Kernel Manifold Alignment

Devis Tuia
Department of Geography, University of Zurich, Switzerland
devis.tuia@geo.uzh.ch
Gustau Camps-Valls
Image Processing Laboratory, Universitat de València, Spain
gustau.camps@uv.es

June 9, 2015

Abstract

We introduce a kernel method for manifold alignment (KEMA) and domain adaptation that can match an arbitrary number of data sources without needing corresponding pairs, just few labeled examples in all domains. KEMA has interesting properties: 1) it generalizes other manifold alignment methods, 2) it can align manifolds of very different complexities, performing a sort of manifold unfolding plus alignment, 3) it can define a domain-specific metric to cope with multimodal specificities, 4) it can align data spaces of different dimensionality, 5) it is robust to strong nonlinear feature deformations, and 6) it is closed-form invertible which allows transfer across-domains and data synthesis. To authors' knowledge this is the first method in addressing all these important issues at once. We also present a reduced-rank version for computational efficiency and discuss the generalization performance of KEMA under Rademacher principles of stability. KEMA exhibits very good performance over competing methods in synthetic examples, visual object recognition and recognition of facial expressions tasks.

1 Introduction

Domain adaptation constitutes a field of high interest in pattern analysis and machine learning. Classification algorithms developed with data from one domain cannot be directly used in another related domain, and hence adaptation of either the classifier or the data representation become strictly imperative [1]. In this paper, we focus on the latter pathway, which has been referred to as *feature representation transfer* [2], *feature transformation learning* [3] or *manifold alignment* [4]. Roughly speaking, aligning data manifolds reduces to finding projections to a common latent space where all datasets show similar statistical characteristics. Depending on the availability of labels in the different domains, three families of adaptation problems have been considered in the literature.

Unsupervised adaptation: First attempts of unsupervised domain adaptation are found in multiview analysis [5], and more precisely in canonical correlation analysis

(CCA) and kernel CCA (KCCA) [6]. Despite their good performance in general, they still require points in different sources to be corresponding pairs, which is often hard to meet in real applications. Alternative methods seek for a set of projectors that minimize a measure of discrepancy between the source and target data distributions, such as the Maximum Mean Discrepancy (MMD) [7] or the recent geodesic distance between distributions [8]. However, to compare distributions, the data are supposed to be represented by the same features in all domains. The idea of exploiting geodesic distances along manifolds was also considered in [9], where a finite set of intermediate transformed data distributions are sampled along the geodesic flow (SGF) between the linear subspaces. The intermediate features are then used to train the classifier. The idea was extended in [10], where a Geodesic Flow Kernel (GFK) was constructed by considering the infinity of transformed subspaces along the geodesic path. However, both SGF and GFK assume input data space of the same dimensionality.

Semi-supervised adaptation with labels in the source domain only: Some of the abovementioned methods can incorporate the information of labeled samples in the source domain: For example, SGF [9] and GFK [10] become semi-supervised if the eigenvectors of the source domain are found with a discriminative feature extractor such as partial least squares (PLS). Another family of methods, collectively known as Optimal Transport (OT) techniques, uses labeled samples in the source domain to maximize coherence in the transportation plan of masses between source and target domains [11].

Supervised adaptation with labels in all domains: SGF and GFK can be also defined for the case in which all the domains are labeled. Alternative approaches try to align target and source features while simultaneously moving labeled examples to the correct side of the decision hyperplane (MMDT) [12]. A last family of supervised methods is known as *manifold alignment*, and aims at concurrently matching the corresponding instances while preserving the topology of each input domain, generally using a graph Laplacian [4, 13]. While appealing, these methods still require specifying a small amount of cross-domain sample correspondences. The problem was addressed in [14] by relaxing the constraint of paired correspondences with the constraint of having the same class labels in all domains. The semi-supervised manifold alignment (SSMA) method proposed in [14] projects data from different domains to a latent space where samples belonging to the same class become closer, those of different classes are pushed far apart, and the geometry of each domain is preserved. The method performs well in general and can deal with multiple domains of different dimensionality. However, SSMA cannot cope with strong nonlinear deformations and high-dimensional data problems.

This paper introduces a generalization of SSMA through kernelization. The proposed Kernel Manifold Alignment (KEMA) has appealing properties: (1) it reduces to SSMA when using a linear kernel, which allows us to deal with high-dimensional data efficiently in the dual form (Q-mode analysis): by this property, KEMA can cope with input space of very large dimension, e.g. those extracted by Fisher vectors or deep features; (2) it goes beyond data rotations so it can align manifolds of very different structures, performing a sort of manifold unfolding simultaneous to the alignment; (3) it can also define a domain-specific metric by the use of different kernel functions in the different domains; (4) as SSMA, KEMA can align data spaces of different dimension-

ality; (5) it is robust to strong (nonlinear) deformations of the manifolds to be aligned, as the kernel compensates for problems in graph estimation and numerical problems; and (6) mapping inversion (and hence data synthesis) can be performed in closed-form without the need of pre-images, which permits measuring the quality of the alignment in meaningful physical units.

