Noname manuscript No.
(will be inserted by the editor)

Interpretable Domain Adaptation via Optimization over the Stiefel Manifold

Christian Pölitz[1] · **Wouter Duivesteijn**[2] · **Katharina Morik**[1]

Abstract In domain adaptation, the goal is to find common ground between two, potentially differently distributed, data sets. By finding common concepts present in two sets of words pertaining to different domains, one could leverage the performance of a classifier for one domain for use on the other domain. We propose a solution to the domain adaptation task, by efficiently solving an optimization problem through Stochastic Gradient Descent. We provide update rules that allow us to run Stochastic Gradient Descent directly on a matrix manifold: the steps compel the solution to stay on the Stiefel manifold. This manifold encompasses projection matrices of word vectors onto low-dimensional latent feature representations, which allows us to interpret the results: the rotation magnitude of the word vector projection for a given word corresponds to the importance of that word towards making the adaptation. Beyond this interpretability benefit, experiments show that the Stiefel manifold method performs better than state-of-the-art methods.

Keywords: domain adaptation, stochastic gradient descent, matrix manifolds.

1 Introduction

Text classification is an important data mining task with many applications. E.g., sentiment analysis assesses texts to be written positively or negatively. This information can help companies to find out how well their products catch on. When we want to solve such text classification tasks via supervised learning, we need labelled training data. Such data can be quite hard to get; in sentiment analysis the identification of a positive tone can be ambiguous, and sarcasm or individual writing styles can make the labelling difficult even for experts with a linguistic background. To forego labelling an unlabelled data set, we can reuse other data sets that have already been labelled for a similar task: review texts for electronic appliances with labels about their sentiment could be used to train a classifier for unlabelled review texts about DVDs. When the dissimilarity increases between texts from the labelled data set and the texts we want to classify, expected performance decreases. Ben-David et al. (2010) showed that the expected error on a data set A of a classifier trained on a data set B correlates positively with the distributional difference between the data sets. The task to find common distributional ground between data sets, with the goal of training a classifier on one data set and applying it on another, is called *domain adaptation*.

[1] TU Dortmund, Computer Science, LS 8, 44221 Dortmund, Germany,
[2] Universiteit Gent, Data Science Lab & iMinds, 9000 Gent, Belgium

One approach to domain adaptation is to find a low-dimensional latent feature representation on which the two data sets of text documents are more similar in distribution. We expect that many data sets share similarities on latent feature representations. For instance, a book might be described as tedious while a toaster might be described as malfunctioning. Both words have a negative connotation and very likely appear together with other negative words like bad, poor or poorly. Projecting the reviews results in a low-dimensional latent feature representation, in which we expect these words to jointly span a dimension representing their common ground. These latent features represent the common concepts (e.g., sentiments) between different words from different domains, and can be expected to contain less noise. We propose to find a latent feature representation in the space spanned by word vectors. This is done by a linear projection that optimally matches text documents from one domain to another domain with different data distributions. The projection is performed on the word vectors of the documents from the different domains and maps into a low-dimensional latent feature representation. The goal is to make the training and the test data more similar in the new feature representation, in order to safely apply a classifier on the test data that is trained on the differently distributed training data. We concentrate on latent features that are linear projections of the original data, for two reasons: linear approaches can be better interpreted in terms of the transformation of individual features (or individual words!) required to make the two data sets similar in distribution, and linear methods scale better than non-linear projections via kernels, as pointed out for instance by Pan et al. (2009). The main disadvantage of these non-linear kernel approaches is that they scale quadratically or even cubically in the number of examples, and new data points must be projected via kernel evaluation of up to all other data points. Linear projections are more efficient since the word vectors from texts are usually sparse, allowing linear maps via a projection matrix to be efficiently implemented with sparse matrix operations.

To find an optimal projection, we propose a matrix-variate optimization that minimizes the distance in distribution between the training and the test data. The optimal matrix is the projection matrix mapping all training and test data into a low-dimensional feature representation with minimal distributional difference between projected training and test data. To solve this optimization problem, we employ Stochastic Gradient Descent (SGD), which allows for larger data sets. This is important, since review text collections are usually large: for Sentiment Analysis, more than 34.000.000 Amazon reviews are available (McAuley and Leskovec 2013). Closed-form solutions or other optimization methods like plain Gradient Descent would be prohibitively expensive when using all data. Solving SGD without constraints on the matrix will easily end in rank-deficient matrices that map the data onto too low-dimensional representations. Hence, we add the constraint to the optimization problem that the matrices must be projection matrices: they must contain only orthogonal columns. This constraint makes the optimization more difficult, which is traditionally resolved by projecting the matrices onto the set of orthogonal matrices. The induced additional error is amplified by SGD, since we perform many optimization steps. To avoid poor convergence, we perform the optimization directly on the matrix manifold $M(p, q)$, encompassing projection matrices from a p-dimensional Euclidean space into q-dimensional linear feature representations. Thus, we remove the constraint and the need of projecting the matrices during the SGD steps onto the set of orthogonal matrices.

1.1 Main Contributions

This paper provides update rules, enabling the running on large document collections of Stochastic Gradient Descent (SGD) directly on the Stiefel manifold. The optimization problem encompassing the SGD steps efficiently identifies projections into low-dimensional latent feature representations for domain adaptation. The resulting projection matrices are interpretable: the rotation magnitude of the word vector projection for a given word into a latent feature dimension represents the contribution of that word towards the underlying concept represented by that latent feature. Therefore, the rotation corresponds to the importance of the word for the domain adaptation. This interpretability of the solutions for the domain adaptation task provided by the Stiefel method is the main contribution of this paper. Collateral benefit is that the Stiefel method delivers high-accuracy results in comparison with state-of-the-art methods.

In contrast to previous approaches like TCA (Pan et al. 2009) and JCA (Long et al. 2013), we propose an optimization that extracts interpretable linear factors based on the Bag-of-Words representation of documents. Echoing the previous approaches, we match the distributions of the documents based on Maximum Mean Discrepancy. This measure estimates the discrepancy of the two data sets based on all moments estimated from the data. This makes the problem harder, since it is no longer convex. We have no closed-form solution, and must resort to gradient-based approaches. The reason to apply SGD is twofold. First, we make our approach applicable to large-scale scenarios. For large text collections, we resort to an online solution. Second, since our problem is non-convex and high-dimensional, we will easily end up with local optima during the optimization. SGD, in contrast to plain Gradient Descent (GD), adds randomness into the optimization that is gradually reduced in the course of the optimization. This allows to skip local minima in the beginning.

2 Related Work

Before turning to the question of transferring knowledge from one domain to another, we need to discuss how to measure the distributional difference between different data sets from different domains. In the context of domain adaptation, divergence measures like KL-divergence (Sugiyama et al. 2008) or A-distance (Ben-David et al. 2006) have been used. We use the kernelized Maximum Mean Discrepancy (MMD) as proposed by Gretton et al. (2008) for an estimation of the difference in distribution between two data domains using samples. We do so, since this method is able to compare distributions by using all moments of the distributions. This choice is not pivotal to the contributions of this paper; it's merely a parameter that can be changed at will.

A large part of the research on domain adaptation concentrates on estimating weights for the target domain: data from one domain will be weighted to increase distributional similarity to data from another domain. Under the so-called sample selection bias, the target domain can be made similar to a source domain by adapted weighted sampling. For instance, Dudík et al. (2005) propose density estimators that incorporate sample selection bias to adapt different test domains to training domains. In (Bickel et al. 2009), the distance between the data from the

two domains is directly minimized to find the optimal weights. Huang et al. (2007) propose to learn weights for a target domain such that the distance in distribution of the weighted target domain to a source domain is minimized, using Kernel Mean Matching as distance measure between the domains and performing the search for optimal weights in a universal Reproducing Kernel Hilbert Space. By contrast, Sugiyama et al. (2007) find the optimal weights via matching distributions by minimizing the KL-divergence.

