DOMAIN ADAPTATION USING MANIFOLD ALIGNMENT

MAXIME TROLLIET

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Projet de Master, 2012, Section d'ingénierie de l'environnement
Master Thesis, 2012, Environmental Engineering

Environnement naturel, architectural et construit(ENAC)
École Polytechnique Fédérale de Lausanne (EPFL)

PROFESSOR: Prof. François Golay
ASSISTANT: Dr. Devis Tuia

July 2012
Lausanne

Maxime Trolliet: *Domain Adaptation using Manifold Alignment,* Master Thesis, © July 2012

SUPERVISORS:
François Golay
Devis Tuia

LOCATION:
Lausanne

TIME FRAME:
July 2012

ABSTRACT

Domain adaptation is a major challenge for future remote sensing applications. Both financial and temporal constraints of data acquisition lead to the developing of new techniques able to use knowledge from alternative sources. Different approaches have been developed by considering the statistical properties of images or by modifying already existing classifiers. We propose a intermediary approach of these two kinds of methods by using a manifold alignment technique constrained by similarity between two images. The two images are mapped in a high dimensional latent space which maximizes the proximity of similar elements, thus allowing classification of the images suffering from label scarcity by using the knowledge of the other image. Such a classification offers improvement compared to various used processes.

RÉSUMÉ

L'adaptation de domaine est un challenge important pour les futures applications de remote sensing. Les contraintes tant financières que temporelles relative à l'aquisition de données labelisées incitent à mettre au point de nouvelles techniques permettant l'utilisation d'information provenant de différentes sources. Différentes approches ont vu le jour, considérant les caractéristiqurs statistiques des images ou l'adaptation de classifieur préalablement développés. Dans ce travail, une approche jointe de ces deux types de techniques est considérée en utilisant une méthode d'alignement de manifold, sous contraintes de similarité entre deux images. Les deux images sont projetés dans un espace latent maximisant la proximité des éléments similaires, permettant ainsi une classification de l'image souffrant d'un manque d'éléments labelisés en utilisant celle du domaine de source. Une telle classification offre une possibilité d'amélioration par rapport à différents procédés utilisés.

ACKNOWLEDGMENTS

I would like to thank Dr. Devis Tuia who introduced me to the field of remote sensing. His kind supervision and availability throughout the work opened me to a new scientific domain, bringing me lots of motivation.

I also want to thank the LASIG team for making my work here very pleasant.

Thanks to Diego Joss, long friend of mine who pushed me, not without pain, into the use of LaTeX. His external point of view helped me to step back from my way of thinking and opened my mind. Thanks also to my co-worker Matthew Parkan who helped me with the use of different software and algorithms.

My thanks also go to Pamela Colins, Doctoral student of the Institute of Environmental Engineering who helped me with my English.

Last but not least, my family who's always been behind my back in the down times during this semester. Mélanie, Sophie, Maman vous êtes ma plus belle source d'énergie et de motivation, Merci!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In remote sensing, the availability of labeled data is a common problem for land cover/use survey. The reliability of classification methods often depends on both the amount and the quality of these data. Along with technological developments, the mass of data collected from satellites or airborne sensors has seen an important increase in the past years. New techniques and algorithm have thus to take into account this quantity of available data. However, imagery comes without the labeled data and a significant part of the work is thus to provide it. This procedure represent an expensive task which might hinder the proper development of analysis . When dealing with multiple sources for a single application, it is then essential to think of techniques able to reduce the cost of such labelling processes.

The present work deals with the concept of domain adaptation, also known as transfer learning. The substantial idea is to allow the use of available data in a source domain into a target domain thus reducing the cost of multiple in field data acquisition campaigns. The proposed approach consists in aligning the manifolds of two images from a multi-angular dataset and is based on the work of Chang Wang (2004), of the Department of Computer Science, University of Massachusetts.

This Thesis is organized as follows. In the second section, the theme of domain adaptation is presented with a review of previous work in this field. The third section deals with the concept of manifold alignment and presents the algorithm used. Section four presents the data and classifiers used in this work. In the same section, we announce the used hypotheses and tests made to evaluate the approach. Finally, section five presents the obtained results.

2

ADAPTATION PROBLEM

Traditionally, image classification techniques have been performed on a single data set in a supervised way. The availability of more images over a same area has modified the situation by bringing a temporal dimension to the problem of land cover analysis. Many applications such as change detection have therefore been developed considering this dimension. Other applications have considered the possibility to use different angular scenes to study certain phenomena. These real applications thus deal with multiple sets of imagery. It is then necessary to find a way to combine the knowledge from these different sources, to be able to define efficient models able to acknowledge many images in a single algorithm. Those models raise the question of domain adaptation (DA). The goal of DA or transfer learning consists in the transfer of knowledge from a source domain X^s, into a target one X^t.

When dealing with multiple sources, classification methods are usually based on the assumption that instances from both source and target data are drown from a similar probability distribution, considering an ideal common distribution. In real applications, the different data sets may be highly related, but the assumption of sharing the exact same probability density function (PDF) is unrealistic.

In the field of remote sensing image processing (RSIP), domain adaptation represents a very important challenge. The amount of data collected by satellites grows exponentially as technological improvements are made. Nowadays, data have been gathered for a large part of the globe over the past decades, increasing the temporal dimension greatly. Moreover, the new generation satellites such as WorldView2 allow multi-angle image acquisition during a single flight which permits reduction of the variations in reflectance between to images of a multi-angular set needing multiple satellite flights.

To perform supervised classification methods, labeled information is needed for each image. In most cases, the source domain comes from intensive campaign of data acquisition in field and therefore holds a large amount of labeled data. On the other hand, the target domain often lacks such data.

This lack of labelling usually comes from economical reasons. One wants to reduce time and money spent in another big campaign of data acquisition to acquire usable data for analysis. Therefore, the amount of labeled data in such new acquisitions has to be sufficient for models to achieve good results, but also have to be the smallest

in order to reduce the costs. Domain adaptation thus represents a possibility to reduce costs by developing algorithms able to counter this scarcity in the target domain by improving knowledge.

The results of the field campaign leads to the computation of the ground-truth (GT) consisting in the labeling of representative data. This process needs to be the most representative for the analysed phenomenon or problem for it is determinant for the classification efficiency. For multiple acquisitions, it is not realistic to provide a complete GT for each image. Using a single GT for a series of acquisitions, even if tempting, represents a dangerous short-cut. As the PDFs of images in multi-angle or multi-temporal set show differences, this simplification of data use is most surely bound to lead to disastrous results. Applied to RSIP, one of the domain adaptation approaches is therefore based on the study of the probability distribution functions (PDF) of the images, their similarities and dissimilarities. The goal is to be able to match them in a way that comparison and learning is possible. Another possible approach consists of modifying the classifiers used to make them efficient for multiple domain analysis.

2.1 ADAPTATION APPLIED TO REMOTE SENSING

2.1.1 *Multi Temporal Images*

Temporal survey of the earth's surface is an important part of remote sensing applications at local, regional and global scales. Images taken in the past can be compared to those from the present to asses over time variation of environmental elements, human activities and their impacts. Various algorithms have been developed for this purpose. [6] makes a review of those algorithms and techniques.

Multi-temporal techniques can be separated into two main groups:

1. bi-temporal change detection
2. temporal trajectory analysis

The former compares the data between two acquisition periods and is the most represented type of analysis. The latter considers their changes based on a continuous time-scale making it possible to study the progress or the rate of change. This type of analysis has become feasible with the archives of imagery collected over the years.

Considering two acquisitions of imagery, the physical parameters are most likely to differ greatly between them. Radiometric conditions are influenced by many factors such as the season at which images are taken, solar altitude, meteorological conditions (clouds,

rain, etc.) making it impossible to consider using the same distribution function for the different images to be analysed. In this case, domain adaptation represents an important challenge to achieve more precise results, especially as change detection analysis is increasingly used as a political decision tool for land use survey.

2.1.2 *Multi Angular Images*

Satellites like WorldView2 and CHRIS/PROBA make it possible to acquire a certain number of images of an area during a single flight. This property comes from the ability to re-target the area rapidly with very high resolution. As mentioned before, the use of the data for supervised classification needs a GT. The question of how to deal with this necessity is discussed in [7]. In this situation users have to make a decision between the time-consuming approach of computing a GT for each image or the development of single GT model able to take into account large datasets based on multi-angle imagery. In this work we choose an intermediate option which will be discussed in Chap. 3. New approaches considering multi-angular datasets allow us to consider morphological properties of scenes. Multi-angular reflectance (MAR) shows different ways adding more the information to the classification problem [7] :

1. The MAR contains a partial bidirectional reflectance distribution function (BRDF) over a single satellite track at a single sun angle.
2. The morphology of objects can be taken into account as the object will show a different aspect of itself at each angle, particularly for pitched elements like buildings or vegetation.
3. The difference in reflectance between two images can be measured for surfaces showing spectral variations, thus adding knowledge on the surface properties.
4. The use of MAR minimizes the effect of sun glint. At a given angle, a surface might be obscured due to the geometry of the surrounding landcover. The use of different angles on the same surface can thus make it available for classification.

For similar reasons, multi-angular remote sensing also represents a new approach of biophysical and geophysical parameters assessment. The classical 2-dimensional representation of the image is increased, allowing the retrieval of physical scene characteristics such as cloud morphology and height or vegetation studies by adding vertical information. The ability to measure off nadir radiance provides improved albedo accuracies, allowing a whole new approach for environmental and ecological issues [8],[9]. In [10], these properties are specifically used for classification of vegetation.

Another approach proposed in [11] uses a step by step fusion of CHRIS/PROBA images with a multi-temporal component. Starting from a single multispectral image, other images are progressively added to a Neural Network model, increasing the dimensionality but also resulting in an increase in the classification efficiency for certain classes (mostly on urban structures).

Finally some methods combine information of the nadir image with the heterogeneity of the angular domain in order to generate new information about forest cover [12], [13].

2.2 PECULIARITY OF VHR IMAGES

VHR stands for Very High Resolution. Over the last decades, the spatial resolution of satellite imagery has been greatly increased. For change detection, the introduction of VHR imagery has seen a series of problems arise. Algorithms usually work on a pixel based scale which, at low dimensionality allows good results [14] when observing relatively big objects. With higher resolution, the variance within a single object is no longer melted in a single pixel. Classification techniques have therefore to be modified to take this into account. In the following, we detail the main sources of problems that occur with VHR use.

2.2.1 *Shadows*

At a given sun angle, it is inevitable to have shaded parts in an image. The shadowed part of the image represents a bias for the learning method as the reflectance acquired by the sensor is not representative of the considered surface. VHR images, being more precise, will therefore consider the shadows more precisely thus increasing the bias. In this situation, dealing with multiple images of the same location might reduce the bias as the shaded part of one image might be overlapped by a its corresponding location on a second images for which there is no (or less) shadow.

2.2.2 *Intra-class variance*

Most frequently, the classification discriminability between different land cover classes is determined simultaneously by the spatial resolution and the spectral resolution. With a higher spatial resolution, objects that were not detected previously can no longer be ignored. Different filters can be applied to the images to prevent such missclassifications but have to keep the new interesting information. At lower resolution, the classifications were less sensitive to small variations on a given surface. The heterogeneity of given sur-

face was captured in a single pixel and could therefore be clustered without major incidence. The arrival of VHR changed the situation. The heterogeneity might more easily lead to missclassification in a surface composed of different pixels such as forest containing different kind of trees. Another good example for this problem is the detection of elements on roofs such as chimney or windows. With a resolution of several meters, those elements were not detectable and the pixels encoding the roofs were biased in an affordable way. With a resolution of the meter scale or less, the objects are fully represented, thus leading the learning algorithm to miss the "usual" clustering (a chimney is still part of a roof but may not have the the same radiance at all) leading to missclassification.

2.2.3 *Moving objects*

At lower resolution, classification is mostly used for land cover analysis and thus for static objects. VHR allows identification of elements likely to be moving, such as cars or for airborne imagery, people. Different approaches have been developed in order to allow the recognition of such objects, as well as velocity and direction estimation [15].