The remainder of the paper is organized as follows. Section 2 briefly reviews the main properties of the SSMA algorithm. Section 3 introduces the KEMA formulation and analyzes its theoretical and practical properties. Section 4 presents the experimental evaluation of the algorithm. We compare KEMA to SSMA and related (linear and kernel) methods in toy examples and real visual object and face recognition problems. We conclude with some remarks in Section 5.

2 Semi-supervised Manifold Alignment

Let us consider D domains $\mathcal{X}_i$ representing similar classification problems. The corresponding data matrices, $\mathbf{X}_i \in \mathbb{R}^{d_i \times n_i}$, $i = 1, \ldots, D$, contain n_i examples (labeled, l_i, and unlabeled, u_i, with $n_i = l_i + u_i$) of dimension d_i, and $n = \sum_{i=1}^{D} n_i$. The SSMA method [14] maps all the data to a latent space $\mathcal{F}$ such that samples belonging to the same class become closer, those of different classes are pushed far apart, and the geometry of the data manifolds is preserved. Therefore, three entities have to be considered, leading to three $n \times n$ matrices: 1) a similarity matrix $\mathbf{W}_s$ that has components $W_s^{ij} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class, and 0 otherwise (including unlabeled); 2) a dissimilarity matrix $\mathbf{W}_d$, which has entries $W_d^{ij} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different classes, and 0 otherwise (including unlabeled); and 3) a similarity matrix that represents the topology of a given domain, $\mathbf{W}$, e.g. a radial basis function (RBF) kernel or a k nearest neighbors graph computed for each domain separatedly and joined in a block-diagonal matrix. The three different entities lead to three different graph Laplacians: $\mathbf{L}_s$, $\mathbf{L}_d$, and $\mathbf{L}$, respectively. Then, the SSMA embedding must minimize a joint cost function essentially given by the eigenvectors corresponding to the smallest non-zero eigenvalues of the following generalized eigenvalue problem:

$$\mathbf{Z}(\mathbf{L} + \mu\mathbf{L}_s)\mathbf{Z}^\top\mathbf{V} = \lambda\mathbf{Z}\mathbf{L}_d\mathbf{Z}^\top\mathbf{V},$$

where $\mathbf{Z}$ is a block diagonal matrix containing the data matrices $\mathbf{X}_i$ and $\mathbf{V}$ contains in the columns the eigenvectors organized in rows for the particular domain, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_D]^\top$. The method allows to extract a maximum of $N_f = \sum_{i=1}^{D} d_i$ features that serve for projecting the data to the common latent domain as follows:

$$P_{\mathcal{F}}(\mathbf{X}_i) = \mathbf{v}_i^\top \mathbf{X}_i.$$

Advantageously, SSMA can easily project data between domains j and i: first mapping the data in $\mathcal{X}_j$ to the latent domain $\mathcal{F}$, and from there inverting back to the target domain $\mathcal{X}_i$ as follows:

$$P_i(\mathbf{X}_j) = (\mathbf{v}_j\mathbf{v}_i^\dagger)^\top \mathbf{X}_j,$$

where † represents the pseudo-inverse of the eigenvectors of the target domain. Therefore, the method can be used for domain adaptation but also for data synthesis.

3 Kernel manifold alignment

In order to kernelize the previous method one needs to first map the data to a Hilbert space, apply the representer's theorem and replace the dot products therein with reproducing kernel functions. Let us first map the D different datasets to D possibly different Hilbert spaces $\mathcal{H}_i$ of dimension H_i, $\boldsymbol{\phi}_i(\cdot) : \mathbf{x} \mapsto \boldsymbol{\phi}_i(\mathbf{x}) \in \mathcal{H}_i$, $i = 1, \dots, D$. Now, by replacing all the samples with their mapped feature vectors, the problem becomes:

$$\boldsymbol{\Phi}(\mathbf{L} + \mu\mathbf{L}_s)\boldsymbol{\Phi}^\top\mathbf{U} = \lambda\boldsymbol{\Phi}\mathbf{L}_d\boldsymbol{\Phi}^\top\mathbf{U},$$

where $\boldsymbol{\Phi}$ is a block diagonal matrix containing the data matrices $\boldsymbol{\Phi}_i = [\boldsymbol{\phi}_i(\mathbf{x}_1), \dots, \boldsymbol{\phi}_i(\mathbf{x}_{n_i})]^\top$ and $\mathbf{U}$ contains the eigenvectors organized in rows for the particular domain defined in Hilbert space $\mathcal{H}_i$, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_H]^\top$ where $H = \sum_i^D H_i$. This operation is possible thanks to the use of the direct sum of Hilbert spaces, a well-known property of Functional Analysis Theory [15]. Note that the eigenvectors $\mathbf{u}_i$ are of possibly infinite dimension and cannot be explicitly computed. Instead, we resort to the definition of D corresponding Riesz representation theorems [16] so the eigenvectors can be expressed as a linear combination of mapped samples [17], $\mathbf{u}_i = \boldsymbol{\Phi}_i\boldsymbol{\alpha}_i$, and in matrix notation $\mathbf{U} = \boldsymbol{\Phi}\boldsymbol{\Lambda}$. This leads to the problem:

$$\boldsymbol{\Phi}(\mathbf{L} + \mu\mathbf{L}_s)\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\Lambda} = \lambda\boldsymbol{\Phi}\mathbf{L}_d\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{\Lambda}. \tag{1}$$