Subspace-based domain adaptation strives to increase similarity, not by adapting distributions, but by transform their support. This results in a low-dimensional feature representation of the original data. The transformation is done by a projection onto an appropriate subspace. Si et al. (2010) propose to minimize the Bregman divergence for regularized subspace learning. Via a matrix-variate optimization problem they find an optimal subspace for a given cost function. On this subspace, two given data sets are gauged to be similar with respect to a divergence criterion. Contrary to the Stiefel approach that we propose in Section 4, this optimization is directly done in $\Re^n$. In (Shao et al. 2012), a low-dimensional subspace is extracted such that the data from a target domain can be expressed as linear combination of a basis from a source domain. The authors solve this problem by inexact Augmented Lagrangian Multipliers, which is computationally expensive, especially since it demands several Singular Value Decompositions (SVDs) on the data matrix. Ni et al. (2013) propose to find a sequence of subspaces in which the data from the target domain can be expressed as linear combination of a source domain. For domain adaptation they project all data onto each subspace and concatenate all resulting feature representations. This approach also needs to perform several expensive SVDs on the data matrix. In (Chen et al. 2009) and (Chattopadhyay et al. 2012), domain adaptation is coupled with the training of a classifier. Chen et al. (2009) do this by inverting the whole data matrix, which can be quite expensive. The approach in (Chattopadhyay et al. 2012) needs additional labels for the target domain, and a kernel matrix which might become prohibitively expensive to use.

As an alternative to the subspaces in $\Re^n$ of the word vectors, Kernel-based methods have been proposed to find non-linear data representations for domain adaptation. Pan et al. (2008) introduce a transfer learning by feature transformation that optimizes the MMD. In Pan et al. (2011), Transfer Component Analysis finds low-dimensional representations in a kernel-defined Hilbert space to make two given data domains more similar. Long et al. (2013) extend this approach by including class label information. Zhang et al. (2013) propose to transfer knowledge in a Hilbert space by aligning a kernel with the target domain. Muandet et al. (2013) propose to learn domain invariant data transformation to minimize differences in source and target domain distributions while preserving functional relations of the data.

2.1 Related Manifold Methods

We use optimization directly on matrix manifolds. A general introduction can be found in (Absil et al. 2008). An early work on such optimization is (Edelman et al. 1999). The authors develop a gradient-based optimization method on Grassmann and Stiefel manifolds. They provide a general framework for the optimization on

Table 1 Notation employed throughout the paper

Symbol	Description
S	Source domain data
T	Target domain data
p_S	Distribution of the source domain
p_T	Distribution of the target domain
P	Projection matrix
M	Stiefel manifold
Exp	Exponential map
Z	Set of matrices $z_i = [x_i, y_i]$ with $x_i \sim p_S, y_i \sim p_T$

these matrix manifolds. Both Balzano et al. (2010) and Bonnabel (2013) describe a stochastic gradient descent on Riemann manifolds and illustrate its use for subspace tracking and optimization on matrices with rank constraints.

Gong et al. (2012) and Gong et al. (2013) perform domain adaptation on manifolds. They project the data onto all subspaces that lie on the shortest path (geodesic) between two subspaces from, respectively, the source and target domain. They define a kernel on the concatenation of all projections to extract a new feature representation. Gopalan et al. (2011) sample interpolated subspaces on the Grassmann manifold between a target and a source subspace, extracting domain, intermediate, and possibly invariant information. Projections onto subspace samples transform the data into new feature representations. Gopalan et al. (2011) sample these subspaces, and use projections onto these samples to transform the data into new feature representations. Baktashmotlagh et al. (2013) perform gradient descent on a Grassmann manifold to find a subspace where the two given data domains have a low distance. In Cheng and Pan (2014), the authors propose semi-supervised learning for domain adaptation on manifolds.

3 Preliminaries

The classic assumption for a supervised classification task is that training and test data come from the same distribution. By contrast, domain adaptation methods adapt data from a *source domain* S with a certain distribution p_S to a *target domain* T with a different distribution p_T. For the source domain we have additional information like labels to train a classifier. For the target domain we have no additional information. The task is to extract information from the source domain that is also relevant for classification on the target domain. In this paper, each domain is represented by a set of word vectors of the corresponding documents, and each word vector contains frequency information of the words in the document. The notation in Table 1 will be used throughout the paper; we will write P for a projection matrix, M for the Stiefel manifold, Exp for the exponential map, and Z for a set of matrices with columns: $z_i = [x_i, y_i]$, with $x_i \sim p_S, y_i \sim p_T$. These terms will be further explained when their time is due.

3.1 Matrix Manifolds

For two given data sets from a source domain and a target domain we want to find an optimal projection matrix onto a low-dimensional feature representation. The optimal projection projects onto a representation in which the distribution of the projected data points from the source domain is the most similar to the distribution of the projected data points from the target domain. Within this representation, a discriminative classifier is trained on the source domain. Since the distributions are similar on this representation, we can expect that this classifier can be safely applied to the projected data points from the target domain. Such projections have been successfully used in text mining and Natural Language Processing. For example, in text classification, latent semantic analysis (LSA) has proven to be quite successful to approximate documents by low-dimensional concept vectors. See for instance (Deerwester et al. 1990) for an introduction. This motivates the hypothesis that such low-dimensional representations in the vector space of the documents might be beneficial for transferring knowledge from one domain of documents to another.

A latent subspace L in a vector space V is identified by a projection matrix P such that $P^T \cdot x \in L \; \forall x \in V$ and $P^T \cdot P = I$, where P^T is the transpose of P and I the identity matrix. The projected data $P^T \cdot x$ is the new low-dimensional feature representation of the data. The optimal projection matrix is found via minimizing the difference between the document distributions projected via P. The set $M(p,q) = \{P \mid P \in \Re^{q \times p}, P^T \cdot P = I\}$, together with an inner product $\cdot$, forms a *Stiefel manifold*. A manifold is a topological space that is locally Euclidean: for each point on the manifold we find a neighbourhood that is isomorphic to $\Re^{q \times p}$. Also, a metric is defined on each manifold that measures the distance between two points on the manifold. This local linearity and the metric enable us to define gradients, required for performing Stochastic Gradient Descent.

3.2 Maximum Mean Discrepancy

In order to make the source and target domain similar, we need a way to measure how different their distributions p_S and p_T are. Gretton et al. (2008) propose to use the Maximum Mean Discrepancy (MMD) to estimate the difference in distribution between two domains:

$$MMD^2[p_S, p_T] \quad = \quad \|\mu[p_S] - \mu[p_T]\|_H^2 \tag{1}$$

where $\mu[p]$ is the mean operator

and H denotes the unit ball in a universal Reproducing Kernel Hilbert Space (RKHS). Hence, the MMD measures the difference in distribution as the norm in the RKHS between the means of the mappings of the distributions into this universal RKHS. In all experiments we use Gaussian kernels, which are universal. Using a universal kernel, the MMD measures the difference based on any moment of the two distributions. Gretton et al. (2008) describe how a linear estimation of MMD^2 can be defined as empirical mean over the distances of random draws from the two distributions in the RKHS:

$$MMD^2[Z] \quad = \quad \frac{1}{m} \sum_{i=1}^{\lfloor m/2 \rfloor} h(z_{2i}, z_{2i+1}) \tag{2}$$

where $Z = \{z_1, \cdots, z_m\}$ is a sample of random variables $z_i = (x_i, y_i)$ with $x_i \sim p_S$, $y_i \sim p_T$, and where $h(z_i, z_j) = k(x_i, x_j) - k(x_i, y_j) - k(x_j, y_i) + k(y_i, y_j)$ for a universal kernel $k(.,.)$ which induces the RKHS H. This estimation enables us to use SGD to minimize the MMD between two distributions p_S and p_T.

