we have: $k(x_i, x_j) = E_\omega[z_\omega(x_i)' z_\omega(x_j)]$. All together results in the bound.

Further, we need to estimate how much the components for the random features deviate from the true components the source samples in the RKHS. For this it suffices to investigate the expected difference of the true kernel matrix K for n data points and the matrix of the inner products of the random features K_ω. An appropriate bound is proposed by [16]:

$$E[\|K_\omega - K\|] \leq \sqrt{\frac{2n^2 \log n}{k}} + \sqrt{\frac{2n \log n}{k}}.$$

Experiments

We test our proposed method to find projections onto subspaces for domain adaptation on three standard benchmark data sets that have been used in previous domain adaptation experiments.

As first data set, we use the Amazon reviews [3] 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. 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. The reviews of a certain product will be used as target domain and all the others as source domain.

The second data set is the Reuters-21578 [15] data set. It contains texts about categories like organizations, people and places. For each two of these categories a classification task is set up to distinguish texts by category. Each category is further split into subcategories and different subcategories are used as source and target domains. The exact configuration of the tasks is given by [7].

The third data set is the 20 Newsgroup data set[1]. We use the four top-categories (comp,rec,sci and talk) in the same configuration and splits as in [1]. For each two of these top-categories a classification task is set up to distinguish texts by category. Each category is further split into subcategories and different subcategories are used as source and target domains.

For the subspace for domain adaptation, we simply extract the first 100 principle components from the kernel matrix K for all samples from the sampled source domain data and the 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 (all source and training data) onto the subspace spanned by the extracted components and train a classifier on the source domain in this subspace. Next, we apply this classifier on the target domain in the

[1] http://qwone.com/ jason/20Newsgroups/

Method	org vs. places	places vs. org	places vs. people	people vs. places	comp vs. rec	comp vs. sci	comp vs. talk	rec vs. sci	rec vs. talk	sci vs. talk
KMM	60.1	56.8	58.5	56.2	96.9	84.4	98.5	91.2	98.5	95.4
TCA	85.4	80.5	76.5	76.5	94.5	87.8	96.2	90.2	94.1	88.9
GFK	72.9	66.1	68.7	66.4	84.1	74.7	91.9	72.5	86.6	79.02
Sampling	**90**	82	83.5	**79.2**	**99.1**	**92**	**99.2**	**98.3**	**99**	**96.2**
Sampling+RF	84.7	**82.9**	**85.5**	77.3	98	88.4	98.7	91.7	98	93.7

Table 1. Accuracies on the Reuters and 20 news groups data sets. We compare our proposed greedy sampling methods (without and with random features) and projection with Kernel Mean Matching (KMM) and Transfer Component Analysis (TCA), Gradient Flow Kernel (GFK).

Method	$\{D \cup B \cup K\} \rightarrow$E	$\{E \cup B \cup K\} \rightarrow$D	$\{E \cup D \cup K\} \rightarrow$B	$\{E \cup D \cup B\} \rightarrow$K
KMM	81.0	75.2	72.5	83.9
TCA	81.4	77.8	74.7	84.9
GFK	68.7	66.3	62.2	70.7
Sampling	**82.4**	79.15	77.25	**85.25**
Sample+RF	81.3	**79.7**	**77.65**	84.85

Table 2. Accuracies on Amazon reviews using one product as target domains and all the other domains as source domain. We compare our proposed greedy sampling methods (without and with random features) and projection with Kernel Mean Matching (KMM) and Transfer Component Analysis (TCA), Gradient Flow Kernel (GFK) and the Landmark method (LM) with projection for domain adaptation.

subspace. We compare the sampling strategies without and with random features (Sampling, Sampling+RF) with Transfer Component Analyses (TCA) [18], Kernel Mean Matching (KMM) [13] and Gradient Flow Kernel (GFK) [11]. For TCA we also use 100 components. We use Gaussian kernels with optimized width parameter σ. For the classification we train an SVM with optimized error weight C. For the random features, the results are mean values over 10 runs with random features of dimension 10.000.

The method by [10] has the same objective as our sampling methods. They find those source domain points that minimizes the MMD to the target domain. Compared to our method, the points are extracted by solving a quadratic optimization problem with constraints. This is computationally challenging when we have large source domains. Further, they do not directly select the points, they propose to learn weights of the points and remove those points that have weights below a threshold. This threshold has to be chosen by hand. In the experiments we use the same threshold as they have done in their experiments.

The results of the first experiment are shown in Tables 1 and 2. The projections onto the components result in the best performances for all the domains. The subspace obviously covers the important invariant parts of the data very well. Using random features to approximate the kernel values results in the second best accuracies compared to the other methods.

We now explore how many source domain points have been chosen from which domain.