Now, by premultiplying both sides by $\boldsymbol{\Phi}^\top$ and replacing the dot products with the corresponding kernel matrices, $\mathbf{K}_i = \boldsymbol{\Phi}_i^\top\boldsymbol{\Phi}_i$, we obtain the final solution:

$$\mathbf{K}(\mathbf{L} + \mu\mathbf{L}_s)\mathbf{K}\boldsymbol{\Lambda} = \lambda\mathbf{K}\mathbf{L}_d\mathbf{K}\boldsymbol{\Lambda},$$

where $\mathbf{K}$ is a block diagonal matrix containing the kernel matrices $\mathbf{K}_i$. Now the eigenproblem becomes of size $n \times n$ instead of $d \times d$, and we can extract a maximum of $N_f = n$ features.

Kernel generalization: When a linear kernel is used for all the domains, $\mathbf{K}_i = \mathbf{X}_i^\top\mathbf{X}_i$, KEMA reduces to SSMA:

$$P_{\mathcal{F}}(\mathbf{X}_i) = \boldsymbol{\alpha}_i^\top\mathbf{X}_i^\top\mathbf{X}_i = (\mathbf{X}_i\boldsymbol{\alpha}_i)^\top\mathbf{X}_i = \mathbf{v}_i^\top\mathbf{X}_i.$$

This dual formulation is advantageous when dealing with very high dimensional datasets, $d_i \gg n_i$ for which the SSMA problem is not well-conditioned. Operating in Q-mode endorses the method with numerical stability and computational efficiency in current high-dimensional problems, e.g. when using Fisher vectors or deep features.

Projections to kernel latent space: Projection to the latent space requires first mapping the data $\mathbf{X}_i$ to its corresponding Hilbert space $\mathcal{H}_i$, thus leading to the mapped data $\boldsymbol{\Phi}_i$, and then applying the projection vector $\mathbf{u}_i$ defined therein:

$$P_{\mathcal{F}}(\mathbf{X}_i) = \mathbf{u}_i^\top\boldsymbol{\Phi}_i = \boldsymbol{\alpha}_i^\top\boldsymbol{\Phi}_i^\top\boldsymbol{\Phi}_i = \boldsymbol{\alpha}_i^\top\mathbf{K}_i. \tag{2}$$

Invertibility has a closed-form solution: In order to map data from $\mathcal{X}_j$ to $\mathcal{X}_i$ with KEMA we would need to estimate $D-1$ inverse mappings, which would make KEMA unstable and useless to measure accuracy in meaningful physical units of the input space. In general, using kernel functions hampers the invertibility of the transformation unless pre-imaging is used, for which some efficient yet inexact solutions exist [18,19]. Here we propose a simple closed-form solution to the mapping inversion: to use a linear kernel for the latent-to-target transformation $\mathbf{K}_i = \mathbf{X}_i^\top \mathbf{X}_i$, and $\mathbf{K}_j$ for $j \neq i$ with any desired form. Then, projection of data $\mathbf{X}_j$ to the target domain i becomes:

$$P_i(\mathbf{X}_j) = (\mathbf{u}_i^\dagger)^\top \boldsymbol{\alpha}_j^\top \mathbf{K}_j = (\boldsymbol{\alpha}_j (\mathbf{X}_i \boldsymbol{\alpha}_i)^\dagger)^\top \mathbf{K}_j, \tag{3}$$

where for the target domain we used $\mathbf{u}_i = \boldsymbol{\Phi}_i \boldsymbol{\alpha}_i = \mathbf{X}_i \boldsymbol{\alpha}_i$. We should note that the solution is not unique since D different inverse solutions can be obtained depending on the selected target domain.

3.1 Reduced rank approximation

KEMA complexity scales quadratically with n in terms of memory, and cubically with respect to the computation time. Feature extraction for new data requires the evaluation of n kernel functions *per* pattern, becoming computationally expensive for large n. To alleviate this problem, we propose a reduced-rank approximation of the span. The so-called Reduced-Rank Kernel Manifold Alignment (REKEMA) formulation imposes reduced-rank solutions for the projection vectors, $\mathbf{W} = \boldsymbol{\Phi}_r \boldsymbol{\Lambda}$, where $\boldsymbol{\Phi}_r$ is a subset of the training data containing r samples ($r \ll n$) and $\boldsymbol{\Lambda}$ is the new argument for the maximization problem. Plugging $\mathbf{W}$ into Eq. (1), and replacing the dot products with the corresponding kernels, $\mathbf{K}_{rn} = \boldsymbol{\Phi}_r^\top \boldsymbol{\Phi}$, we obtain the final solution:

$$\mathbf{K}_{rn}(\mathbf{L} + \mu \mathbf{L}_s)\mathbf{K}_{nr}\boldsymbol{\Lambda} = \lambda \mathbf{K}_{rn}\mathbf{L}_d\mathbf{K}_{nr}\boldsymbol{\Lambda},$$

where $\mathbf{K}_{rn}$ is a block diagonal matrix containing the kernel matrices $\mathbf{K}_i$ comparing a reduced set of r representative vectors and *all* training data points, n. REKEMA reports clear benefits for obtaining the projection vectors (eigenproblem becomes of size $r \times r$ instead of $n \times n$), compacting the solution (now $N_f = r \ll n$ features), and in storage requirements (quadratic with r).