4 Optimization on the Stiefel Manifold

To find the optimal projection matrix onto a low-dimensional feature representation for domain adaptation, we define an optimization problem that minimizes the MMD with respect to a matrix P such that $P^T \cdot P = I$. The latter constraint is added to avoid rank deficiency. Minimizing the distance with respect to a projection matrix will easily end up with projections that make the data points small in length, collapse them into the origin, or destroy the data structure to match the two distributions (regardless of the rank). To avoid this, we propose to regularize P via $\|P \cdot Z\|_2^2$. This leads to the optimization problem:

$$\min_P MMD[Z_P]^2 - \lambda \frac{1}{m} \cdot \sum_{i=1}^{n} \left\|z_i'\right\|_2^2 \quad \text{s.t. } P^T \cdot P = I$$

with samples $Z_P = \{z_1', \cdots, z_m'\}$ of random variables $z_i' = (P^T \cdot x_i, P^T \cdot y_i)$ for $x_i \sim p_S$ and $y_i \sim p_T$.

To derive a joint update rule for stochastic gradient descent for both the MMD and the expected length, we define the partial cost C_p of the optimization problem for the pair of matrices (z_{2i}, z_{2i+1}) from Z as:

$$C_p([z_{2i}, z_{2i+1}], P) = h(z_{2i}', z_{2i+1}') - \lambda \cdot \left\|[z_{2i}', z_{2i+1}']\right\|_2^2 \tag{3}$$

where the first term comes from the linear approximation of the MMD and the second term regularizes the length of the new feature representation for the drawn data points from the sources. The overall cost after having seen m pairs is derived from the m partial costs:

$$C(Z, P) = \frac{1}{m} \cdot \sum_{i=1}^{m} C_p([z_{2i}, z_{2i+1}], P) \tag{4}$$

4.1 Stochastic Gradient Descent over the Stiefel Manifold

We perform Stochastic Gradient Descent (SGD) on the Stiefel manifold M to find the optimal projection matrix that solves the optimization problem. SGD estimates a sequence of gradients with respect to random draws from the data. Under simple conditions, this sequence converges to the optimum of the corresponding optimization problem; cf. (Bottou 1998). For the SGD, we use the following update rule for the projection matrix P at step t (Bonnabel 2013):

$$P_{t+1} = Exp_{P_t}(H(z_t, P_t), -\gamma_t \cdot \|H(z_t, P_t)\|) \tag{5}$$

where H is the gradient of the cost function on the manifold. From the current projection matrix P_t, we move along the geodesic in the direction of the negative

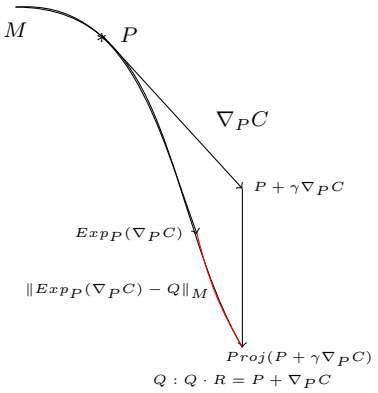

Fig. 1 An optimization step on the Stiefel manifold M. Starting at point P on M, we move in the direction of the gradient $\nabla_P C$. Moving along M ends in $Exp_P(\nabla_P C)$. Moving simply in direction of the gradient ends in a point that must be projected back onto M via (for instance) QR decomposition. The difference of the two points is $\|Exp_P(\nabla_P C) - Q\|_M$, the norm of the difference on the Stiefel manifold.

gradient of the cost function with respect to P_t. We denote by Exp the exponential map that moves a point along the manifold in a given direction (Wen and Yin 2013):

$$Exp_P(H,t) = \left(I + \frac{t}{2} \cdot H\right)^{-1} \cdot \left(I - \frac{t}{2} \cdot H\right) \cdot P \tag{6}$$

The major reason for directly optimizing on the Stiefel manifold is that SGD performs a large number of gradient steps. If we do not stay on the Stiefel manifold, we need to project back onto the manifold after each step due to the constraint $P^T \cdot P = I$. Figure 1 illustrates this with a schematic view on the manifold. The curved line pictures the Stiefel manifold. At each step in the SGD we move from a current point P in the direction of the gradient $\nabla_P C$. Moving just in the direction of the gradient can result in matrices that are far away from the manifold. These matrices must be projected back onto the Stiefel manifold. This results in an error at each step. These errors can result in slower convergence and suboptimal solutions. We investigate this issue in detail in the experimental section.

Nevertheless, we will explore the use of projections onto the Stiefel manifold. This is much easier to compute than the exponential map. Such a projection is a smooth mapping from tangent space (in which the gradient H lies) to the manifold. We can calculate the projection $Proj^{St}$ onto the Stiefel manifold via QR decomposition (Absil et al. 2008):

$$Proj_P^{St}(H,t) = Q \tag{7}$$

$$Q \cdot R = P + t \cdot H \tag{8}$$

For the cost function $C_p([z_i, z_j], P)$ and the next random pair (z_i, z_j) from Z building a new matrix $\hat{z}_t = [z_i, z_j]$ we get the gradient:

$$\begin{aligned} H([z_i, z_j], P) &= \partial_P C_p([z_i, z_j], P) \\ &= \partial_P h(z_i, z_j) - \lambda 2(z_i + z_j)^T \cdot (z_i + z_j) \cdot P^T \end{aligned} \tag{9}$$

consisting of the gradient of the new part of the linear approximation of the MMD and the gradient of the norm of the projected data: we minimize the distance on any two samples from the target and the source domain in Z, projected onto a low-dimensional subspace, in a universal RKHS, while maximizing their length.

The gradient of h depends on the used kernel. For the Gaussian kernel k on the projected points, for instance, we obtain the following kernel definition with respect to the projection matrix P:

$$k\left(P^T \cdot x, P^T \cdot y\right) = \exp\left(-\frac{(x-y)^T \cdot P \cdot P^T \cdot (x-y)}{2 \cdot \sigma^2}\right) \tag{10}$$

which has a gradient of:

$$\partial_P k\left(P^T \cdot x, P^T \cdot y\right) = -\frac{1}{\sigma^2} \cdot k\left(P^T \cdot x, P^T \cdot y\right) \cdot (x-y)^T \cdot (x-y) \cdot P^T \tag{11}$$

4.2 Convergence and Optimality

For Stochastic Gradient Descent convergence, we need a bounded cost function and a compact set over which we optimize. Further, we need to specify the step size γ such that $\sum \gamma_t^2 < \infty$ and $\sum \gamma_t = \infty$. For further details on SGD and convergences see Bonnabel (2013). Our cost function C consists of two parts that both are bounded. On the one hand, the MMD is bounded since it is the norm in a universal RKHS with bounded kernel $k(x, y) \leq K$, hence $0 \leq MMD^2[Z_P] < \infty$. On the other hand, the norm of the projected documents is bounded, since we know that norm of the projection matrix is one and the norm of the data matrix is bounded since the data is already bounded. All together, we see that $0 \leq \left\|P^T \cdot Z\right\|_2 \leq \|P\|_2 \cdot \|Z\|_2 < \infty$. The Stiefel manifold is a compact set, since any sequence of projection matrices from the Stiefel manifold stays on the manifold. Therefore, our proposed optimization by SGD on the Stiefel manifold converges.

Stochastic Gradient Descent might converge to a local minimum or saddle point instead of a global minimum; cf. (Bottou 1998). To overcome this, we perform multiple starts for the optimization: we randomly sample starting points on the manifold and perform the optimization. The optimization result with the smallest cost is used as projection matrix. We use the following calculations to draw uniformly distributed points as starting points for the optimizations over the manifolds. For an arbitrary point W on the Stiefel manifold, $X = X_1 \cdot W \cdot X_2$ is uniformly distributed over the Stiefel manifold, with X_1 a $q \times q$ and X_2 a $p \times p$ normally distributed orthogonal matrix; cf. (Mezzadri 2007).