2.3 LITERATURE REVIEW

In this section a review of domain adaptation techniques proposed in remote sensing is presented. Different approaches have been explored. While some focus on adapting the classifier, others try to modify the intrinsic properties of the images such as histogram matching or PDF matching. The method used in this work belongs to the latter and uses manifolds alignment (described in Chap. 3) as a PDF matching method with constraints.

2.3.1 *Adaptation of images' statistical properties*

Different approaches to modify the images' properties have been used, such as linear regression, image normalization or non linear histogram shape matching. These methods can usually be used on a band-scale, meaning that for a multi spectral images, the methods are done band by band thus ignoring the correlation existing between different bands. To account for these dependencies, [16] proposes using Gaussian Mixture of the band-histogram,[17] and [18] consider matching the cumulative distributions of the different images by properly taking the correlations into account. The algorithms take advantages of the similarity between the statistical properties of the considered multitemporal images. Different learn-

ing algorithms are tested to assess the improvement of the transfer techniques such as expectation maximization (EM) or maximum likelihood (ML).

2.3.1.1 *Graph Matching*

The consideration of comparing image manifolds has been recently studied to assess the PDF differences. In [19] an approach based on graph matching is proposed for multiple application requiring transfer learning. The method considers maximizing the similarity of two graphs to match them. The application proposes transformation on the source, the target or both domains depending on the desired goal. It looks at specific centroids in both images in sufficient number in order to determine the nonlinear shape of the manifolds. The centroids are to be matched to allow the adaptation.

2.3.2 *Classifier Adaptation*

We hereby present different techiques of classifier adaptation to show the plurality of the possible approaches.

2.3.2.1 *Binary Hierarchical Classifier*

The binary hierarchical classifier (BHC) was first developed for hyperspectral data classification. For a given data set with classes C_i, BHC first divides the classes in two meta-groups based on inter-class affinity. Each meta-group is then treated the same way, constituting a binary tree until there is only one class in the last meta-group level. One of the hypotheses proposed in [20] for transfer learning, is that the class hierarchy might be the same between the source and the target domain meaning that the correlation between classes in X^t should show a similar behaviour as in X^s. In this work BHC is first run on the training data. To consider the change in statistics between the two domains considered, the BHC is modified with the Fisher feature extractor, also known as the Fisher Kernel. This technique is presented in [21] and [22] and consists of a vector of parameter derivatives of loglikelihood of a probabilistic model. It is used to project the data of both X^s and X^t in a reduced dimensional space to assess the shift in distributions. For each node of the BHC tree, a Gaussian mixture model is made to evaluate the initial parameters for expectation maximisation (EM) algorithm. At each iteration of the algorithm, the posterior probability for X^t is determined, allowing to correct the initial Gaussian parameters. As for every optimisation iterative algorithm, the process goes on until a certain threshold is reached. To increase overall accuracies, randomization of the tree structure can be made. However, this randomization has a purpose only if there exists labeled data for X^t; otherwise

it would be impossible to know which random BHC tree bests suits the target domain.

A semi-supervised approach is also presented by weighting the different BHC trees. The algorithm starts with weights $\omega_i = 1$, with $i = 1 : n$. At each iteration weights corresponding to a tree leading to missclassification are reduced by half. The resulting classifier is therefore a combination of the weighted BHC.

2.3.2.2 *Unsupervised Retraining of ML Classifier*

When applied to a single image, ML classifier allows estimation of the statistical properties acting as supervised learning. As explained, using the trained classifier on a new image is not suitable and has to be modified in order to take into account the properties of the new image. The proposal of [23] is to retrain the classifier to estimate the new image *a priori* probability and density function for each class. The first step is to use the parameters of ML previously computed. Then a EM algorithm is used for the estimation of the parameters allowing ML to efficiently work on the new image.

2.3.2.3 *Kernel Adaptation*

In [24], the transfer learning considers a synergistic use of both labeled and unlabeled data. The approach proposes the modification of the prior kernel constructed over labeled data by considering the information contained in the unlabeled data of X^t. The structure of the core kernel is therefore deformed by the unlabeled samples. Samples undergo a first clustering algorithm and EM gives the maximum likelihood estimation of the PDF of a Gaussian mixture of samples. Once the samples are assigned to the clusters, the similarity between clusters is computed by considering the mass centres of clusters. Note that the authors leaves the choice of doing this in the feature space or in the original input space to the user.

2.3.2.4 *Active Learning Strategy*

In [25], an active learning (AL) approach is presented. During AL, the algorithm is iteratively choosing the best set of samples (the more informative ones) for constituting the training set. This process is well defined for a single image classification problem. The proposed transfer learning approach is based on two query functions. In the first query, q^+, the minimum number of the most informative elements of the target domain are taken iteratively the same way that AL algorithms usually do. They are then labeled and added to the training set, thus reducing the costly step of labeling of a new domain. In the second query, q^-, samples from X^s which are not representative are being removed from the training set. These

elements do not fit the distribution of the classes in X^t and could lead to missclassification. For both queries, the user defines a parameter, respectively h^+, the total number of elements taken in the target domain, and h^-, the number of elements to be removed at each iteration. These two parameters define the ratio $\alpha = h^+/h^-$ to be optimized. α depends on both the size of the original training set and the correlation between X^s and X^t.

The aforementioned unsupervised techniques (2.3.1) allow interesting ways to modify the data in a blindfolded way. The choice of the classifier can be left to the user, as only the statistical properties of the data are changed. However, the performance of such techniques often require the two PDFs to be close enough. Depending on the analysis, this choice of adaptation might be questioned.

The semisupervised learning methods(2.3.2) show that using few labeled points, with unlabeled points greatly improves the classifiers performances. This conclusion leads to the question of the number of samples to be taken in target sets susceptible to increase the results, but at an affordable cost.

In this work, we propose a mixed approached by adding constraints to the PDF matching in order to asses both topology preservation and matching instances. The following chapter presents the concepts used for this approach.

3

DOMAIN ADAPTATION WITH MANIFOLD ALIGNMENT

3.1 MANIFOLDS

To understand the concepts used in this work, one has to understand the notion of manifold. Mathematically speaking, a manifold $\mathcal{M}$ is a topological space that is locally Euclidean. We say that the n-dimensional manifold is homeomorphic to the Euclidean space of the same dimension n meaning they share the same topological properties [26]. Practically speaking, a manifold $\mathcal{M}$ is the low dimensional embedding of an object defined in a higher dimensional space. A typical example is shown in fig. 1.

In real life problems, it is generally impossible to have access to the true underlying manifold of a data set. The user has to approximate it from a point cloud [27]. To do so, an adjacency graph associated with the point cloud is constructed and will act as an empirical proxy for the researched manifold. The manifold of an image allows to consider its intrinsic structure. The comparison of images' manifolds for multi-angular or multi-temporal data sets can then be used for domain adaptation. In this adaptation problem, we speak of manifold alignment.

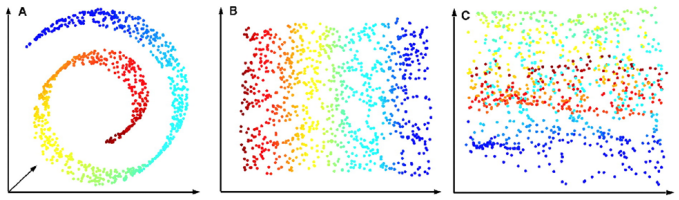


Figure 1: *(A) The "Swiss roll" data used in [1] to illustrate their algorithm. (B) The two-dimensional representation, nearby points in the 2D embedding are also nearby points in the 3D manifold, as desired. (C) "False" representation of data. Neither the metric nor the topological structure is preserved. (B) is unrolled while (C) is a projection.* Image taken from [1]

3.2 MANIFOLDS ALIGNMENT

Alignment of manifolds was presented in [28] for semi-supervised methods as a dimensionality reduction purpose for very high dimensional data sets, sensitive to computational demand. In this field, non linear thechniques have recently been developed, such as

ISOMAP [1] and Laplacian eigenmaps [29]. [2] states that the use of such non linear techniques, when studying the manifold, are more effective than linear ones such as PCA.

In order to be able to achieve transfer learning between X^s and X^t, the two manifolds have to correspond. Rather than forcing one (or both) to match the other by deforming and loosing information, [2] approaches the problem by unifying the representation of the data sets by *aligning* the different domains' manifolds on a joint manifold. To do so, these methods search for a common embedding in a joint latent space. One of the main assumption for this to be done is that the initial manifolds must share a similar structure, as if they were samples of a same unique manifold. This unique manifold underwent different deformations by several external factors such as the acquisition conditions or angle, resulting in the observed domains. The author defines alignment as *dimensionality reduction with constraints induced by the correspondences among the data sets*. Formally, we need to find the mapping function which will project both X^s and X^t into the shared latent space (called the Hilbert space) $\mathcal{H}$ in which we want the corresponding instances across data sets to be the closest. The Laplacian eigenmap [29] is a representation of the data similarity within a single dataset. In this approach, it is necessary to consider multiple data sets. This is achieved by joining the single Laplacians into a joint graph holding the information of the multiple domain set.

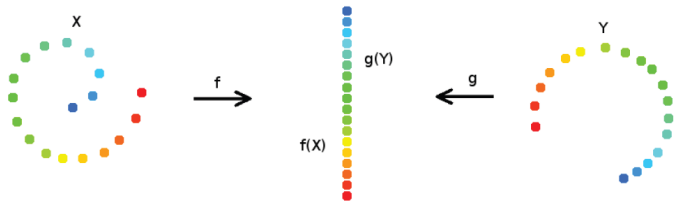


Figure 2: From [2], the figure illustrates to problem of manifold alignment. The twos sets are embedded in a common space, preserving the similarity of both sets

3.2.1 *How to match domains*

We first need to compute the similarity graph, by defining a similarity matrix W for each data set. In this graph, the vertices correspond to the data points $\boldsymbol{x_i}$. If the similarity w_{ij} between two data points is greater or equal to a user-defined threshold, the corresponding vertices are connected, creating an edge. The edges is then weighted by w_{ij}. In [30] different commonly-used similarity graphs are presented. The user has to chose the one which will suit the problem. The most-used graphs are the ϵ-neighborhood graph,

the k-nearest neighbour (knn) graph and the Gaussian graph, also known as the fully connected graph.

In the case of the knn graph, the similarity is not symmetric and results in a directed graph meaning that for two elements $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, if $\boldsymbol{x}_j$ is considered a neighbour of $\boldsymbol{x}_i$, the reciprocity is not assured.

A way to symmetrize it is to consider the mutual knn graph in which the vertices are connected if and only if they are mutual neighbours. Another way to do this is to allow the connection of the vertices if one of them is considered neighbour by the knn method. In this work, we use this last version.

Considering the weight w_{ij}, we have the weight matrix W where $w_{i,j} = w_{j,i} \geqslant 0$. This holds for the similarity within a single domain $W^{(a)}$. To align the manifolds, we also need to consider correspondences between the different domains, encoded in other matrices $W^{a,b}$, with a and b : indexes of two different input sets. The resulting overall geometrical similarity matrix for k domains is then :

$$W = \begin{pmatrix} \nu W^{(1)} & \mu W^{(1,2)} & ... & \mu W^{(1,K)} \\ ... & ... & ... & ... \\ \mu W^{(K,1)} & \mu W^{(K,2)} & ... & \nu W^{(K)} \end{pmatrix} \tag{3.1}$$

Here the two factors ν and μ are scalars weighting the two principles of the alignment: local similarity and correspondence information. When the two of them are of equivalent importance they can be considered equal to one. In this case, $W^{(a,b)}$ and W^K share the same kind of similarity. However, this work deals with two different kinds of similarity: the geometrical (or local) similarity, being the spectral similarity within a single domain, and the class correspondence similarity. Knowing this, a new approach for the definition of the similarity matrix has to be defined. The geometrical similarity is taken individually for each image because of shifts existing between different images. The matrix holding the geometrical similarity is then :

$$W = \begin{pmatrix} W^{(1)} & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & W^{(K)} \end{pmatrix} \tag{3.2}$$

This gives us a joint graph representation of the multiple sets and is the basis to construct the joint Laplacian. The first step is to compute the degree matrix :

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{3.3}$$

and the degree matrix D is defined as the diagonal matrix with degrees d_i on its diagonal.