3.2 Stability of KEMA

The use of KEMA in practice raises the question of the amount of data needed to provide an accurate empirical estimate and how the quality of the solution differs depending on the datasets. Such results have been previously derived for KPCA [20] and KPLS [21] and here we adapt them to our setting. The following properties are based on the concentration of sums of eigenvalues of the generalized KEMA eigenproblem solved using a finite number of samples, where new points are projected into the m-dimensional space spanned by the m eigenvectors corresponding to the largest m eigenvalues. Following the notation in [20], we refer to the projection onto a subspace U of the eigenvectors of our eigenproblem as $P_U(\phi(\mathbf{x}))$. We represent the projection

onto the orthogonal complement of U by $P_{\mathbf{U}^\perp}(\phi(\mathbf{x}))$. The norm of the orthogonal projection is also referred to as the residual since it corresponds to the distance between the points and their projections.

Theorem 1 (Th. 1 and 2 in [20]) *If we perform KEMA in the feature space defined by* $\mathbf{K}^* = (\mathbf{K}(\mathbf{L} + \mu\mathbf{L}_s)\mathbf{K})^{-1}\mathbf{K}\mathbf{L}_d\mathbf{K}$*, then with probability greater than* $1 - \delta$ *over* n *random samples* S*, for all* $1 \leq m \leq n$*, if we project data on the space* $\hat{U}_m$*, the expected squared residual is bounded by*

$$\sum_{j=m+1}^{n} \lambda_j \leq \mathbb{E}\left[\|P_{\hat{U}_m^\perp}\|^2\right] \leq \min_{1\leq l\leq m}\left[\frac{1}{n}\sum_{j=l+1}^{n}\hat{\lambda}_j(S) + \frac{1+\sqrt{l}}{\sqrt{n}}\sqrt{\frac{2}{n}\sum_{i=1}^{n}K_{ii}^{*2}}\right] + R^2\sqrt{\frac{18}{n}\ln\left(\frac{2n}{\delta}\right)}$$

and

$$\sum_{j=1}^{m} \lambda_j \leq \mathbb{E}\left[\|P_{\hat{U}_m}\|^2\right] \leq \max_{1\leq l\leq m}\left[\frac{1}{n}\sum_{j=1}^{l}\hat{\lambda}_j(S) - \frac{1+\sqrt{l}}{\sqrt{n}}\sqrt{\frac{2}{n}\sum_{i=1}^{n}K_{ii}^{*2}}\right] - R^2\sqrt{\frac{19}{n}\ln\left(\frac{2(n+1)}{\delta}\right)},$$

where the support of the distribution is in a ball of radius R *in the feature space and* λ_i *are* $\hat{\lambda}_i$ *are the process and empirical eigenvalues, respectively.*

The lower bound confirms that a good representation of the data can be achieved by using the first m eigenvectors if the empirical eigenvalues quickly decrease before $\sqrt{l/n}$ becomes large, while the upper bound suggests that a good approximation is achievable for values of m where $\sqrt{m/n}$ is small. These results can be used as a benchmark to test different approaches or to select among possible candidate kernels. Also, note that depending on how much non-diagonal is $\mathbf{K}^*$ (i.e. how large are the manifold mis-alignments), the KEMA bounds may be tighter than those of KPCA. With an appropriate estimation of the manifold structures via the graph Laplacians and tuning of the kernel parameters, the performance of KEMA will be at least as fitted as that of KPCA.

4 Experimental results

We analyze the behavior of KEMA in a series of artificial datasets of controlled level of distortion and mis-alignment, and on real domain adaptation problems of visual object recognition from multi-source commercial databases, and recognition of multi-subject facial expressions.

4.1 Toy examples with controlled distortions and manifold mis-alignments

Setup: the first battery of experiments contains a series of toy examples composed of two domains with data matrices $\mathbf{X}_1$ and $\mathbf{X}_2$, which are spirals with three classes (see the two first columns of Fig. 1). Then, a series of deformations are applied to the

second domain: scaling, rotation, inversion of the order of the classes, the shape of the domain (spiral or line) or the data dimensionality. For each experiment, 20 labeled pixels *per* class were sampled in each domain, as well as 1000 unlabeled samples that were randomly selected. Classification performance was assessed on 1000 held-out samples from each domain.