4.3 Informativeness

An advantage of using linear projections to find low-dimensional latent feature representations for domain adaptation is that they are interpretable. The projection is performed in the vector space that is spanned by the words. Hence, the projection in the individual dimensions corresponds to the word adaptation required to make two domains similar in distribution. The word vectors are rotated and stretched, where the stretching is limited due to the regularization on the feature vector sizes. The amount of rotation in the vector space in certain dimensions tells how much individual words need to be adapted (of weighted). We can gauge how strongly individual words need to be adapted by inspecting the magnitude of the rotation in the vector space in the corresponding dimensions.

Figure 2 illustrates this concept with an artificial example. In two dimensions of a vector space, word vectors of two domains are plotted. Each axis displays the normalized term frequency values (tf-idf values) in one component; each component tells the frequency of a certain word in a document multiplied by a normalization term. The Stiefel method finds latent subspaces such as the diagonal line in the figure. Projecting the vectors from both domains onto this space via the found projection matrix P implies rotating the word vectors. The vectors for "word 1" and "word 2" are rotated to adapt domains. The average rotation required for the red circles is lower than the average rotation required for the blue circles. Hence, although both words are important to adapt domains, "word 2" is more different in the two domains than "word 1". If we find little or no rotation in some dimensions, we conclude that the corresponding words are less important for domain adaptation. In the experimental section, we explore this concept on concrete real-world results.

4.4 Complexity

The complexity of our proposed method depends on two factors. On the one hand, the initialization needs to sample random points on the Stiefel manifold. For this, two random Gaussian matrices X_1 and X_2 must be sampled and orthogonalized. For the matrix X_1 this can be done in $\mathcal{O}(q^3)$ and for matrix X_2 in $\mathcal{O}(p^3)$, where p is the dimension of the word vectors and q is the dimension of the latent feature representations. In general, we assume that $q < p \ll n_T + n_S$, where n_x denotes the number of data samples from domain x. On the other hand, the exponential maps that move the projection matrices along a geodesic need the inversion on the matrix $(I + \frac{t}{2} \cdot H)^{-1}$. This can be done in $\mathcal{O}(p^3)$ with standard techniques. All this results in a complexity of $\mathcal{O}(sp^3)$ for s SGD steps. Transfer Component Analysis; cf. (Pan et al. 2009), which has similar objectives to our methods, has complexity $\mathcal{O}(q(n_S + n_T)^2)$. Hence, our method is to be preferred on data sets that are so large, that $\mathcal{O}\left((n_S + n_T)^2\right)$ storage space or computational complexity is prohibitively expensive.

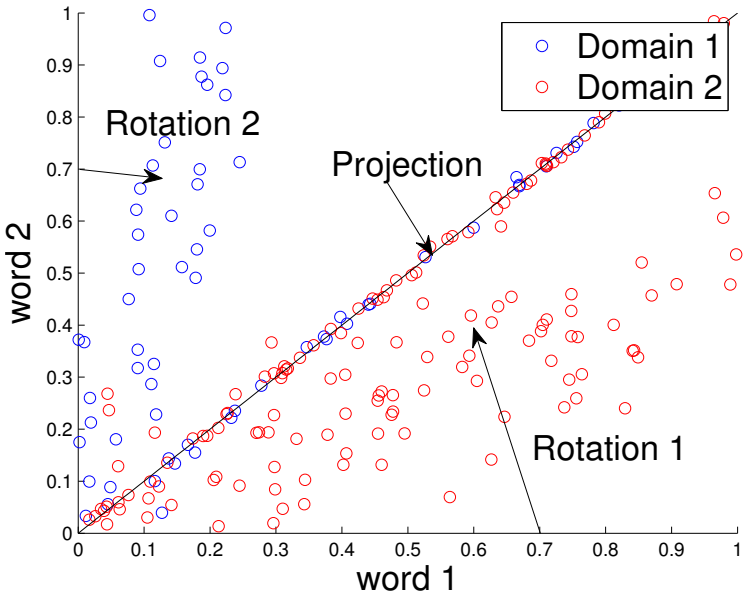

Fig. 2 Informativeness of the projections for domain adaptation: word vectors from Domain 1 and Domain 2 are rotated onto a common latent feature; the rotation magnitude represents how strongly the words need to be adapted to make the domains similar.

5 Experiments

We test the proposed method to find projection matrices onto low-dimensional latent feature representations for domain adaptation, on three standard benchmark data sets that are commonly used in the domain adaptation literature. As first data set, we use the *Amazon* reviews (Blitzer et al. 2007) 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 in less than 95% and more than 5% of all documents. This results in $n = 1993$ words. The second data set is *Reuters-21578* (Lewis et al. 2004). 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; different subcategories are used as source and target domains. We denote the categories Organization by C1, Places by C2 and People by C3. The third data set is *20 newsgroups* (`http://qwone.com/~jason/20Newsgroups/`). We use the top-four categories (comp, rec, sci, and talk) in the experiments. Again, we set up a classification task for each pair of categories. Each category is further split into subcategories and different subcategories are used as source and target domains; each such configuration is denoted by Conf$i$. Documents of categories comp and rec shall be distinguished in Conf1, comp and sci in Conf2, comp and talk in Conf3, rec and sci in Conf4, rec and talk in Conf5, and sci and talk in Conf6.

We implement the SGD in Matlab in the ManOpt library (Boumal et al. 2013) for general Riemann manifolds. The implementations are available at: `http://sfb876.tu-dortmund.de/auto?self=Software` under the link to *Stochastic Gradient Descent on Stiefel Manifolds*.

We compare our proposed SGD on the Stiefel manifold using exponential maps (StOpt) and projections based on QR decomposition (PrOpt) with five state-of-the-art domain adaptation methods: covariate shift adaptation by Kernel Mean Matching (KMM) by Huang et al. (2007), Transfer Component Analysis (TCA) by Pan et al. (2009), SGD on the Grassmann manifold (GrExp) by Baktashmotlagh et al. (2013), Gradient Flow Kernel (GFK) by Gong et al. (2012) and Joint Distribution Adaptation (JCA) by Long et al. (2013).

All experiments were repeated several times; the reported accuracy values correspond to the smallest cost reached during the optimizations. The start points for the optimization are uniformly drawn from the Stiefel manifold.

For all experiments we set the dimension $q = 100$ for all methods and the weight λ to 5. These values have proven empirically to perform best over all data sets; additionally, we show a sensitivity analysis on these two parameters. Unless stated otherwise, we let the SGD perform 1000 steps, after which all experiments showed convergence. We also investigate how the dimension q influences the quality of the domain adaptation for the subspace based methods. Although we get better results for higher dimensions on some data sets, the ranking of the methods by accuracy does not change.

We project all sampled documents onto the new feature representation, and train an SVM classifier on the source documents (after projection) and their labels. Finally, we use labels for the target domain to evaluate the accuracy of the classifier on the target domain (after projection). The labels from the target domain are only used for evaluation. We use an RBF kernel for the SVM with the meta parameter γ. The reported accuracies are the highest ones found by a grid search over the two parameters γ for the kernel and C for the misclassification penalty for the training of the SVM.

5.1 Single-to-Single Domain Experiments

In the first experiment, we use only documents belonging to one designated domain (different from the target domain) as source domain. For example, we use DVD reviews as source domain and book reviews as target domain. On the Amazon data set, we experiment with all possible choices for source and target domain. On the Reuters and the 20 newsgroups data set, we configure the target and source domains as explained above. We perform SGD on the Stiefel manifold to obtain an optimal projection matrix. Here, we use both domains but no labels. Then, the reviews from both domains are projected into the new low-dimensional latent feature representation. An SVM is trained on the projected source domain reviews and evaluated on the projected target domain reviews.

In Tables 2 and 3 we report the results of the first experiment. The SGD on the Stiefel manifold results in a new feature representation for domain adaptation with the highest accuracies over all domains. KMM, TCA and GFK also show good results on some of the domains, but on average they deliver worse accuracies than SGD on the Stiefel manifold. On the Reuters data set, Stiefel outperforms

Table 2 Accuracies on the Amazon reviews, performing domain adaptation from one source domain to one target domain. X→Y denotes training on reviews from X and testing the classifier on reviews from Y.