The unormalized graph Laplacian can then be computed as:

$$L = D - W = \begin{pmatrix} L_1 & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & L_K \end{pmatrix} \tag{3.4}$$

L is symmetric and positive semi-definite.

The next step is to propose a cost function to be minimized. Considering the unified representation of the data, Z (the joint embedding of the data), we can write the cost function:

$$C(Z) = \sum_{ij} \|Z(i,\cdot) - Z(j,\cdot)\|^2 W(i,j) \tag{3.5}$$

[2] shows that this formulation is equivalent to :

$$C(Z) = Tr(Z^T L Z) \tag{3.6}$$

This equation says that if two instances are similar, their location in the latent space should be close. Therefore minimizing this cost function is the same as mapping the similar instances closer. Considering $Z = [z_1, ..., z_N]$ with N the total number of samples, the optimal solution is given by :

$$\underset{Z:Z'DZ=1}{\arg\min}\ C(Z) = \underset{z_1,...z_d}{\arg\min} \sum_i z_i' L z_i + \lambda(1 - z_i' L z_i) \tag{3.7}$$

The solution of optimisation is the d eigenvectors corresponding to the d smallest eigenvalue of the generalized decomposition $Z^T L Z v = \lambda Z^T D Z v$.

3.2.2 *How to force proximity*

With the geometrical similarity being defined, we need to consider the class-similarity between two sets. In order to assess this correspondence, we consider the labels available in the different domains as proposed in [3]. Two graph Laplacians are constructed over the labeled data of both sets. Therefore we have to compute two new Laplacians: one for the similarity between labeled features of the different sets, and one for their dissimilarity. In this work we only consider two data sets, but the methods is valid for K different data sets.

1. Similarity matrix for labeled data :

$$W_s = \begin{pmatrix} W_s^{1,1} & ... & W_s^{1,K} \\ ... & ... & ... \\ W_s^{K,1} & ... & W_s^{K,K} \end{pmatrix} \tag{3.8}$$

$$W_s^{a,b}(i,j) = \begin{cases} 1 & \text{if } x_a^i \text{ and } x_b^j \text{ are from the same class,} \\ 0 & \text{otherwise} \end{cases}$$

2. Dissimilarity matrix for labeled data :

$$W_d = \begin{pmatrix} W_d^{1,1} & ... & W_d^{1,K} \\ ... & ... & ... \\ W_d^{K,1} & ... & W_d^{K,K} \end{pmatrix} \tag{3.9}$$

$$W_d^{a,b}(i,j) = \begin{cases} 1 & \text{if } x_a^i \text{ and } x_b^j \text{ are from different classes,} \\ 0 & \text{otherwise} \end{cases}$$

For both matrices, the case of $W_{s,d}(i,j) = 0$ includes the case of an unlabeled feature. The associated degree matrices and Laplacians are computed the same way as in eq. (3.4) and (3.3).

With the correspondences between data sets being changed, the joint adjacency matrix (3.1) has to be modified. The element of the diagonal being the similarity within a same set, we only need to keep the diagonal elements. The similarity graph can thus be evaluated for each data set and the resulting joint Laplacian is a bloc diagonal matrix.

$$L = \begin{pmatrix} L_1 & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & ...L_K \end{pmatrix} \tag{3.10}$$

Note that, since similarity is computed for each domain separately, there is no obligations for the domains to have the same dimensionality. We then have three different Laplacians : L, L_s and L_d respectively for the spectral similarity, the class similarity and the class dissimilarity. We need to define a new cost function which will consider those three elements. The intuition leading to the formulation is that elements of the same original data set should be embedded closely; elements from different data sets, but sharing same labels should also be as close as possible, and, on the contrary, dissimilar ones should be far.

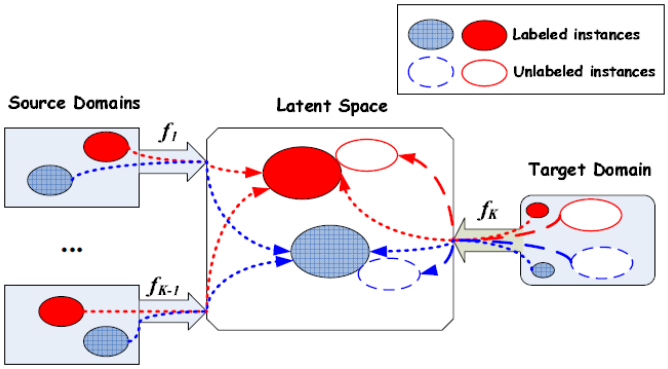


Figure 3: From [3], the figure illustrates the principle of manifold alignment using the labels. The case shows two different classes, blue and red

First let us consider the notation used, according to [3]. We have K different input data sets $X_k = p_k \times m_k$ with p_k, the number of features (here the number of spectral bands), and m_k the number of elements for data set X_k. We want to find the k mapping function f_k to map the input data in $\mathcal{H} \in \mathbb{R}^d$, the latent space. The mapping functions are gathered in $\gamma = (f_1^T, ..., f_K^T)$ which is a $p_1 + ... + p_K \times d$ matrix. Let us define Z as :

$$Z = \begin{pmatrix} X_1 & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & ...X_K \end{pmatrix} \tag{3.11}$$

Then let us formalize the intuition we had for the cost function. For the similarity of classes between different input sets, [3] gives :

$$A = 0.5 \sum_{a=1}^{K} \sum_{b=1}^{K} \sum_{i=1}^{m_a} \sum_{j=1}^{m_b} \|f_a^T x_a^i - f_b^T x_b^j\|^2 W_s^{a,b}(i,j) \tag{3.12}$$

A represents how far points of the same class are mapped and thus has to be minimized. A will constrain two similar instances $\boldsymbol{x}_a^i$ and $\boldsymbol{x}_b^j$ to be projected to a close location in $\mathcal{H}$.

$$B = 0.5 \sum_{a=1}^{K} \sum_{b=1}^{K} \sum_{i=1}^{m_a} \sum_{j=1}^{m_b} \|f_a^T x_a^i - f_b^T x_b^j\|^2 W_d^{a,b}(i,j) \tag{3.13}$$

B is the opposite of A, meaning that we want the two dissimilar instances x_a^i and x_b^j to be projected far from each other. Therefore B has to be maximised.

$$C = 0.5\mu \sum_{k=1}^{K} \sum_{i=1}^{m_a} \sum_{j=1}^{m_b} \|f_a^T x_k^i - f_b^T x_k^j\|^2 W_k(i,j) \tag{3.14}$$

C is the geometrical constraint. Here, for the same input space, if x_k^i and x_k^j are geometrically similar in their input space, they must be mapped close to each other, meaning we need to minimize C. The μ factor acts as a weight parameter. It has to be adapted to treat the preservation of topology and matching instances as equally important. Note that C is the cost function of Laplacian eigenmaps for each domain respectively.

The three elements are then assembled in a unique cost function for the algorithm to achieve the optimisation on those three elements. Since A and C must be minimized and B must be maximized, the cost function to minimize can thus be formulated as :

$$\mathcal{C}(f_{i=1}^k) = \frac{A + C}{B} \tag{3.15}$$

Yet this formulation is pretty cumbersome. [29] shows a way to make it more usable. Remembering $D_{i,i} = \sum_j W_{i,j}$ and $L = D - W$, for a given Laplacian equation we have :

$$\begin{aligned} \sum_{i,j} (f_i - f_j)^2 W i,j &= \sum_{i,j} (f_i^2 + f_j^2 - 2 f_i f_j) W i,j \\ &= \sum_i f_i^2 D_{i,i} + \sum_j f_j^2 D i,i - 2 \sum_{i,j} f_i f_j W_{i,j} \\ &= 2 \boldsymbol{f}^T L \boldsymbol{f} \end{aligned} \tag{3.16}$$

The factor 2 can be removed as we placed a 0.5 in front of A, B and C.

We can rewrite the three terms as :

1. $A = Tr(\gamma^T Z L_s Z^T \gamma)$
2. $B = Tr(\gamma^T Z L_d Z^T \gamma)$

3. $C = Tr(\gamma^T Z L Z^T \gamma)$

For two matrices G and Q, the eigenvalue decomposition allows to solve :

$$G^{-1} Q v = \lambda v \tag{3.17}$$

However if G is not invertible, we can use the generalized eigenvalue decomposition. The formulation is then :

$$Q v = \lambda G v \tag{3.18}$$

v is an eigenvector of both G and Q corresponding to the eigenvalue λ.

The solution of the optimization of the cost function is provided by the eigenvector corresponding to the lowest eigenvalues of the generalized eigenvalue decomposition equation :

$$Z(\mu L + L_s) Z^T v = \lambda Z L_d Z^T v \tag{3.19}$$

λ being the eigenvalue and v its associated eigenvector.

The eigenvectors found this way correspond to the desired mapping functions. As the dimensionality of the latent space is not defined, the algorithm gives an appreciation of its maximum dimension :

$$d_{max} = \sum_{i=1}^{K} p_k \tag{3.20}$$

3.2.2.1 *Choice of similarity graph*

To compute the graph Laplacian we need to chose an adequate similarity function. [3] proposes a variant of the Gaussian similarity with $\sigma = 1$. However the choice of σ is an important factor to obtain good results. A to big σ would result in over-estimate the similarity. Figs. 4 and 5 shows the differences between the graphs. Two cases of Gaussian graphs are shown, illustrating the problem of σ estimation. In the *k-nn* graph, a black dot means neighbourhood between two elements.

However, the Gaussian similarity graph attributes a similarity value for each pair of samples which resulted in an important computational cost. Moreover the appreciation of σ could be tricky.

Therefore a *k nearest neighbour* graph was chosen with $k = 4$ and forced the symmetry by having two samples be set as similar if at least one of them was recognized neighbour.

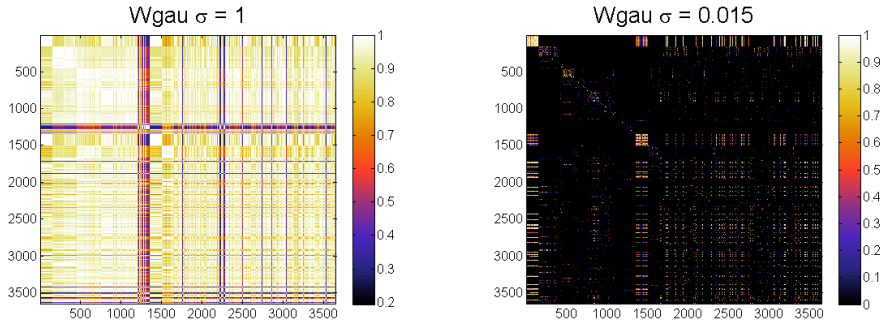


Figure 4: The figures show the difference between two Gaussian similarity graphs. a)shows the case where a to big σ is considered.

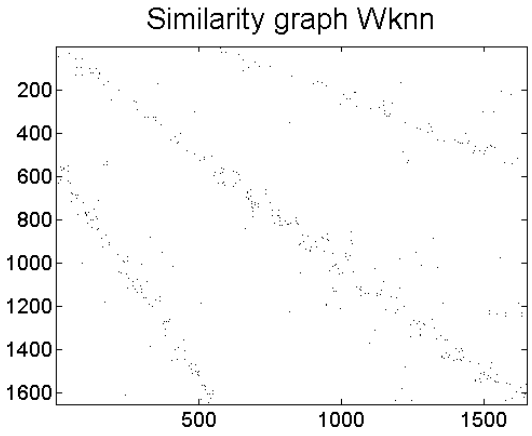


Figure 5: Illustration of the *K-nn* graphs.