Latent space and domain adaptation: Figure 1 illustrates the projections obtained by KEMA when using a linear and an RBF kernel (lengthscale was set as the average distance between labeled samples) and the classification errors for the samples from the source domain ($7th$ column) and the target ($8th$ column). The linear KEMA (SSMA) can align effectively the domains in experiments #1 and #4, which are basically scalings and rotations of the data. However, it fails on experiments #2 and #3, where the manifolds have undergone stronger deformations. The use of a nonlinear kernel allows much more flexible solution, performing a sort of unfolding plus alignment in all experiments. In experiment #1, even if the alignment is correct, the linear classifier trained on the projections of KEMA_{lin} and SSMA cannot resolve the classification of the two domains, while KEMA_{RBF} solution provides a latent space where both domains can be classified correctly. Experiment #2 shows a different picture: the baseline error (green line) is much smaller in the source domain, since the dataset in 3D is linearly separable. Even if the classification of this first domain (•) is correct for all methods, classification after SSMA/KEMA_{lin} projection of the second domain (•) is poor, since their projection in the latent space does not unfold the blue spiral. KEMA_{RBF} provides the best result. For experiment #3, the same trend as in experiment #2 is observed. Finally, experiment #4 shows a very accurate baseline (both domains are linearly separable in the input spaces) and all methods provide accurate classification accuracies. Again, KEMA_{RBF} provides the best match between the domains in the latent space.

Alignment with REKEMA: We now consider the reduced-rank approximation of KEMA proposed in Section 3.1. We used the data in the experiment #1 above. Figure 2 illustrates the solutions of the standard SSMA (or KEMA with linear kernel), and for REKEMA using a varying rate of samples. We also give the classification accuracies of a SVM (with both a linear and an RBF kernel) in the projected latent space. Samples were randomly chosen and the sigma parameter for the RBF kernel in KEMA was fixed to the average distance between all used labeled samples. We can observe that SSMA successfully aligns the two domains, but we still need to resort to nonlinear classification to achieve good results. REKEMA, on the contrary, essentially does two operations simultaneously: alignment and data unfolding. Excessive sparsification leads to poor results. Virtually no difference between the full and the reduced-rank solutions are obtained for small values of r: just 10% of examples are actually needed to saturate accuracies.

Invertibility of the projections: Figure 3 shows the results of invertibility of SSMA and KEMA (using Eq. (3)) on the previous toy examples. We use a linear kernel for the inversion part (latent-to-source) and use for the direct part (target-to-latent space) either a linear or an RBF kernel. All results are shown in the source domain space. All

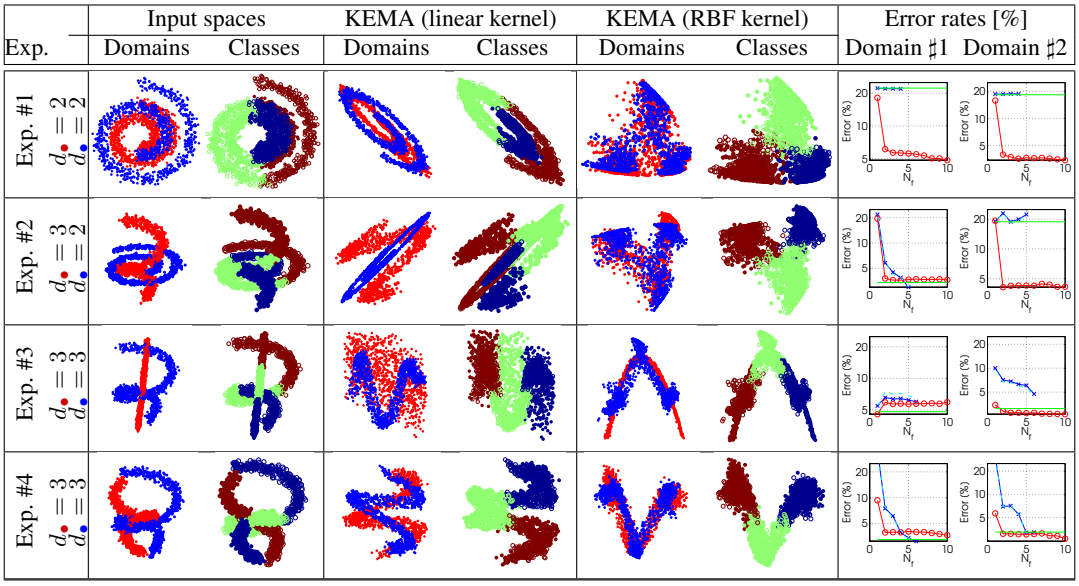


Figure 1: Illustration of linear and kernel manifold alignment on the toy experiments. Left to right: data in the original domains (X1 = •, X2 = •) and *per* class (•, • and •), data projected with the linear and the RBF kernels, and error rates as a function of the extracted features when predicting data for the first (left inset) or the second (right inset) domain (KEMA_{Lin}, KEMA_{RBF}, SSMA, Baseline).