	E→D	E→B	E→K	D→E	D→B	D→K
KMM	64.7	65.2	80.3	73.7	69.55	77.2
TCA	68.7	70.7	**81.8**	70.7	74.3	74.1
GrExp	61.8	61.8	66.2	58.2	66.0	58.8
GFK	59.8	59.3	68.2	59.4	56.3	61.2
JCA	71.0	67.2	80.8	71.6	76.6	75.4
PrOpt	75.0	73.7	77.2	67.6	71.7	71.2
StOpt	**75.2**	**75.0**	81.4	**75.0**	**78.9**	**76.2**
	B→E	B→D	B→K	K→E	K→D	K→B
KMM	73.0	69.55	73.8	76.7	67.8	63.7
TCA	68.0	71.2	69.6	**83.9**	73.5	74.6
GrExp	57.0	59.6	59.2	62.2	60.4	60.4
GFK	60.4	58.5	61.7	66.2	62.7	60.5
JCA	70.8	73.8	75.7	77.4	71.0	62.6
PrOpt	66.0	71.5	68.2	79.8	78.5	74.1
StOpt	**73.4**	**78.1**	**76.8**	83.3	**78.9**	**76.2**

Table 3 Accuracies on the Reuters and 20 newsgroups data sets.

	Reuters			
	C1→C2	C2→C1	C2→C3	C3→C2
KMM	60.1	56.8	58.5	56.2
TCA	53.0	51.5	58.1	55.8
GrExp	65.0	65.0	70.0	56.8
GFK	72.9	66.1	68.7	66.4
JCA	77.4	80.7	**75.3**	**72.8**
PrOpt	70.0	69.3	72.9	58.2
StOpt	**84.2**	**80.9**	74.7	62.4

	20 newsgroups					
	Conf1	Conf2	Conf3	Conf4	Conf5	Conf6
KMM	96.8	84.4	98.4	91.2	98.5	95.3
TCA	94.4	87.7	96.1	90.1	94.0	88.9
GrExp	88.8	86.4	98.6	87.8	96.7	89.3
GFK	84.0	74.6	91.9	72.4	86.5	79.0
JCA	**99.7**	73.6	55.5	73.0	96.8	88.6
PrOpt	98.7	87.1	**99.4**	96.2	**99.6**	96.4
StOpt	99.4	**93.0**	99.3	**96.6**	99.5	**97.4**

KMM and TCA. On the 20 newsgroups data set, Stiefel outperforms TCA and GrExp. The optimization on the Grassmann manifold has the worst performance of all methods tested.

Comparing the projection and exponential map on the Stiefel manifold, we see differences on all data sets. On the Amazon data set and the Reuters data set, the optimization with exponential map performs much better.

To investigate the quality of the SGD solution, we perform additional experiments. We compare SGD to standard gradient descent (GD1) with random starting points. Further, we use the optimal projection matrix P^* found by SGD as starting point for a gradient descent (GD2). The second setting serves to illustrate that the optimum found by SGD cannot improve much more. The rationale behind using SGD is, besides its applicability to large data sets, that the random behaviour at the start of the SGD process makes its less prone to get stuck in local optima. While GD will stay in the first local optimum it finds, SGD still can escape the

Table 4 Minimal MMD values found when using row X as source domain and column Y as target domain; the first part of the Table gives these values directly for SGD, while the second and third part gives differentials with MMD values found through GD with various starting points. The second part investigates minima reached by GD with random starting points (GD1), and the third part investigates minima reached by GD starting from the SGD results (GD2). The values given are MMD_{GDx}–MMD_{SGD}, and since lower values for MMD are better, a negative value in this cell means that the GD variant finds a better result than SGD, while a positive value means that the GD variant finds a worse result than SGD. This effect is highlighted through cell shading.

MMD_{SGD}	E	D	B	K
E	0	0.0024364	0.0021104	0.0046920
D	0.0028555	0	0.0004567	0.0033506
B	0.0020263	0.0004198	0	0.0027398
K	0.0044783	0.0034033	0.0026004	0

MMD_{GD1}–MMD_{SGD}	E	D	B	K
E	0	0.0006637	0.0012029	-0.0021750
D	0.0000310	0	0.0008182	0.0001125
B	0.0006695	0.0011025	0	0.0007358
K	-0.0019814	0.0000122	0.0009033	0

MMD_{GD2}–MMD_{SGD}	E	D	B	K
E	0	-0.0000004	-0.0000003	0
D	-0.0000002	0	-0.0000007	-0.0000003
B	-0.0000009	-0.0000004	0	-0.0000005
K	-0.0000001	-0.0000001	-0.0000004	0

trap and end up in a possibly better local optimum. This is important, since our optimization problem is non-convex: while the MMD is convex in the Hilbert space induced by the corresponding kernel, it is not convex with respect to a projection matrix of the word vectors. All experiments are repeated 10 times and the results presented are the lowest minimum found for the corresponding methods.

In Table 4, we report the difference of the optimal values found by minimizing only the linearized MMD (see Equation (2)) using the gradient methods with SGD. The first part of the table displays the optimal MMD values found by SGD using row X as source domain and column Y as target domain (where $X, Y \in \{E, D, B, K\}$). The second part of the table displays the difference in MMD optima found by SGD and found by gradient descent using random starting points (GD1). The third and final part of the table displays the same differences, between the optima found by SGD and the optima found by gradient descent using the result from SGD as starting point for optimization (GD2).

Comparing the different gradient methods, SGD finds always a better local optimum than GD1 except for the categories kitchen (K) and electronics (E). These two text collections are already similar in terms of MMD, as we will discuss in the next section. We assume that this closeness in distribution results in fewer local minima. When we start a standard gradient descent from the result found by SGD (GD2), we see that MMD values can only be insignificantly improved (at the seventh position after the decimal point; less than 1‰ of the raw MMD value).

5.2 Multiple-to-Single Domain Experiment

Table 2 shows the accuracies on the target domains using documents from only one category as source domain. Choosing the right category might result in bet-

Table 5 Maximum Mean Discrepancy (MMD) measure on the Amazon data set.

	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	0.0174	0.0200	0

Table 6 Accuracies on the target domains using all the other categories as source domain. The column with label X corresponds to the domain adaptation task $(E \cup D \cup B \cup K \backslash X) \rightarrow X$.

	E	D	B	K
KMM	81.0	75.2	72.5	83.9
TCA	81.4	77.8	74.7	**84.9**
GrExp	68.7	66.3	62.2	70.7
GFK	68.7	66.3	62.2	70.7
JCA	77.0	72.7	74.9	82.3
PrOpt	81.0	75.1	72.7	80.8
StOpt	**82.0**	**78.6**	**76.3**	83.7

ter performance. In the experiments on the Amazon reviews data, we find always one category that outperforms the other categories. For instance, for the categories kitchen (K) the best results are attained when we use the documents from the category electronics (E) as source domain. All other categories cannot bring equivalently good results when employed as source domain.

To investigate this behavior we calculate the Maximum Mean Discrepancy as defined in Equation (1) to estimate the difference of the distributions of the target and source domains; results are displayed in Table 5. For the category electronics (E), the documents from the category kitchen (K) are closest in distributions. Comparing this result with the accuracies in Table 2 on the target domain with documents from category electronics, the documents from category kitchen performs best for domain adaptation. The documents from reviews about DVDs (D) have similar MMD-values among the other categories. This is also reflected in the accuracies above that show no clear category that performs best as source domain. The category kitchen behaves similar to electronics, and books similar to DVDs.

Hence, employing prior knowledge of the target domain to choose the right source domain would be beneficial. Since in many cases this information might not be available, one could resort to using documents from a mixture of all categories but the one used as target domain. In the next experiment, we investigate this setting on the Amazon data set.

The documents from a designated category (E,D,B,K) are used as target domain. From this category we use only the documents. From the other categories we use documents and labels as source domain (as before). Since the source documents stem from three times as many categories as before, in this experiment we let the SGD run for three times as many steps.