3.2.2.2 *Latent space optimal dimension*

To apply manifold alignment (MA), the choice of the dimension of the latent space had to be evaluated in order to assess the contribution of each dimension. As the projection functions found are given by eigenvectors, the question of the direction of the projection also arose. The result of the generalized eigenvalue decomposition is shown in Fig. 6.

$$\gamma = \begin{bmatrix} f^{(1)} & \dots & f^{(d_{max})} \end{bmatrix} = \begin{bmatrix} f_1^{(1)} & \dots & f_1^{(d_{max})} \\ f_K^{(1)} & \dots & f_K^{(d_{max})} \end{bmatrix}$$

Figure 6: Representation of the matrix γ.

A single eigenvector holds the projection functions for all domains. γ is thus the concatenation of the different domains' projection functions. The eigenvectors give a match but can point in opposite directions. This results in an axial reflection of one domain in the latent space. Fig 7 and 8 illustrate this.

The three first dimensions gave a suitable result as the two domains seem to match each other. The visualisation of dimensions 6-8 showed the problem of reflection. This kind of geometrical mismatch could most likely decrease the performance of the classification. As the direction of the eigenvectors can cause such errors, we investigated the relation between different parts of the eigenvectors e.g the one containing the projection for X^t (in blue in Fig. 6) and the one for X^s (in green). In such situation the different parts of the eigenvectors correspond to the p_k lines corresponding to the K sets. The angle between to vectors is given by the relation :

$$\cos\theta = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|} \tag{3.21}$$

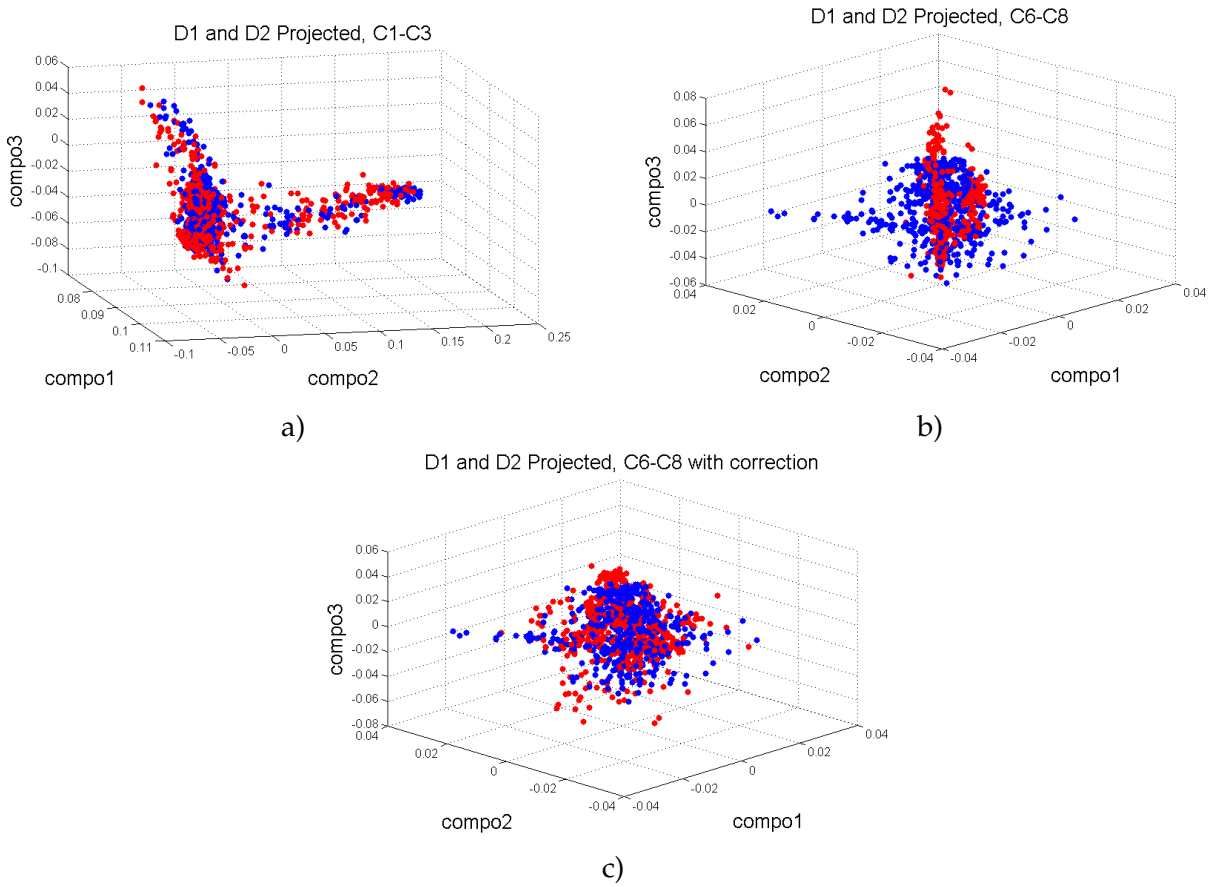


Figure 7: Visualisation of two different dimensional projections. a) the first three dimensions, b) dimensions $6-8$, c) $6-8$ with correction

The direction of the vector can be considered using the trigonometric circle. For a given vector, the direction will not impact the value of the cosine of the angle it makes with the axis, but only its sign. Eq. (3.2.2.2) tells us that the sign of the cosine is given by the dot product as the denominator is strictly positive.

In order to correct this, we consider the sign of the vector cosine for each dimension. For a given projection function $f^{(n)} \in \gamma$ (e.g for a given dimension) we apply the dot product to the two parts of the eigenvector corresponding to the two domains. If the result is negative, a correction can be made by multiplying $f_2^{(n)}$ by -1. By doing so, one must be sure to always apply the modification to the same part of the vector. In order to assess the amelioration we can build the Bhattacharyya's distance over the two sets.

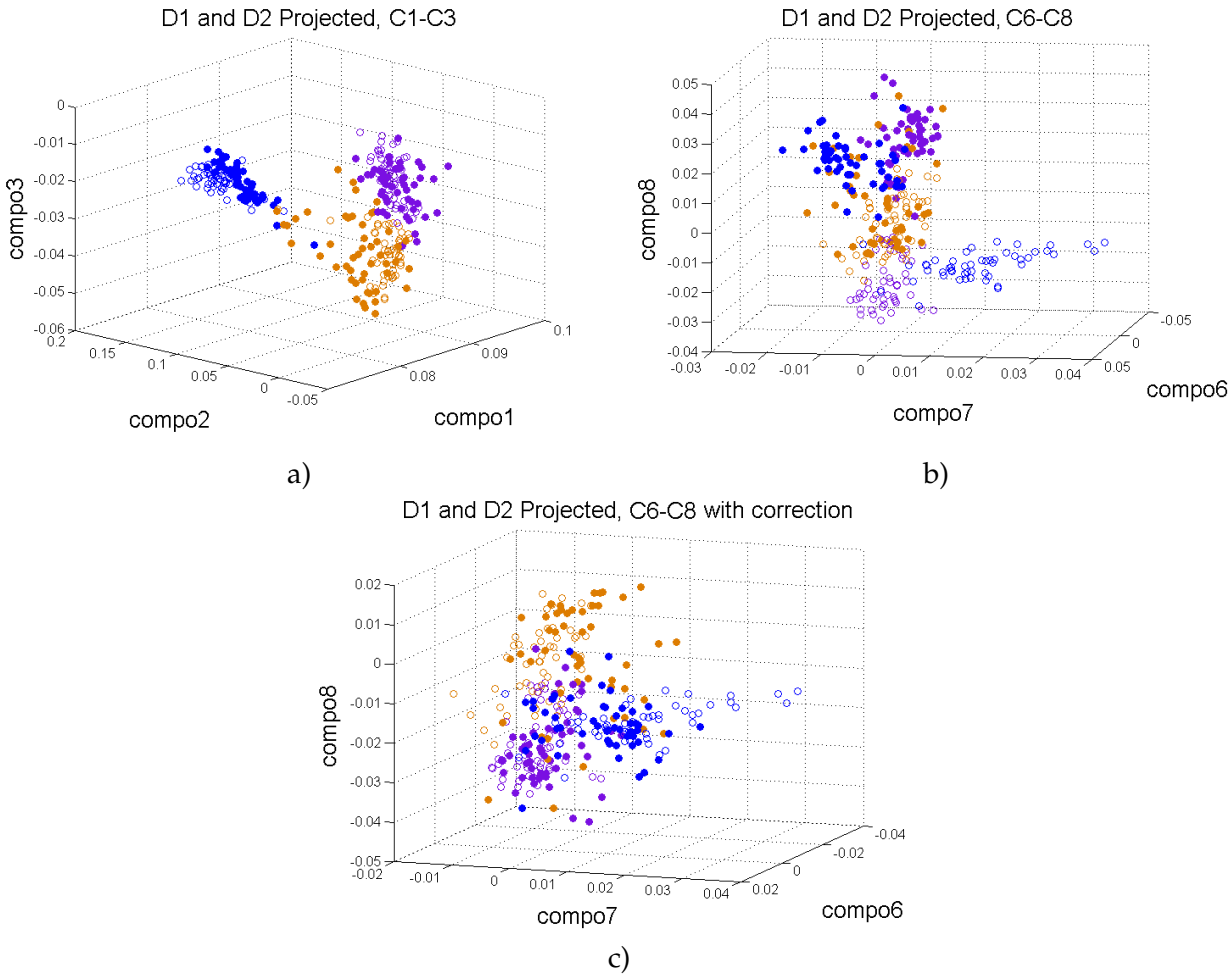


Figure 8: Visualisation of two different dimensional projections of three classes. a) the first three dimensions, b) dimensions $6-8$, c) $6-8$ with correction.

The Bhattacharyya's distance is a similarity measurement between two distributions and is an approximate measurement of the amount of overlap between two statistical samples [31]. We compute de Bhattacharyya's distant measurement (BDM) for each class before and after the correction [1]. Practically speaking, the BDM gives the distance between two samples' centroid. In our case, we want the centroids of projected samples for a same class to be the closest.

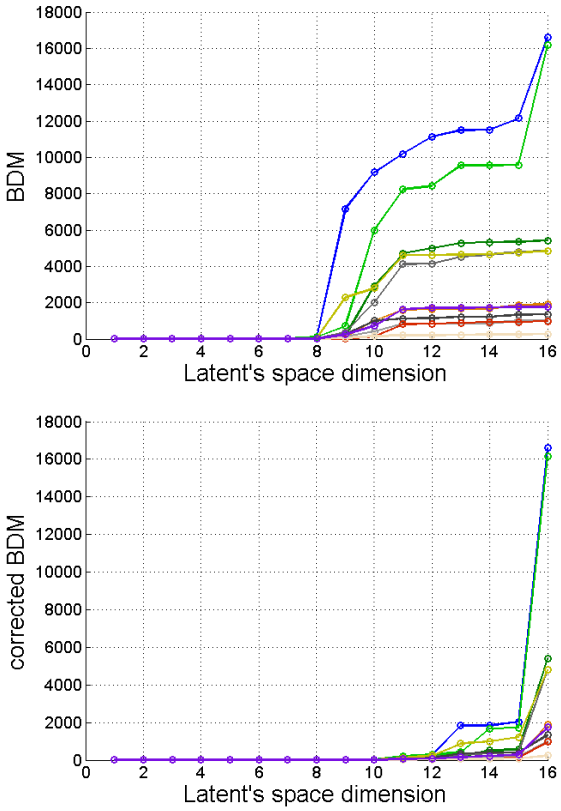


Figure 9: Bhattacharyya's distance for each class between the two sets. For color significance, please refer to Tab. 1.

As we see on Fig. 9, the correction brought by the "*dot product correction*" (DPC) seems to improve the projections. All classes show a similar behaviour with a very low BDM until the 12^{th} dimension of the latent space after correction. We can assume that the alignment gets more noisy from this dimension on.