Alignment	No adapt.	SSMA	$\text{REKEMA}_{\text{RBF}}$				KEMA_{RBF}
$r/n \times 100$	100%	100%	1%	10%	25%	50%	100%
SVM_{LIN}	72.33%	81.83%	50.83%	97.17%	97.17%	97.67%	97.83%
SVM_{RBF}	83.21%	95.12%	55.17%	98.12%	98.12%	98.87%	98.89%

Figure 2: Linear and kernel manifold alignment on the scaled interwined spirals toy experiment (Exp. #1 in Fig. 1). REKEMA is compared to SSMA for different rates of training samples (we used $l_i = 100$ and $u_i = 50$ per class for both domains).

the other settings (# labeled and unlabeled, μ, graphs) are kept as in the experiments shown in Fig. 1. The reconstruction error, averaged on 10 runs, is also reported: KEMA is capable of inverting the projections and is always as accurate as the SSMA method in the simplest cases (#1, #4). For the cases related to higher levels of deformation, KEMA is either as accurate as SSMA (#3, where the inversion is basically a projection on a line) or significantly better: e.g. for experiment #2, where the two domain are strongly deformed, only KEMA with RBF kernel can achieve satisfying inversion, as

it unfolds the target domain and then only needs a rotation to match the distribution in the source domain.

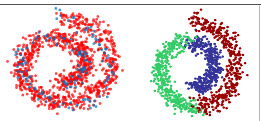
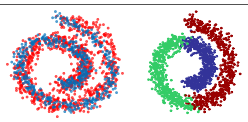
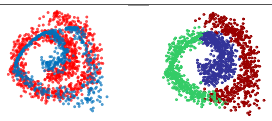
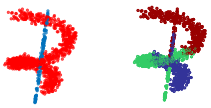
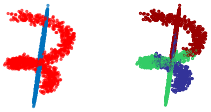
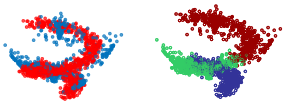
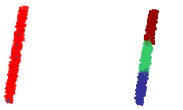
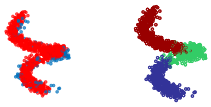
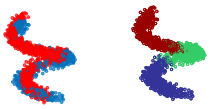
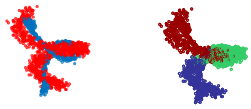
Exp.	SSMA		KEMA ($K_{\text{lin}} \rightarrow K_{\text{lin}}$)		KEMA ($K_{\text{RBF}} \rightarrow K_{\text{lin}}$)	
	Domains	Classes	Domains	Classes	Domains	Classes
Exp. #1						
	0.15 ± 0.01		$\mathbf{0.14 \pm 0.01}$		0.23 ± 0.06	
Exp. #2						
	0.86 ± 0.01		0.87 ± 0.01		$\mathbf{0.34 \pm 0.01}$	
Exp. #3						
	0.19 ± 0.05		$\mathbf{0.15 \pm 0.01}$		0.16 ± 0.01	
Exp. #4						
	0.43 ± 0.03		0.43 ± 0.03		$\mathbf{0.42 \pm 0.04}$	

Figure 3: Domain inversion with SSMA and KEMA. (●) = samples in the source domain, (●) = target domain samples projected onto the source domain, and (●, ●, ●) = class distributions. Each plot shows the result of a single run, and the averaged ℓ_2-norm reconstruction error over 10 runs.

4.2 Visual object recognition in multi-modal datasets

We here evaluate KEMA on visual object recognition tasks by using the dataset introduced in [22]. We consider the four domains Webcam (W), Caltech (C), Amazon (A) and DSLR (D), and selected the 10 common classes in the four datasets following [10]. By doing so, the domains contain 295 (Webcam), 1123 (Caltech), 958 (Amazon) and 157 (DSLR) images, respectively. The features were extracted as described in [22]: we use a 800-dimensional normalized histogram of visual words obtained from a codebook constructed from a subset of the Amazon dataset on points of interest detected by the Speeded Up Robust Features (SURF) method. We used the same experimental setting as [9, 10], in order to compare with these unsupervised domain adaptation methods. Additionally, we compare our proposal with the following semi-supervised domain adaptation methods: SGF [9], GFK [10], OT-lab [11] and MMDT [12].

For all methods, we used 20 labeled pixels *per* class in the source domain for the C,

A and D domains and 8 samples *per* class for the W domain. After alignment, an ordinary 1-NN classifier was trained with the labeled samples. The same labeled samples in the source domain were used to define the PLS eigenvectors for GFK and OT-lab. For all the methods using labeled samples in the target domain (including KEMA), we used 3 labeled samples in target domain to define the projections.

We used sensible kernels for this problem in KEMA: the (fast) histogram intersection kernel, $K_i(\mathbf{x}_j, \mathbf{x}_k) = \sum_d \min\{x_j^d, x_k^d\}$, and the χ_2 kernel, $K_{\chi 2}(\mathbf{x}_j, \mathbf{x}_k) = \exp(-\chi^2/(2\sigma^2))$, with $\chi^2 = \frac{1}{2}\sum_d (x_j^d - x_k^d)^2/(x_j^d + x_k^d)$ [23]. We used $u = 300$ unlabeled samples to compute the graph Laplacians, for which a k-NN graph with $k = 21$ was used.