In Table 6 we report the accuracies on the target domains for one category using all other categories as source domains. The overall performance on the subspace found by the optimization on the Stiefel manifold is better than KMM and TCA. Again, the optimization on the Grassmann manifold results in the worst results. Comparing the exponential maps to the projections, the computationally more expensive exponential maps find more optimal subspaces. This shows that also on

Table 7 Accuracies on the target domains using all the other categories as source domain using cross validation for the optimal dimension parameter. The column with label X corresponds to the domain adaptation task $(E \cup D \cup B \cup K \backslash X) \rightarrow X$.

	E	D	B	K
TCA	81.4	78.3	75.4	85.2
GrExp	68.7	66.3	62.2	70.7
GFK	81.0	77.5	76.3	82.7
StOpt	**82.3**	**78.4**	**77.0**	**85.3**

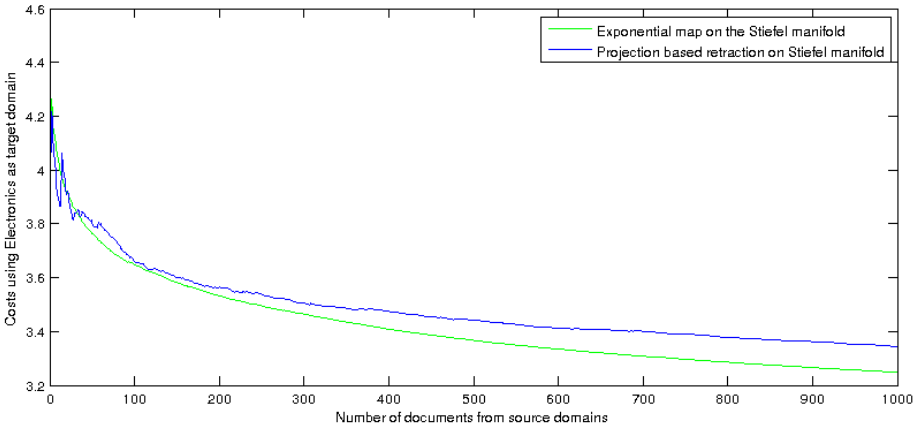

Fig. 3 Convergence of the costs from the optimization problem after a number of documents have been seen. As target domain we use electronic reviews and the source domain consists of the kitchen reviews. On all other possible settings of target and source domains, we get similar convergence results.

a mixture of different categories as source domain, Stiefel manifold optimization results in suitable projection matrices for domain adaptation.

Additionally, we perform an experiment with cross validation for the dimensionality of the subspace for the methods: Transfer Component Analysis (TCA), SGD on the Grassmann manifold (GrExp), Gradient Flow Kernel (GFK) and our approach (Stiefel). We cut off 10% of the target data to find the optimal dimensionality by maximizing the accuracy. On the remaining data, we calculate the final accuracies. The results are reported in Table 7. The SGD on the Stiefel manifold results in the highest accuracies, TCA and GFK perform slightly worse.

5.3 Convergence

The advantage of SGD directly on the Stiefel manifold is that we avoid additional projection steps after each SGD step to satisfy the orthogonality constraint of the matrices. This additional step will induce errors after each SGD step. Consequently, we expect slower convergence when we perform only projections onto the Stiefel manifold. Here we investigate the convergence of the stochastic gradient descent on the Stiefel manifold. We show the costs of the optimization function for the target domain of electronic reviews. As source domain we use reviews about kitchens. Figure 3 plots these costs against the number of documents from both

the target and source domain for the optimization. We report the course of the costs during the optimization of the Stiefel manifold using both a projection by a QR decomposition onto the manifold and the exponential map that moves along the manifold.

Figure 3 shows a fast convergence for both methods. The exponential map has a faster convergence than the projection method, from having seen only few documents onwards. The convergence is quite stable for both methods. The optimization with exponential maps reaches lower cost than the optimization with the projection. This shows that exponential maps can indeed result in better optimization performance using the proposed cost function: optimization on the Stiefel manifold with exponential maps converges faster and reaches a lower cost. This matches the results from the previous experiment that showed typically better performance when using exponential maps as opposed to using projections.

6 Parameter Sensitivity Analysis

The proposed optimization method fixes the dimension of the latent feature representation and the regularization parameter in the cost function. While in the main experiments we used fixed values for the dimension and the regularization parameter, here we investigate different values in a sensitivity analysis.

The dimensionality of the latent feature representation and hence the used manifold M is a meta parameter that has to be chosen beforehand. It is clear that for a good performance we need a large enough number of dimensions to capture all necessary information. On the other hand, the higher the dimensionality, the more computation is needed to estimate the gradient steps. Beside this, too high-dimensional representations might introduce too much variance from the different domains. In Table 8 we show the accuracies on the target domains in the feature representations from the projection matrices found by SGD on the Stiefel manifold for various dimensionalities q. The results show that higher numbers of dimensions generally but not consistently correspond to slightly better accuracies. Hence, without labels for the target domain, the choice should be in favour of large dimensionalities. In case we have labels for the target domain, we can perform cross validation to find the optimal parameter q.

In the experiments so far, we used maximum mean discrepancy and regularization on the norm for the optimization with a fixed parameter $\lambda = 5$. Here, we analyse the difference of the accuracy from the projections that have been found by SGD with various weights on the regularization of the norm. Table 9 shows the accuracies for various weights λ. We see that the regularization of the norm is vital for the performance of the domain adaptation. Without the regularization, the found projection is not able to capture enough information from the domains for a good classifier on the target domain. Higher weights result in better performance on average. This means, that the regularization on the norm helps retaining enough information from the domains necessary to train a good classifier for the target domain.

Table 8 Accuracies on the projected target domain onto subspaces of various dimensionalities q for the target domains. The optimization is on the Stiefel manifold. The classifier is trained on the source domains projected onto the corresponding subspace. The first four columns with label X corresponds to the domain adaptation task on Amazon reviews $(E \cup D \cup B \cup K \setminus X) \to X$; the next four columns correspond to the domain adaptation task on Reuters; the last six columns correspond to the domain adaptation task on the 20 newsgroups data set.

dimensionality q	E	D	B	K	C1→C2	C2→C1	C2→C3	C3→C1
40	77.0	75.9	73.3	79.9	76.5	70.7	66.8	59.0
60	72.2	66.3	70.8	75.4	75.7	70.6	65.3	58.6
80	76.3	74.1	72.2	78.3	73.2	72.3	69.8	58.1
100	74.8	73.5	72.4	80.0	72.6	72.4	72.9	58.2

dimensionality q	Conf1	Conf2	Conf3	Conf4	Conf5	Conf6
40	97.7	92.5	99.9	97.3	99.2	97.0
60	99.6	92.0	99.6	97.7	99.5	98.2
80	99.2	90.8	99.7	96.3	99.7	97.4
100	99.4	91.7	99.5	95.8	99.6	98.2

Table 9 Accuracies on the projected target domain onto subspaces with various weights λ in the optimization problem. The optimization is on the Stiefel manifold. The classifier is trained on the source domains projected onto the corresponding subspace. The first four columns with label X corresponds to the domain adaptation task on Amazon reviews $(E \cup D \cup B \cup K \setminus X) \to X$; the next four columns correspond to the domain adaptation task on Reuters; the last six columns correspond to the domain adaptation task on the 20 newsgroups data set.

weights λ	E	D	B	K	C1→C2	C2→C1	C2→C3	C3→C1
0	64.7	62.3	62.2	64.4	76.5	70.7	66.8	59
1	79.1	71.8	70.9	80.8	74.9	70.8	69.9	58.4
4	78.9	73.2	73.9	82.3	72.4	71.6	71.3	58.8
5	79.4	73.6	74.6	81.7	75.7	70.8	71.0	59.4
10	78.9	73.6	72.8	81.5	73.4	70.6	71.1	58.8

weights λ	Conf1	Conf2	Conf3	Conf4	Conf5	Conf6
0	92.4	82.2	98.8	78.3	94.0	87.7
1	99.3	89.9	99.5	97.4	99.5	98.5
4	99.1	89.4	99.4	96.6	99.5	96.4
5	99.7	87.9	98.9	96.5	99.7	96.4
10	98.8	86.8	99.5	97.1	99.4	97.0

6.1 Informativeness

An important argument for the proposed method is its interpretability. In Figure 4 we plot (as introduced in Section 4.3) for two words the tf-idf values in the vector space of the word vectors for Amazon reviews about books (the source domain) and electronics (the target domain). The top figure shows the tf-idf values that correspond to the words "professional" and "interesting" in the word vectors from both domains. The bottom figure shows the tf-idf values that correspond to the words "display" and "author". The word vectors from the book reviews are represented by blue crosses and the word vectors from the electronic reviews are plotted as red circles. In each figure, the left plot shows the word vectors that correspond to the words before projection, and the right plot shows the word vectors after projecting them with the matrix we found with the proposed method.