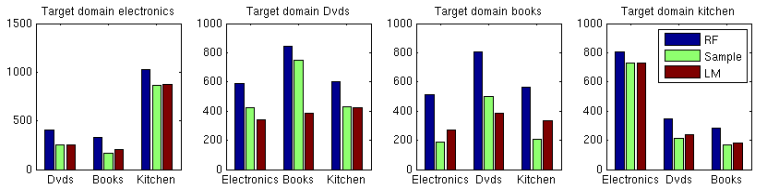

Fig. 2. Histograms of the selected points from Amazon reviews.

Figure 2 shows histograms of the selected data points from the source domain for the different methods. The sampling strategy without and with random features and the GFK method uses a similar amount of samples from the source domains. The histograms show that for each target domain the methods have always one domain in the mixture of source domain where most of the samples are drawn from. For sampling there is always on clear domain from which the method samples most from.

To investigate this further we calculate the Maximum Mean Discrepancy as defined in Equation 2 to estimate the difference of the distributions of the target and source domains. Table 3 shows the MMD values using reviews from the domains. For the electronics reviews (E), the reviews about kitchens (K) are closest in distributions. Comparing this result with the accuracies from above, on the target domain with reviews about electronics, source domain kitchen performs best for domain adaptation. Similar results can be seen for the other domains. Comparing the MMD of the domains with the sampled points from the last experiments, we see that the sampling method chooses the source domain points that results in low MMD best.

MMD	E	D	B	K
E	0	0.0177	0.0207	0.0067
D	0.0177	0	0.0174	0.0173
B	0.0207	0.0174	0	0.0200
K	0.0067	0.0173	0.0200	0

Table 3. Maximum Mean Discrepancy (MMD) measure on the different domains from the categories from the Amazon reviews.

Finally, we investigate the influence of the random features on the quality of the domain adaptation. We perform several runs using different feature sizes.

The plots in Figure 3 show a fast convergence already after some thousand random features. Experiments with random features of dimension less than one thousand has let to poor performance. This might be due to the slower convergence of the kernel matrix to the matrix of the inner products of the random features in the norm. In the future we will investigate this further.

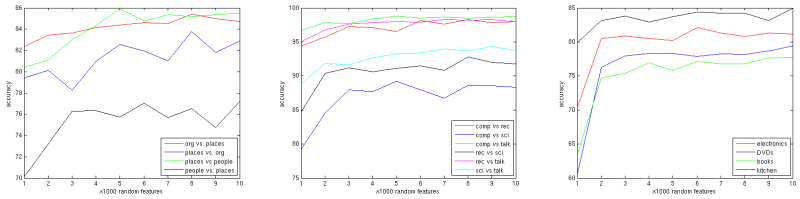

Fig. 3. The classification accuracies using different numbers of random features. Left: the Reuters data set; middle: 20 news groups data set; right: Amazon reviews

Conclusion and Future Work

We proposed a selection strategy on samples from a source domain that are best suited for domain adaptation to a target domain with a different data distribution. The samples are selected to keep the structure of the target domain points while adding some structure from the source domain points. Projecting onto the subspace of the selected samples and the target samples results in a subspace that is well suited for domain adaptation from the source to the target domain. To apply this approach also on large scale data sets, we use random features to approximate kernel values. On benchmark data sets, we showed that our methods perform well on domain adaptation tasks. In the future we want to investigate domain adaptation across different feature spaces. In this context, we want to look at the connections to MKL and domain adaptation using multiple sources.

References

1. Yang Bao, Nigel Collier, and Anindya Datta. A partially supervised cross-collection topic model for cross-domain text classification. In *Proceedings of the 22Nd ACM International Conference on Conference on Information &38; Knowledge Management*, CIKM '13, pages 239–248, New York, NY, USA, 2013. ACM.
2. 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.
3. 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.
4. Yutian Chen, Max Welling, and Alex J. Smola. Super-samples from kernel herding. *CoRR*, abs/1203.3472, 2012.
5. 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.
6. 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.

7. Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 210–219, New York, NY, USA, 2007. ACM.
8. Miroslav Dudík, Robert E. Schapire, and Steven J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *NIPS*, 2005.
9. Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Subspace alignment for domain adaptation. *CoRR*, abs/1409.5241, 2014.
10. 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.
11. Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073. IEEE, 2012.
12. 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.
13. 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.
14. Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *J. Mach. Learn. Res.*, 10:1391–1445, December 2009.
15. David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004.
16. D. Lopez-Paz, S. Sra, A. Smola, Z. Ghahramani, and B. Schölkopf. Randomized nonlinear component analysis. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning, W and CP 32 (1)*, pages 1359–1367. JMLR, 2014.
17. Krikamol Muandet, David Balduzzi, and Bernhard Schlkopf. Domain generalization via invariant feature representation. *CoRR*, abs/1301.2115, 2013.
18. 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.
19. Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. 2007.
20. 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.
21. Ming Shao, Carlos Castillo, Zhenghong Gu, and Yun Fu. Low-rank transfer subspace learning. *Data Mining, IEEE International Conference on*, 0:1104–1109, 2012.
22. 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.
23. 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.
24. 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.
25. 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.