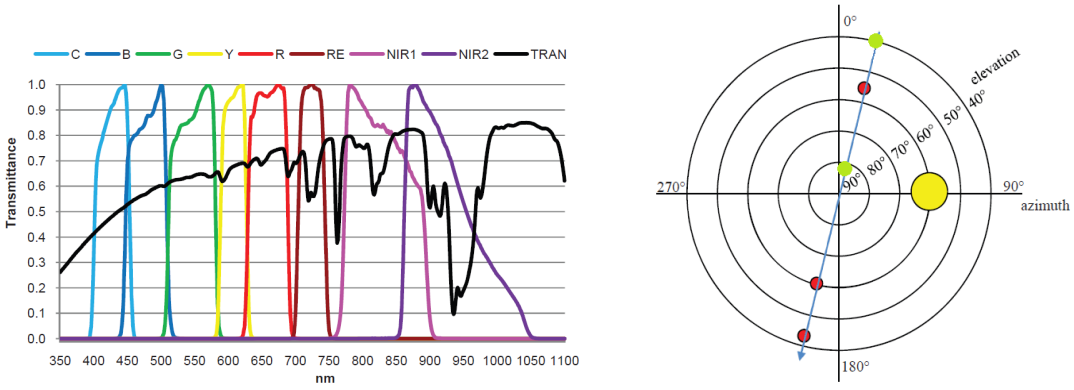
1. Here we use the matlab function proposed by Yi Cao based on [32]

4 APPLICATION

4.1 DATA

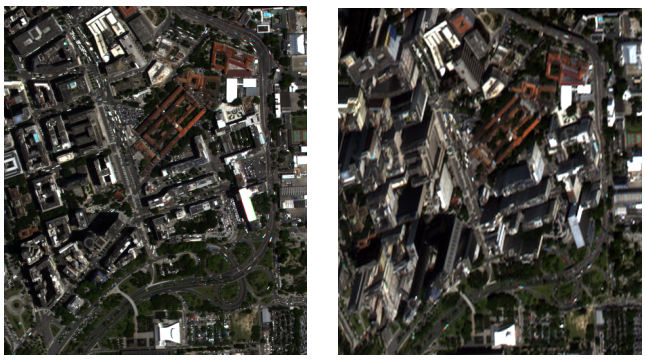
The imagery used in this work was collected over Rio de Janeiro (Brazil) with DigitalGlobe's WorldView-2. This satellite was launched in 2009 and provides 46 centimeters panchromatic and 8 spectral bands at 1.85 meters resolution. In this study, we used two acquisitions from the multiangular dataset of [4]. The first image used is the most nadir one of the set with an angle of 6.09 degrees. The other one is one of the two most off-nadir image with an angle of 47.29 degrees, in order to have the maximum differences between the two sets.

For both data, a ground truth was computed using the software ERDAS IMAGINE. The ground truths hold twelve different classes presented in Fig. 1. A problem was encountered when using the labeled data. Class 3 : Maintained Water was under represented and did not have enough instances to allow the proper use of an interesting amount of element per class. Moreover, the pixels in this class were swimming pools and so were not representative of a particular scene observed in landcover. Therefore class 3 was fused in class one : Natural Deep Water. The classes and the number of elements available in the GT for each image are shown in Tab. 1



a) b)

Figure 10: a) spectral domain for the most nadiral acquisition. Image taken from [4], b)azimuth and elevation of the multi angular acquisition. The green dots are the scenes used in this work and red ones are other scene acquires during the same flight.



a) b)

Figure 11: a) part of most nadir image used, b) part of the off-nadir image used

Num	Class Name	color	Num El X_1	Num El X_2
1	Natural Deep Water		66307	92317
2	Grass		8127	8127
3	Trees		13066	4278
4	Concrete		719	707
5	Soil		790	818
6	Asphalt		2949	2780
7	Building Grey		5936	7194
8	Building Red		1070	1205
9	Building White		1571	1742
10	Shadows		705	2172
11	Plane Ground		5179	5179

Table 1: Legend used in this work and number of labeled elements available per image

4.2 CLASSIFIERS

In this work, we aim at assessing the feasibility of transferring a classification model with the manifold alignment technique proposed. Since the technique requires labeled samples in both domains, we test it on supervised classification scenarios. In supervised techniques, the learning algorithm is fed with a set of "true" data to learn the model. This means that for a subset of the data, there are labels available constituting $\{x_i, y_i\}_{i=1}^{n}$ pairs which constraint the output. This work is only using supervised techniques like the support vector machine (SVM) or the naïve Bayes classifier.

4.2.1 *Naive Bayes*

Classification with a Bayesian framework has been vastly used because of its ease of use and relatively good results [33], [34]. Naive bayes (NB) is a linear classifier with strong independent assumptions from which the *naive* terms comes. It is based on the Bayes' Theorem which finds the probability of an event occurring given the probability of another event that has already occurred. The classification is based on the probability of a sample x to belong to a class c_i. The sample will be classified in the class showing the biggest *a-posteriory* probability $p(c_i|\boldsymbol{x})$ for $i = 1 : C$, the total number of classes. The probability functions of the different classes are computed with the training set $(\boldsymbol{x_i}, y_i)_{i=1}^N$ and allows to evaluate the optimal class for the new unseen samples.

4.2.2 *Support Vector Machine Classifiers*

The Support Vector Machine (SVM) is a binary linear classifier based on the statistical learning theory (SLT) proposed by Vapnik and Chervonenkis in the 70's. In a non linear classification problem, SVM is able to use methods allowing it to project the data in a higher dimensional space where the data are linearly separable. For those reasons, SVM is often considered one of the bests supervised classifiers [5],[35], [36]. As for all supervised methods, SVM needs labeled data to train a model that will then be able to predict unlabeled data.

For a given set of data X, we want to learn a mapping function $\boldsymbol{x}_i \longmapsto y_i$. To achieve this, we consider a set of functions $F(\boldsymbol{x}, \boldsymbol{\theta})$ with weight $\boldsymbol{\theta}$ able to approximate the output $\hat{y}$ for a new unseen data. In this family of functions, we search the optimal $f(\boldsymbol{x}, \boldsymbol{\theta})$ according to a given cost function L which measure the performance of the functions $f \in F$.

As the SVM is a supervised learning method, the data set is composed of N pairs $(\boldsymbol{x_i}, y_i)_{i=1}^N$ of independent and identically distributed (iid) observations with y_i the labels. This means that the data follow an unknown joint distribution $P(\boldsymbol{x}, y)$ with a probability density of $p(\boldsymbol{x}, y)$. The main idea in SVM is to find a (hyper)plan that achieve a binary classification of data by maximizing the distance between itself and the data.

4.2.2.1 *The Linear Case*

For better understanding of SVM's mechanisms, let us consider a linear functional model :

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional input vector, $\boldsymbol{w} \in \mathbb{R}^d$ is the parameter vector of the hyperplane to be optimized and b is a scalar. By stating a binary problem, we mean that the labels consist in $y_i \in \{-1; 1\}$.

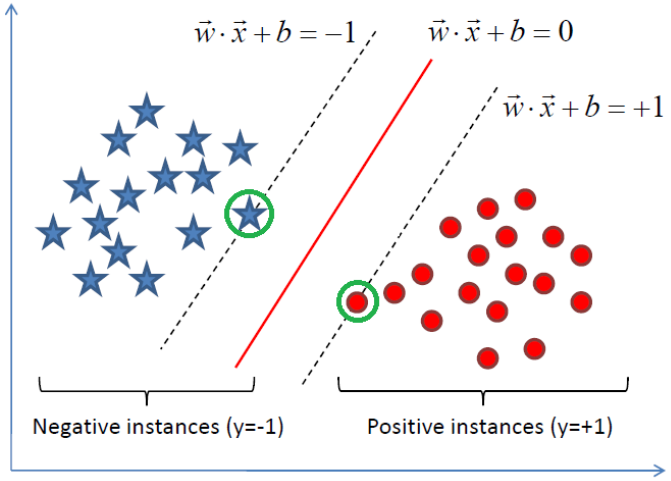


Figure 12: A margin classifier. Image taken from [5]

By considering its position to the hyperplane, and thus the results of $\boldsymbol{w} \cdot \boldsymbol{x} - b$, a data point will be attributed to either class "1" ou "-1". The job of SVM will specifically to find the one hyperplane (and thus the decision function) that maximizes the distance between itself and the data points. This distance is called the margin and corresponds to the perpendicular distance between the hyperplan and the nearest point of each class. AS shown in Fig. 12 the distance must be equal for both classes, thus the margin is symetrical. Since then, the decision function can be written as :

$$y_i = (\mathbf{w} \cdot \boldsymbol{x}_i + b) \geqslant 1 \tag{4.2}$$

We must then maximize the width of the margin d. We compute the euclidean distance between the hyperplan and the closest point of each class which is equal to twice the margin's width. The two closest points considered lie on the margin and are called the support vectors. For each point $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ the distance is respectively $\boldsymbol{w} \cdot \boldsymbol{x}_1 + b = 1$ and $\boldsymbol{w} \cdot \boldsymbol{x}_2 + b = -1$. By rearranging in terms of $\boldsymbol{x}$, and knowing the distance is $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ we have :

$$\boldsymbol{x}_1 = \frac{1-b}{\boldsymbol{w}} \qquad \boldsymbol{x}_2 = \frac{-1-b}{\boldsymbol{w}} \tag{4.3}$$

$$d = \left\| \frac{1-b+1+b}{\boldsymbol{w}} \right\| = \frac{2}{\|\boldsymbol{w}\|} \tag{4.4}$$

Therefore, to maximize d we need to minimize $\boldsymbol{w}$.

However, the formulation of the term to minimize does not give a convex problem to solve meaning there is no global minimum to reach. Therefore we put a quadratic term to$\|\boldsymbol{w}\|$. For computational reason, it is also necessary to add a $\frac{1}{2}$ coefficient. The term to minimize thus becomes $\frac{1}{2}\|\boldsymbol{w}\|^2$.

As the optimization problem to solve is now a strictly convex case, Lagrangian multipliers α_i can be used as linear constraints for the resolution. Therefore we have :

$$L_p(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}^2\| - \sum_{i=1}^{n} \alpha_i \left[y_i((\boldsymbol{w} \cdot \boldsymbol{x}_i) + b) - 1\right] \tag{4.5}$$

L_P needs to be minimized with respect to $\mathbf{w}$ and b and maximized with respect to α. This is achieved by the resolution of the partial differential equations. We can rewrite L_P as :

$$L_P(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}^2\| - \sum_{i=1}^{n} \alpha_i y_i(\boldsymbol{w} \cdot \boldsymbol{x_i}) - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \tag{4.6}$$

and the partial differential equations gives the results :

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \text{and} \qquad \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i = \boldsymbol{w} \tag{4.7}$$

By substitution, we have a new formulation which only is in terms of α :

$$L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\boldsymbol{x_i} \cdot \boldsymbol{x_j}) \tag{4.8}$$

with the constraint on α :

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \text{and} \qquad \alpha_i \geqslant 0 \tag{4.9}$$

This solution depends only on the dot product between points associated with a non-zero α. Those specific points are called the support vectors and are located on the margin. All other points have an associated $\alpha = 0$. The equation for a new unseen data can then be formulated, constituting the linear support vector machine classifier :

$$\hat{y} = f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}) + b \tag{4.10}$$

4.2.2.2 *Soft margin*

In real application though, linearly separable data are seldom seen. Overlapping between classes and noise make it impossible for SVM to achieve this complete linear classification. The margin has to take in count singularities to allow misclassification by softening the constraint. In such a case, the SVM is said to be a soft margin classifier. The softening is performed by adding a penalty element ξ in Eq (4.2) representing the tolerance of the hyperplane to missclassification.

$$y_i = (\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geqslant 1 - \xi \tag{4.11}$$

The minimization is no more on

$$\frac{1}{2}\|\boldsymbol{w}\|^2 \qquad \text{but on} \qquad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{4.12}$$

with $C \geqslant 0$ a parameter controlling the generalization capabilities of the classifier. C allows to give more or less importance to the missclassification distances ξ_i, the greater C, the more strict the model is.