We see that the words "professional" and "interesting" are important for the domain adaptation since the corresponding word vectors are rotated in the vector space. The found projection matrix makes the corresponding components of the word vector also more similar in the latent feature representation. This makes

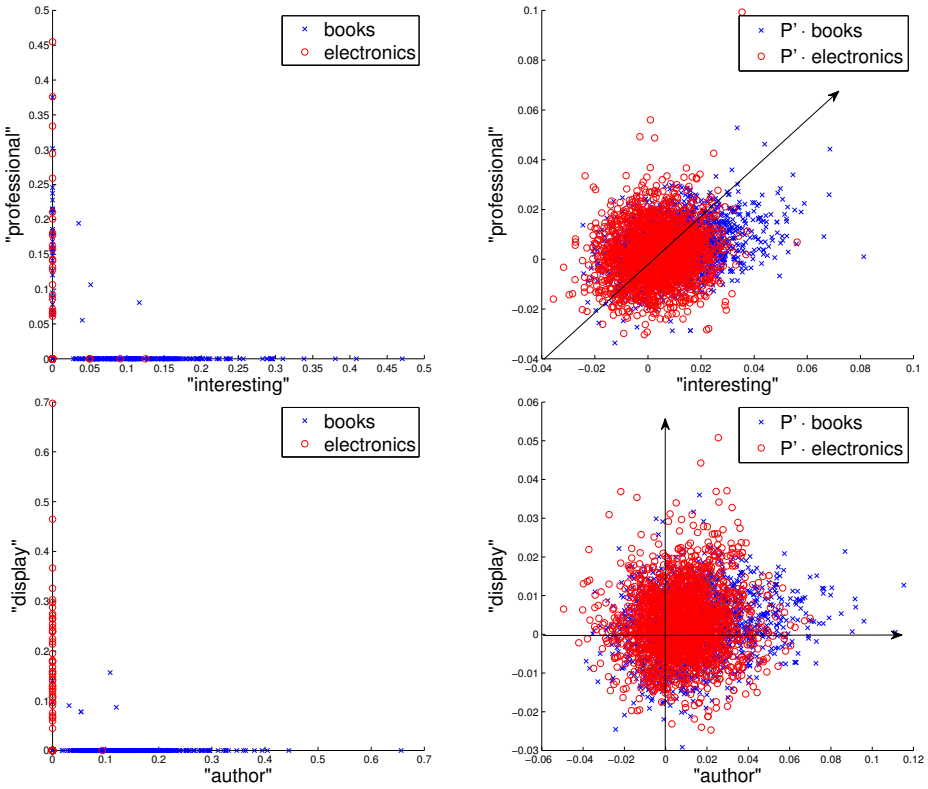

Fig. 4 Tf-idf values of word vectors in the original feature space (left) and projected space (right), for the adaptation of "professional" and "interesting" (top), and "display" and "author" (bottom).

sense, since both words represent a common positive connotation; they are only differently distributed in the two original domains. On the other hand, the conceptually orthogonal words "display" and "author" are less important for domain adaptation: there is only little rotation of the word vectors in the corresponding components. This corroborates the hypothesis that the found projections help interpreting the adaptation needed to adapt the given domains of word vectors.

To further investigate the informativeness of the projections learned for domain adaptation, we visualize the words in a 2-dimensional map. We use the method of Stochastic Neighbourhood Embedding by van der Maaten and Hinton (2008). This methods models the joint probability of two words w_i, w_j as $p(w_i, w_j) \propto e^{-\|x_i - x_j\|^2}$, where x_i, x_j are low-dimensional feature representations of the words.

In Figure 5 we visualize positive adjectives before and after projection with the optimal projection matrix for domain adaptation in the same two-dimensional space for reviews from books and electronic articles. The distance between the adjectives gets smaller after projecting. For instance, the words "perfect" and "useful" are much closer after projection compared to the original data. The word "perfect" appears in 54 reviews of books but in none of the reviews of electronic

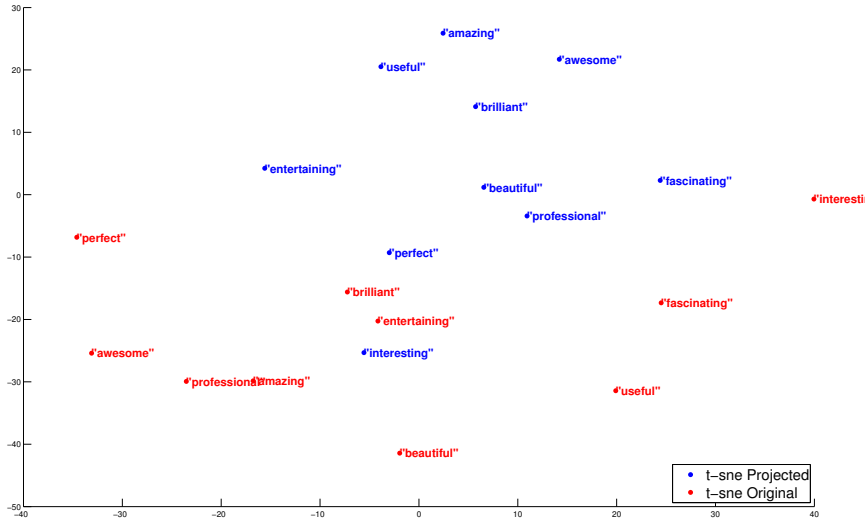

Fig. 5 Visualization of positive adjectives before and after projection.

articles. The word "useful" appears in 106 reviews of electronic articles but only in 54 reviews of books. This distributional mismatch can be seen in the distance of the words in the original space. Clearly, the new feature representation by the optimal projection matrix results in smaller Euclidean distance and hence larger joint probability of the two words.

7 Conclusions

We propose to use Stochastic Gradient Descent (SGD) on Stiefel manifolds to find a projection onto a latent subspace that is best suited for domain adaptation. We provide update rules that compel the SGD steps to remain on the Stiefel manifold, and solve an optimization problem employing these steps. Since the Stiefel manifold encompasses projection matrices on word vectors, the results are interpretable: the importance of a word towards making the domain adaptation can be gauged by measuring the rotation magnitude of the projection of that word, as is illustrated by Figure 4. Furthermore, we have seen that in terms of accuracy, the Stiefel method performs at least as good as or simply better than competing state-of-the-art domain adaptation methods; optimization on the Grassmann manifold cannot compete (cf. Table 2). Kernel Mean Matching and Transfer Component Analysis can deliver comparable accuracies, but these methods are regularly outperformed by Stiefel method as well (cf. Table 3). When increasing the amount of domains from which source documents are taken, this behavior remains (cf. Table 6): accuracy of the Stiefel method is typically best or equivalent to best, while every competing method performs sometimes equivalently and sometimes substantially worse. For domain adaptation, the Stiefel method delivers interpretable results without substantial loss, and even regularly to the benefit, of accuracy.