The numerical results obtained in all the eight problems are reported in Table 1: KEMA outperforms all the unsupervised competing methods and, in most of the cases, improves the results obtained by the semi-supervised methods using labels in the source domain only. KEMA provides the most accurate results in 3 out of the 8 settings when confronted to state-of-the-art (semi)-supervised algorithms, and similar peformance to state-of-the-art GFK in 6 out of the 8 settings. KEMA is as accurate as the state of the art, but with the advantage of handling naturally domains of different dimensionality, and not requiring a discriminative classifier to align the domains such as for MMDT.

Table 1: Accuracy in the visual object recognition study (C: Caltech, A: Amazon, D: DSLR, W: Webcam). 1-NN classification testing on all samples from the target domain (l_{domain}: number of labels per class, * = results reported in [11], † = results reported in [12]).

	Train on source	DA								Train on target
		Unsupervised		Labeled from source only		Labeled from source and target				
	No adapt.	SGF [9]*	GFK [10]*	GFK [10]*	OT-lab [11]*	GFK [10]†	MMDT [12]†	KEMA K_i	KEMA $K_{\chi 2}$	No. adapt
l_S	0	0	0	20	20	20	20	20	20	0
l_T	0	0	0	0	0	3	3	3	3	8
C → A	21.4 ± 3.7	36.8 ± 0.5	36.9 ± 0.4	40.4 ± 0.7	43.5 ± 2.1	44.7 ± 0.8	49.4 ± 0.8	47.1 ± 3.0	47.9 ± 3.2	35.4 ± 2.4
C → D	12.3 ± 2.8	32.6 ± 0.7	35.2 ± 1.0	41.1 ± 1.3	41.8 ± 2.8	57.7 ± 1.1	56.5 ± 0.9	61.5 ± 2.8	63.4 ± 3.4	65.1 ± 1.9
A → C	19.9 ± 1.9	35.3 ± 0.5	35.6 ± 0.4	37.9 ± 0.4	35.2 ± 0.8	36.0 ± 0.5	36.4 ± 0.8	29.5 ± 3.0	30.4 ± 3.3	28.4 ± 1.6
A → W	17.5 ± 3.7	31.0 ± 0.7	34.4 ± 0.9	35.7 ± 0.9	38.4 ± 5.4	58.6 ± 1.0	64.6 ± 1.2	65.4 ± 2.7	66.5 ± 2.9	63.5 ± 2.6
W → C	24.2 ± 1.4	21.7 ± 0.4	27.2 ± 0.5	29.3 ± 0.4	35.5 ± 0.9	31.1 ± 0.6	32.2 ± 0.8	32.9 ± 3.3	32.4 ± 3.0	28.4 ± 1.6
W → A	27.0 ± 1.5	27.5 ± 0.5	31.1 ± 0.7	35.5 ± 0.7	40.0 ± 1.0	44.1 ± 0.4	47.7 ± 0.9	44.9 ± 4.5	45.9 ± 3.9	35.4 ± 2.4
D → A	19.0 ± 2.2	32.0 ± 0.4	32.5 ± 0.5	36.1 ± 0.4	34.9 ± 1.3	45.7 ± 0.6	46.9 ± 1.0	44.2 ± 3.1	45.2 ± 3.4	35.4 ± 2.4
D → W	37.4 ± 3.0	66.0 ± 0.5	74.9 ± 0.6	79.1 ± 0.7	84.2 ± 1.0	76.5 ± 0.5	74.1 ± 0.8	64.1 ± 2.9	66.7 ± 3.1	63.5 ± 2.6
Mean	22.34	35.36	38.48	41.89	44.19	49.30	50.98	48.70	49.80	44.39

4.3 Recognition of facial expressions in multi-subject databases

This experiment deals with the task of recognizing facial expressions. We used the dataset in [24], where 185 photos of three subjects depicting three facial expressions (happy, neutral and shocked) are available. Each image is 217×308 pixels and we take each pixel as one dimension for classification: the problem is $200'508$ dimensional. Each pair $\{$subject,expression$\}$ has around 20 repetitions.

Different subjects represent the domains and we align them with respect to the three expression classes. We used only three labeled examples *per* class and subject.

Results are given in Fig. 4(a): since it works directly in the dual, KEMA can effectively cast the three-domains problem into a single ten-dimensional latent space, where all domains are classified with less than 5% error. This shows an additional advantage of KEMA with respect to SSMA in high dimensional spaces: SSMA would have required to solve a $601'524$-dimensional eigenproblem, while KEMA solves only a 55-dimensional problem. Figures 4(b)-(d) present different visualizations of the two first dimensions of the latent space: subject #1 seems to be the most difficult to align with the two others, difficulty that is also reflected in the higher classification errors. Actually, subject #1 shows little variations in his facial traits from one expression to the other compared to the other subjects (see Fig. 3 in [24]).