4.2.2.3 *Non linear case : the Kernel based SVM*

For real life application, linear separation is difficult to achieve, even with the soft margin. The introduction of the kernel function will make it possible to overcome this limitation. If, for a dataset $X \in \mathbb{R}^d$, linear classification is not possible, it is possible to map the data into a higher dimensional Hilbert space $\mathcal{H}$. According to Cover's theorem stating that a complexe pattern-classification problem is more likely to be linearly separable in a higher dimensional space [37][38], the mapping of the data set in $\mathcal{H}$ gives the possibility to run a now linear classification.

However it not easy to compute explicitly the mapping, since it is unknown *a priori* and is high dimensional (possible infinite). To overcome this, we use a property of SVM that it only depends on the dot protuct. In Eq (4.10), $\hat{y}$ depends on the dot product between $\boldsymbol{x}$ and $\boldsymbol{x_i}$ more than on their real value. The dot product can be appreciated as the similarity of two elements. The kernel function represent this similarity in $\mathcal{H}$, and we have :

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x_i}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x_i}) \rangle \tag{4.13}$$

with $\Phi : \mathcal{X} \to \mathcal{H}$ the mapping function.

Therefore, Eq (4.13) can be substituted in Eq (4.10) to obtain the SVM classification decision function :

$$\hat{y} = f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{4.14}$$

Many kernels are known. For a classification problem, one has to find the appropriate one which will define the right type of feature space, with the right parameters. In this work, we consider two kernel functions, the linear one and the Gaussian one, also known as the Radial Basis Function kernel (RBF).

Using a linear kernel reduces to the case discussed for a linear SVM, the kernel function being the dot product :

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i) = \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle \tag{4.15}$$

The RBF kernel has the form :

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|}{2\sigma^2}\right) \tag{4.16}$$

where σ parameter is the standard deviation of the Gaussian bell. This kernel is known to give good results and allows the results to be interpreted as similarity between points. For a classification problem, one has to evaluate the best σ to avoid over or under fitting.

For this work, the LibSVM library written by Chih-Jen Li, and modified by Jordi Muñioz to run on Matlab [1].

4.3 SETUP

The data were computed according to Chap.3. From now on we will refer to the nadir and off-nadir as X^1 and X^2 respectively. In order to have comparable values of pixels and to allow LibSVM to run well, all pixels from both images were rescaled by dividing by the maximum value of the two images $maxvalue = max(X_1, X_2)$. In order to work within a controlled framework to run the algorithms, all the data used were taken from the GTs.

Both sets X_1 and X_2 were split in two subsets, a training set $X_n train$, and a test set $X_n test$, $n = [1, 2]$. The training sets represent the pool of data for the algorithm to compute the manifold alignment and to train the models. The test sets were used to assess the algorithm performance.

1. available at : http://gpds.uv.es/ jordi/soft.htm.

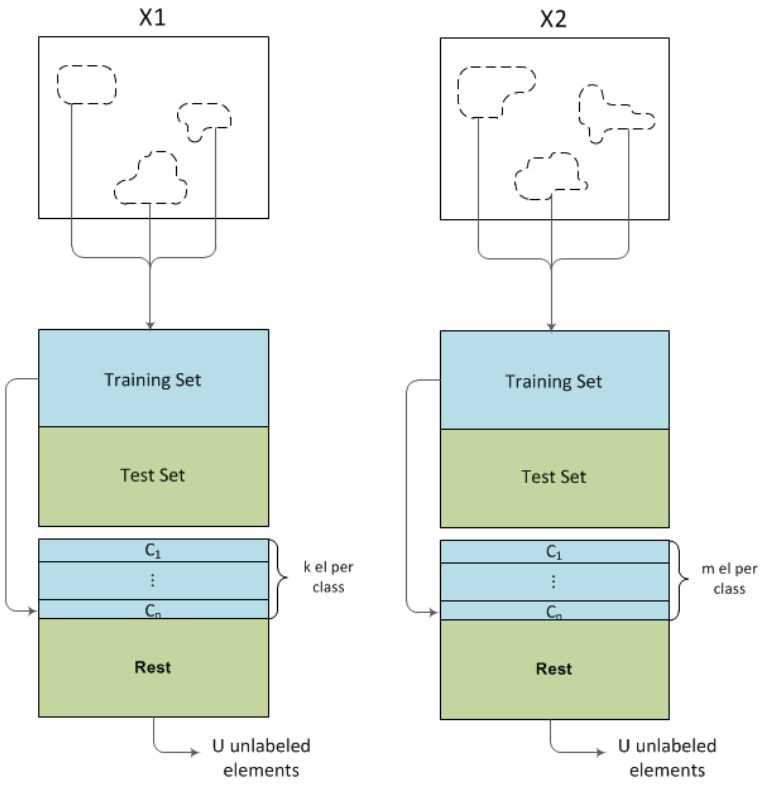


Figure 13: Workflow process of data preparation.

A problem arose on the consistency of the labeled data. The difference in the amount of pixels per class was of very high with varying orders of magnitude of 10^2 between the minimum and the maximum number of elements per class. If one were to use the data in this state, the algorithms would most certainly suffer from overrepresentation of the water class,leading to erroneous results. The proposed method was thus modified to account this fact. Considering the number of labeled samples per class, we consider the median number of elements med_N and impose the algorithm to randomly choose at most med_N per class to create our training set.

4.3.1 *Hypothesis*

As explained in Chap. 3.2.2, the MA technique used is a semisupervised method and thus needs both labeled and unlabeled elements from both domains. Remembering Chap.2, we know that the availability of labeled samples for multiple images set is a costly step in domain adaptation. The proposed approach of [3] does not pay any particular attention to the amount of labeled elements per domain to be taken to achieve a reasonable classification accuracy. In this work, we dispose of a fully computed GT for each image. To assess this dependency, we need to set the number of labeled element

per class in X_2 as a parameter of the global performance. A total of 150 points sample per class was taken for X_1 and the algorithm iterates over the number of labeled element per class in X_2. We call m this parameter to test.

$$m \in [10; 150] \tag{4.17}$$

In the same way, no information on the ratio between labeled and unlabeled element for each image was proposed. The relation is indirectly held in the cost function (eq. 3.15). The way the graph W_s and W_d were computed stated to assign a 0 value if one or both of instances' labels were unknown. Note that unlabeled samples only play a role in the geometrical similarity graph W. In order to evaluate the impact of the amount of unlabeled elements taken, we define it as an other parameter of performance analysis and call it U. As we only use data from the GT, the actual labels are known. The labeles corresponding to the U elements were set to 0.

$$U \in [500, 3000] \tag{4.18}$$

As said in Chap. 3.2.2, the μ parameter in the cost function acts as a weight parameter between topology preservation and matching instances. [3] holds that it should be equal to 1 in the case were both properties are equally important. Therefore, an attention is given to the variation of the parameter. During the computation of the cost function elements, we considered the magnitude of the different graph Laplacian computed [2].

We observed that the the magnitude of both L_s and L_d was way greater than the one of L. Remembering the derivation of the different Laplacians in Chap. 3.2.2, the Laplacian representing the geometrical similarity with a *k-nn* graph will have small values on its diagonal. On the other hand, as we choose 150 samples per class in X_1 and varying numbers for X_2, L_s will have values of the range of 10^2. L_d considering the unlabeled samples will have a magnitude of 10^3. Therefore, two cases are considered : $\mu = 1$ and $\mu = 100$ to evaluate the impact of this gaps of magnitude. As said, μ is a weight parameter and will not modify L in itself, but rescale its magnitude regarding L_d and L_s.

4.3.2 *Scenarios*

Four different scenarios were considered to evaluate the performance of the domain adaptation. We first consider ordinary supervised classification for both domain. Then we use the case of a single

2. L, L_s and L_d

ground truth computed over X_1 samples without adaptation to classify X_2. A supervised approach is then made by creating a model with labeled element from both domains. Finally we assess the performance of the manifold alignment method with regards to m and U.

For notation purpose and understandability, let's call $model_1$ the model built on X_1 elements, $model_2$ the model built on X_2 elements, $model_{1,2}$ the model built over elements of the two domains, and $model^p_{1,2}$, the model computed after the manifold alignment process. The different scenarios are :

A : Classification of X_1 with $model_1$

B : Classification of X_2 with $model_2$

C : Classification of X_2 with $model_{1,2}$

D : Classification of the projected data of X_2 with $model^p_{1,2}$

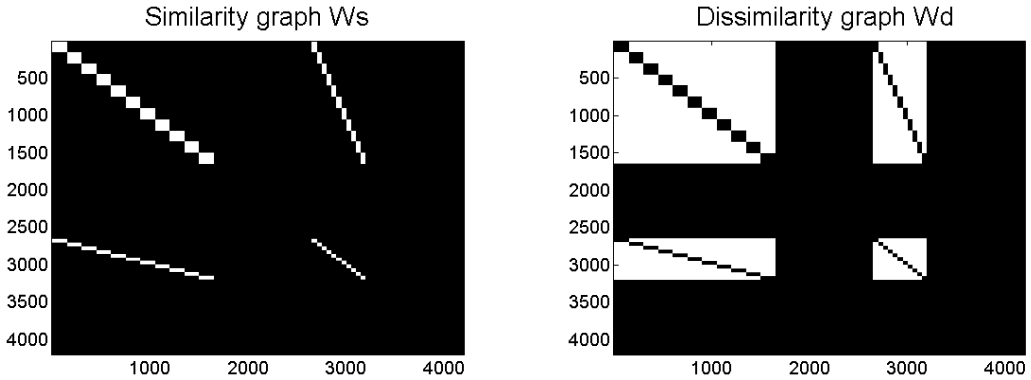
The four scenarios are tested on the different classifiers presented in Chap. 4.2 : the Naive Bayes classifier and the SVM using an RBF kernel. To Evaluate the performance, two measures are done; the overall accuracy (OA) of the classification and the Cohen's Kappa presented in [39] which measures the agreement of the estimation and the observation.

5

RESULTS

5.1 MATRICES INVOLVED

The geometrical similarity graphs were presented in Chap. 3. In Fig. 14, we present the similarity and dissimilarity graphs built on the labeled data. The graphs illustrates the symmetrical property of the similarity. The upper left part corresponds to the similarity within X_1, and the lower right the one for X_2. The two other blocs are the similarity between the two domains. Note that the size corresponding to X_2 grows with the increase of parameter m and U.



(a) (b)

Figure 14: Instance matching similarity graphs

In this figure, the white part means a value of 1 and the black a value of 0.

5.2 NUMERICAL PERFORMANCES

The performance for the two classifiers is shown as a function of parameter m in Fig. 15. The non-continuous lines are the results for the projected results for three different U : 1000, 2000 and 3000. The red line corresponds to scenario C, where no unlabeled elements are used. This scenario is the combination of the two GT without projection and is therefore the goal to overtake. The green line is the results for the classification of X_2 with its own ground truth. It is a best case scenario that we also want to overtake. In both cases, the results between OA and Kappa are very similar. As scenario A corresponds to the the case of scenario C with $m = 150$ for X_1, the result could not represented as function of m. The results coincided

with the the specific case of the scenario C, being the upper right point on the full red lines. We can see that the NB gives poor results for the use of two GT without manifold alignment. Its sensibility to the variation in PDF between two images does not allow it to appreciate the input of both images. The added labeled element from X_2 allow to increase its performance but at a low rate. Compared to the SMV method, the amount of sample from the second image needs to be greater in order to show an increase of accuracy. When using the manifold alignment, NB is able to reach the performance of a classification preformed on the single GT case with both a μ of 1 and 100. It also interesting to note that there is no increasing trend for scenario B with this classifier.