Analysis of the (dis-)similarities between the multiple category domains on the Amazon data set (cf. Table 5), and their relation to domain adaptation accuracies,

suggests that the Stiefel methods might deliver the greatest benefit when source and target domain are more dissimilar, when domain adaptation typically struggles (Ben-David et al. 2010); the MMD scores in Table 5 show that the E and K domains form the pair which is by far the most similar, and in Table 2 we see that this is the only pair on which the Stiefel method is outperformed. We plan to fortify the hypothesis that Stiefel shines in difficult cases in future work, by further studying larger data sets with many more domains. Also, we plan to investigate what cost functions can be used to find good projections for domain adaptation. Especially, we are interested in different regularizations on the projections. One direction could also be how to integrate external knowledge in the optimization. We could use different sources and different views of the data to bridge the domains for domain adaptation. One possible extension is to use additional class labels given for the different domains. So far we assume to have no such label information for the domain adaptation. Since we show the quality of the domain adaptation based on a trained classifier with given labels for a source domain, the domain adaptation might also use this information. As a preliminary empirical exploration into the potential benefit of using class labels, we can compare the results from JCA and TCA, since JCA can be seen as a variant of TCA which incorporates knowledge about the class labels. For the Single-to-Single domain adaptation, taken over all 22 test cases in Tables 2 and 3, we find that JCA outperforms TCA in 13 cases, whereas TCA ourperforms JCA in 9 cases. In fact, if we remove the Reuters data set (where JCA dominates TCA) from this experiment, we find a perfect tie between the two methods on the remaining two data sets: on the Amazon and 20 newsgroups data sets, TCA outperforms JCA exactly as often (9 times) as vice versa. This is a surprising negative initial result regarding the value of class label information for domain adaptation, which clearly requires further study. domains are already

In the multi-to-Single domain adaptation, the used method also showed no benefit in using additional class label information. In this setting, the label distribution is a mixture from the different source domains. This makes the adaptation with respect to the labels complicated. Different domain adaptation methods using class label information might results in different results but are not the scope of this paper. For the future, we want to further investigate when and how to integrate label information into the domain adaptation, and we would like to explore the generality of the newly proposed methods by experimentally evaluating them on data sets beyond the text domain, such as image data.

Acknowledgments

Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", project A1. Further, this work was supported in part by the European Union through the ERC Consolidator Grant FORSIED (project reference 615517).

References

Absil PA, Mahony RE, Sepulchre R (2008) Optimization Algorithms on Matrix Manifolds. Princeton University Press

Baktashmotlagh M, Harandi M, Lovell B, Salzmann M (2013) Unsupervised domain adaptation by domain invariant projection. In: ICCV 2013

Balzano L, Nowak R, Recht B (2010) Online identification and tracking of subspaces from highly incomplete information. In: Proceedings of Allerton

Ben-David S, Blitzer J, Crammer K, Pereira F (2006) Analysis of representations for domain adaptation. In: Schölkopf B, Platt J, Hoffman T (eds) NIPS, MIT Press, pp 137–144

Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains. Mach Learn 79(1-2):151–175

Bickel S, Brückner M, Scheffer T (2009) Discriminative learning under covariate shift. J Mach Learn Res 10:2137–2155

Blitzer J, Dredze M, Pereira F (2007) Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp 440–447

Bonnabel S (2013) Stochastic gradient descent on Riemannian manifolds. IEEE Trans Automat Contr 58(9):2217–2229

Bottou L (1998) Online Algorithms and Stochastic Approximations. In: Online Learning and Neural Networks, Cambridge University Press

Boumal N, Mishra B, Absil PA, Sepulchre R (2013) Manopt: a Matlab toolbox for optimization on manifolds. arXiv preprint arXiv:13085200 [csMS]

Chattopadhyay R, Sun Q, Fan W, Davidson I, Panchanathan S, Ye J (2012) Multisource domain adaptation and its application to early detection of fatigue. ACM Trans Knowl Discov Data 6(4):18:1–18:26, DOI 10.1145/2382577.2382582

Chen B, Lam W, Tsang I, Wong TL (2009) Extracting discriminative concepts for domain adaptation in text mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pp 179–188

Cheng L, Pan SJ (2014) Semi-supervised domain adaptation on manifolds. IEEE Transactions on Neural Networks and Learning Systems 25(12):2240–2249, DOI 10.1109/TNNLS.2014.2308325

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. Journal of the american society for Information science 41(6):391–407

Dudík M, Schapire RE, Phillips SJ (2005) Correcting sample selection bias in maximum entropy density estimation. In: NIPS

Edelman A, Arias TA, Smith ST (1999) The geometry of algorithms with orthogonality constraints. SIAM J Matrix Anal Appl 20(2):303–353, DOI 10.1137/S0895479895290954

Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: CVPR, IEEE, pp 2066–2073

Gong B, Grauman K, Sha F (2013) Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: Proceedings of the 30th International Conference on Machine Learning, ICML, pp 222–230

Gopalan R, Li R, Chellappa R (2011) Domain adaptation for object recognition: An unsupervised approach. In: Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pp 999–1006, DOI 10.1109/ICCV.2011.6126344

Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola AJ (2008) A kernel method for the two-sample problem. CoRR abs/0805.2368

Huang J, Smola AJ, Gretton A, Borgwardt KM, Schölkopf B (2007) Correcting Sample Selection Bias by Unlabeled Data. In: Advances in Neural Information Processing Systems 19, pp 601–608

Lewis DD, Yang Y, Rose TG, Li F (2004) Rcv1: A new benchmark collection for text categorization research. J Mach Learn Res 5:361–397

Long M, Wang J, Ding G, Sun J, Yu PS (2013) Transfer feature learning with joint distribution adaptation. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp 2200–2207, DOI 10.1109/ICCV.2013.274

van der Maaten L, Hinton GE (2008) Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research 9:2579–2605

McAuley J, Leskovec J (2013) Hidden factors and hidden topics: Understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Sys-

tems, ACM, New York, NY, USA, RecSys '13, pp 165–172, DOI 10.1145/2507157.2507163
Mezzadri F (2007) How to generate random matrices from the classical compact groups. Notices of the AMS 54:592–604
Muandet K, Balduzzi D, Schölkopf B (2013) Domain generalization via invariant feature representation. CoRR abs/1301.2115
Ni J, Qiu Q, Chellappa R (2013) Subspace interpolation via dictionary learning for unsupervised domain adaptation. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, pp 692–699, DOI 10.1109/CVPR.2013.95
Pan SJ, Kwok JT, Yang Q (2008) Transfer learning via dimensionality reduction. In: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI Press, AAAI'08, pp 677–682, URL http://dl.acm.org/citation.cfm?id=1620163.1620177
Pan SJ, Tsang IW, Kwok JT, Yang Q (2009) Domain adaptation via transfer component analysis. In: Proceedings of the 21st International Joint Conference on Artifical Intelligence, IJCAI'09, pp 1187–1192
Pan SJ, Tsang IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22(2):199–210, DOI 10.1109/TNN.2010.2091281
Shao M, Castillo C, Gu Z, Fu Y (2012) Low-rank transfer subspace learning. IEEE International Conference on Data Mining 12:1104–1109
Si S, Tao D, Geng B (2010) Bregman divergence-based regularization for transfer subspace learning. IEEE Transactions on Knowledge and Data Engineering 22(7):929–942
Sugiyama M, Nakajima S, Kashima H, von Bünau P, Kawanabe M (2007) Direct importance estimation with model selection and its application to covariate shift adaptation. In: NIPS
Sugiyama M, Nakajima S, Kashima H, von Bünau P, Kawanabe M (2008) Direct Importance Estimation with Model Selectionand Its Application to Covariate Shift Adaptation. In: Advances in Neural Information Processing Systems 20
Wen Z, Yin W (2013) A feasible method for optimization with orthogonality constraints. Math Program 142(1-2):397–434
Zhang K, Zheng V, Wang Q, Kwok J, Yang Q, Marsic I (2013) Covariate shift in Hilbert space: A solution via sorrogate kernels. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), vol 28, pp 388–395