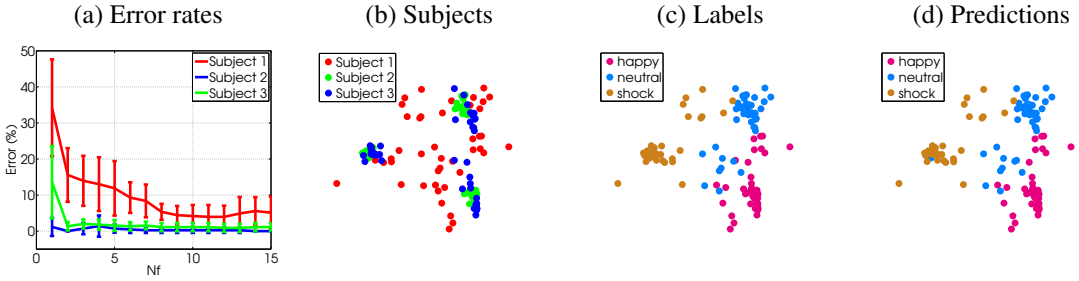


Figure 4: Results of the classification of facial expressions.

5 Conclusions

We introduced a kernel method for semi-supervised manifold alignment. We want to stress that this particular kernelization goes beyond the standard academic exercise as the method addresses many problems in the literature of domain adaptation and manifold learning. The so-called KEMA can actually align an arbitrary number of domains of different dimensionality without needing corresponding pairs, just few labeled examples in all domains. We also showed that KEMA generalizes SSMA when using a linear kernel, which allows us to deal with high-dimensional data efficiently in the dual form. Working in the dual can be computationally costly because of the construction of the graph Laplacians and the size of the involved kernel matrices. Regarding the Laplacians, they can be computed just once and off-line, while regarding the size of the kernels, we introduced a reduced-ranked version that allows to work with a fraction of the samples while maintaining the accuracy of the representation. Advantageously, KEMA can align manifolds of very different structures and dimensionality, performing a sort of manifold unfolding along with the alignment. Importantly, the inversion of the KEMA projections has a closed-form solution without the need of pre-imaging. This is an important feature that allows synthesis applications, but more remarkably allows to study and characterize the distortion of the manifolds in physically meaningful units. To authors' knowledge this is the first method in addressing all these important issues

at once. All these features were illustrated through toy examples and real problems in computer vision and machine learning.

References

[1] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. Neural information processing series. MIT Press, Cambridge, Mass., London, 2009.

[2] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[3] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Proc. Mag.*, 32(3):53–69, May 2015.

[4] C. Wang, P. Krafft, and S. Mahadevan. Manifold alignment. In Y. Ma and Y. Fu, editors, *Manifold Learning: Theory and Applications*. CRC Press, 2011.

[5] D. W. Jacobs, H. Daume, A. Kumar, and A. Sharma. Generalized multiview analysis: A discriminative latent space. In *Proc. CVPR*, pages 2160–2167, Providence, RH, 2012.

[6] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. In *Int. J. Neural Sys.*, pages 365–377, 2000.

[7] S. J. Pan and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Networks*, 22:199–210, 2011.

[8] M. Baktashmotlagh, M. T Harandi, B. C. Lovell, and M. Salzmann. Domain adaptation on the statistical manifold. In *Proc. CVPR*, pages 2481–2488, Columbus, OH, 2014.

[9] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. ICCV*, pages 999–1006, Barcelona, Spain, 2011.

[10] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, pages 2066–2073, Providence, RH, 2012. IEEE.

[11] N. Courty, R. Flamary, and D. Tuia. Domain adaptation with regularized optimal transport. In *Proc. ECML*, pages 274–289, Nancy, France, 2014.

[12] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell. Efficient learning of domain invariant image representations. In *Proc. ICLR*, Scottsdale, AZ, 2013.

[13] J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proc. AISTATS*, pages 120–127, London, UK, 2005.

[14] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI*, pages 1541–1546, Barcelona, Spain, 2011.

[15] M. Reed and B. Simon. *I: Functional Analysis, Volume 1 (Methods of Modern Mathematical Physics) (vol 1)*. Academic Press, 1 edition, January 1981.

[16] F. Riesz and B. S. Nagy. *Functional Analysis*. Frederick Ungar Publishing Co., 1955.

[17] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Patt. Anal. Mach. Intell.*, 29(1):40–51, 2007.

[18] G. Bakır, J. Weston, and B. Schölkopf. Learning to find pre-images. In *Proc. NIPS*, 2003.

[19] J.T. Kwok and I.W. Tsang. The pre-image problem in kernel methods. *IEEE Trans. Neural Networks*, 15(6):1517–1525, 2004.

[20] J. Shawe-Taylor, C. K. I. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Trans. Info. Theory*, 51(7):2510–2522, 2005.

[21] C. Dhanjal, S.R. Gunn, and J. Shawe-Taylor. Efficient sparse kernel feature extraction based on partial least squares. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(8):1347–1361, 2009.

[22] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, pages 213–226, Berlin, Heidelberg, 2010. Springer-Verlag.

[23] V. Sreekanth, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Generalized RBF feature maps for efficient detection. In *British Machine Vision Conference*, 2010.

[24] L. Song, A. Smola, A. Gretton, and K. M. Borgwardt. A dependence maximization view of clustering. In *Proc. ICML*, pages 815–822, Corvallis, OR, 2007.