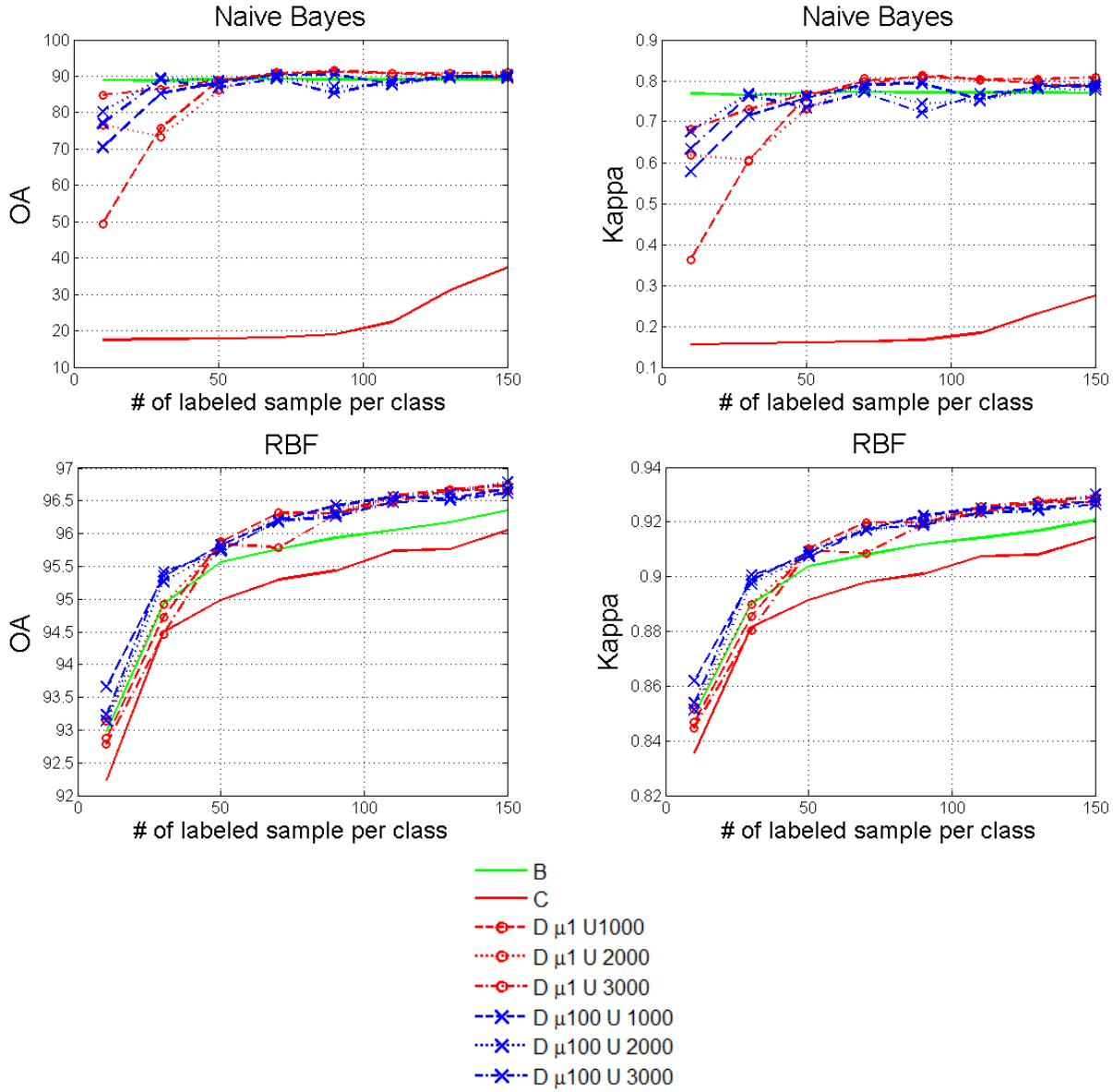


Figure 15: Performance for the two classifiers.

The RBF SVM shows better results for every scenarios showing its ability to adapt itself to the data's complexity, which is a clear advantage in comparison to NB. The gap between B and the different set for D is also smaller than what is seen for NB at small m. The main advantage of this classifier, when using MA, is that it is able to overpass scenario B with both μ parameters.

The DPC performance is shown in Fig. 16. The plots show the results for both corrected and uncorrected projection functions for the two μ analysed. In this case, the dimension of the latent space is set to d_{max}. Surprisingly, the assumption of increasing the performance by correcting the geometrical error in the projections is not validated for any classifier. The residual differences is of the order of 10^{-6} which clearly refutes the hypothesis.

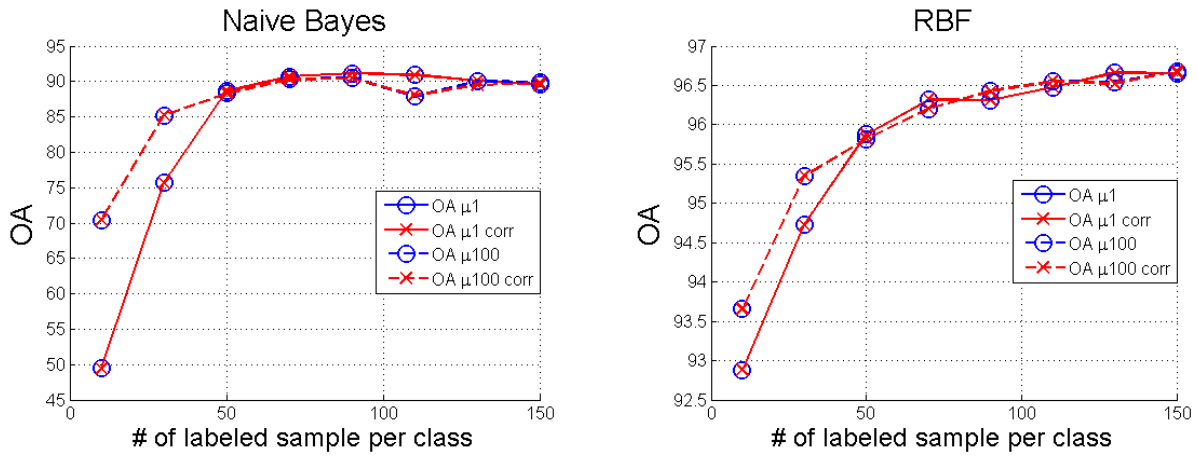


Figure 16: Sensibility of the different classifiers to the DPC

Remembering Fig. 9, the BDM was showing mismatched projections for the 8^{th} dimension forth in the uncorrected case. [3] did not give much information on the optimal dimension of $\mathcal{H}$. In the case of using d_{max} as the dimension of the latent space, we assumed to bestow the maximum variance of data for the classification. However, if one makes the parallel between the MA technique and a principal component analysis (PCA), we can assume that the first dimension of $\mathcal{H}$ should inherit the maximum of the variance explaining common information among the domains, both in geometrical and correspondences. This way, each new dimension should add knowledge but with a decreasing amount as the dimension of $\mathcal{H}$ increases, the last dimension should therefore contain the noise, e.g. the difference among domains. To assess this, we computed the performance of the classifiers for each single dimension of the latent space. This is shown in Fig. 17. This analysis gives very interesting results as the aforementioned hypthesis of the last dimensions hold noise is refuted. Indeed, the figures shows that the different dimension are not ordered in decreasingly manner regarding their single

performance. Further analysis should allow to better understand this property.

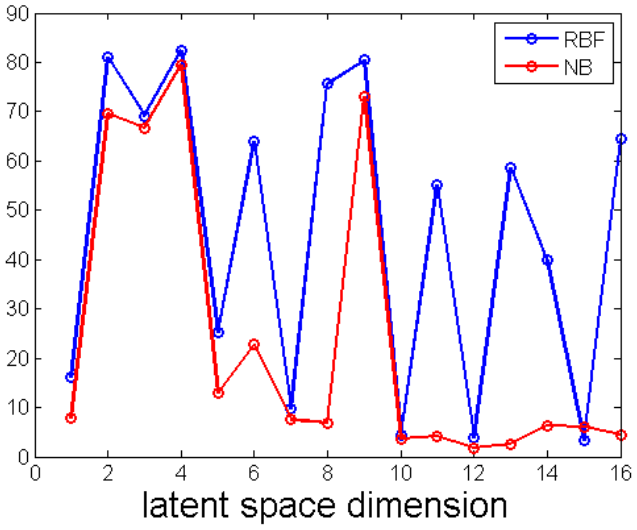


Figure 17: performance of classifiers for each dimension of $\mathcal{H}$ separaly

Beyond assessing the performance of each dimension one its own, we also considered the case of increasing the dimension of $\mathcal{H}$. This is represented Fig.18. The accuracy grows very fast until the 4^{th} dimension where it reaches a near maximum state for all three classifiers. Interestingly enough, the dimension where the performance reaches this asymptotic behaviour is smaller than the initial dimension of each domain. The source of this might be the initial property of manifold alignment being a dimensionality reduction problem. Anyhow, the BDM was still showing a possibility to improve the projection and might be more useful as a measure of the residual noise in the data found in the highest dimensions of the latent space.

5.3 PARAMETERS SENSIBILITY

5.3.1 *To parameter* μ

Fig. 15 also showed the role of μ in MA. For both classifiers at low m, the case when μ is scaled to the Laplacians magnitudes shows better results than the unscaled case. It is particularly interesting for the SVM as the results in this case are always overpassing the scenario B. In the case of very few labeled data available for X_2, this parameter could therefore play an valuable role.

In Fig. 19 μ shows once more some interesting results. For NB, increasing the value of μ seems to introduce noise in the results, but also shows better performance at low m. The graph is more chaotic and does not show a homogeneous evolution of the performance.

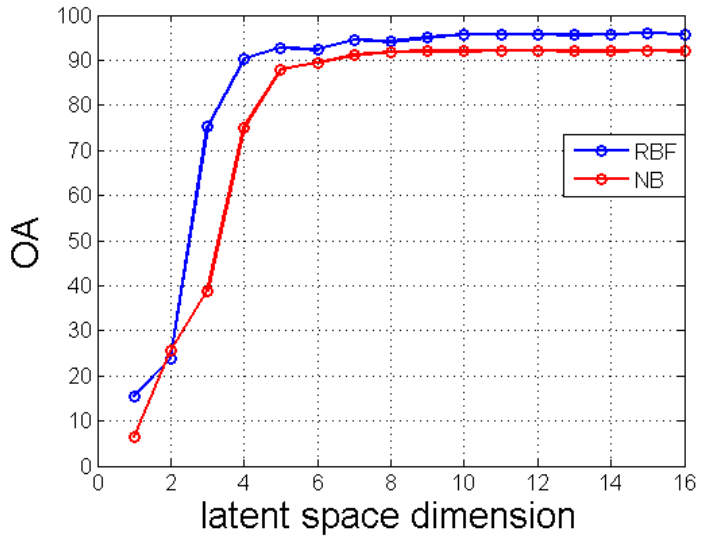


Figure 18: performance as function of the latent space's dimension

However, RBF does not suffer this heterogeneity. The rescaling step must therefore be made with precaution, with good consideration on the classifier used.

5.3.2 *To m and U*

The study of the performance as a function of m and U is presented in Fig.19. For each classifier, the dependency over the number of labeled sample from X_2 is clear. The increase is relatively fast for the first addition of samples but rapidly reaches a flatter state around $m = 50$ elements per class taken for X_2. From there on the progression of performance is relatively stable for NB. In the case of RBF SVM, it keeps getting a little higher. The ability of RBF to fit more complex data might be the explanation. Indeed, in the point of view of PDF matching, the more m increases, the more the system is constraint in order to fit the two images together. As the amount of samples increases, so does the size of both W_s and W_d and thus the complexity of the algorithm, which might explain why the RBF shows this light progression for bigger m.

The number of unlabeled samples show a different results. It seems only relevant for the NB at very low m. The SVMs seem to be relatively unaffected by this parameter. The spectral information needed for the preservation of topology has to be sufficiently represented by the labeled data used.

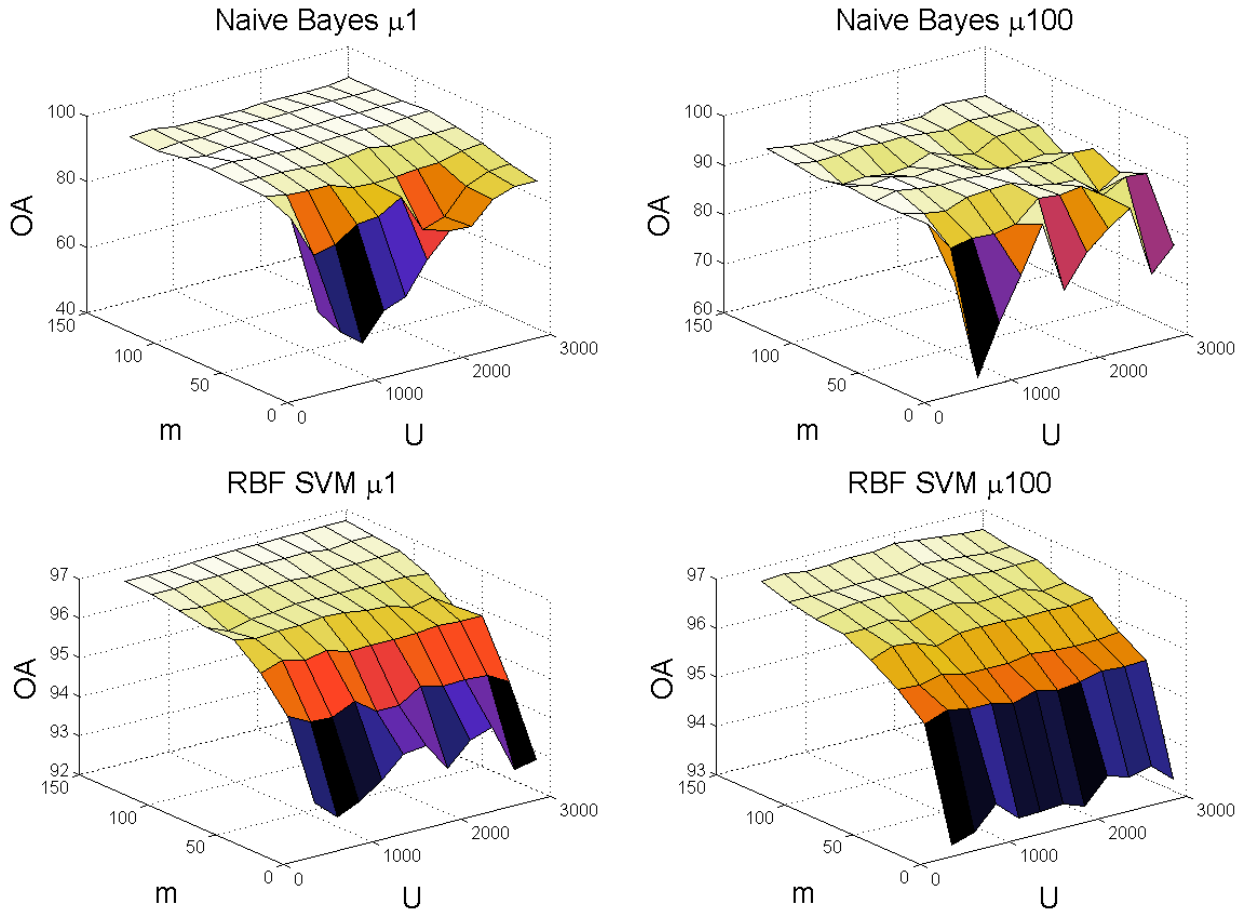


Figure 19: performance for the three classifiers Kappas on the left and Overall Accuracy (OA) on the right.

5.4 CLASSIFICATION MAPS

The results of classification are shown in Fig. 20. The previous observations are confirmed with respect to the classifiers' accuracy. The RBF offers more precise classified maps than the NB for each scenario. Indeed, NB seems to assign to much pixels to the class "Shadows" which is also sometimes mixed with water. The use of feature recognition algorithm might increase the results, as we can see that some of the missclassified pixels lie inside bigger structures.

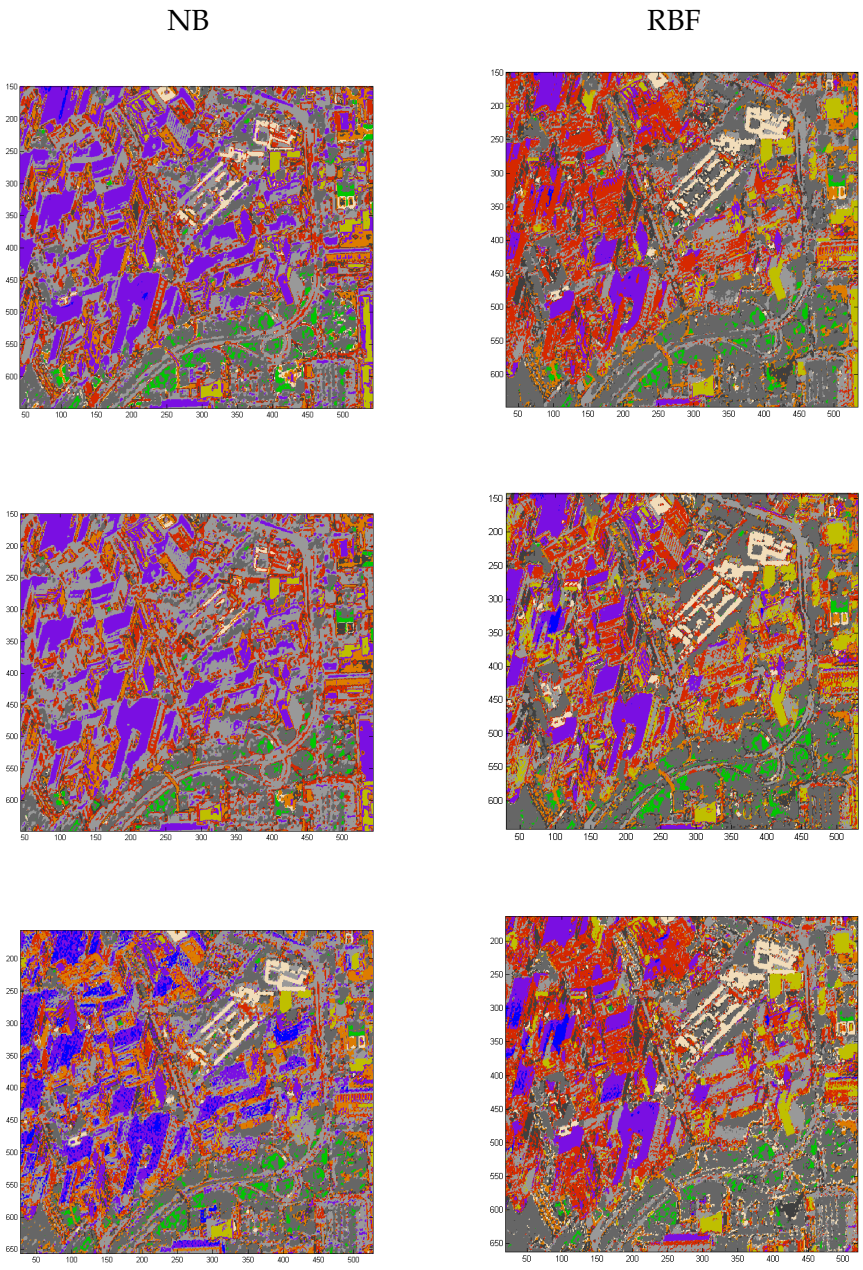


Figure 20: Classified maps of X_2. In a descending way we have scenarios B, C and D, all runned with $m = 50$.

5.5 DISCUSSION

The topological preservation with respect to matching instances seems to be well respected. However, further analysis should be made to be sure of this. We could think of taking less labeled points in X_1, or having varying amount of labeled data for both domains. One should also remember that the two images were taken from the same satellite at a very short time shift, making the spectral variance

between the two images somehow the smallest possible. Further analysis such as a multi temporal problem or matching images from different sensors are ought to show different results by using the MA technique. In such analysis the dimension of the latent space at which the performance becomes asymptotic might also change.

The link between the p_k input dimensionalities and the optimal d cannot be shown on a two images dataset, and opens the door to an important quantity of further analysis on the proposed MA technique.

6

CONCLUSION

The approach considered in this work proposed to use manifold alignment for domain adaptation between two images from a multi-angular dataset one considered as the source domain, and the second as the target domain. The algorithm allowed to project both domain in a common high dimensional latent space were two classifiers were trained, a Naive Bayes and a support vector machine. The results were compared to corresponding analysis in the initial domains. Different parameters were tested to determine the key elements of the method : the amount of labeled data available for the target domain, the amount of unlabeled data taken in both images and a parameter acting as a weight between the preservation of topology and the matching of instances.

For both classifiers the method allowed to improve classification accuracy with regards to conventional approaches. The use of few labeled samples from the target domain was sufficient to achieve this improvement. The amount of unlabeled data used was mostly interesting for few labeled data. However, unlabeled data only holds spectral information whereas labeled data contains both spectral and correspondence information. As the two images were acquired by the same sensor in a very small time shift, the sensitivity of the approach with respect to these two kind of information could not be properly evaluated. This opens the field to further analysis, such as multi-temporal approach or the use of multi-sensor datasets. The approach also showed an appreciation of the latent space's dimension. However, results could not define an optimal case and this element still needs to be studied for a better understanding of manifold alignment performance in domain adaptation.

A

ANNEXES

A.1 DOT PRODUCT CORRECTION GRAPHS

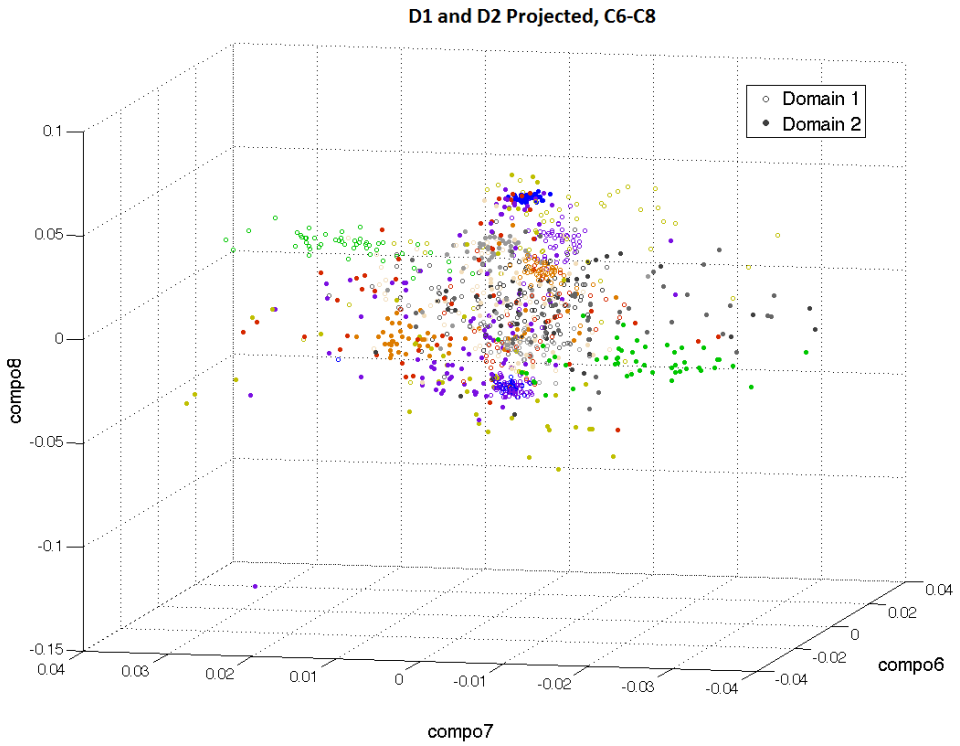


Figure 21: Projection of the two domains on the dimension 6-8 without correction

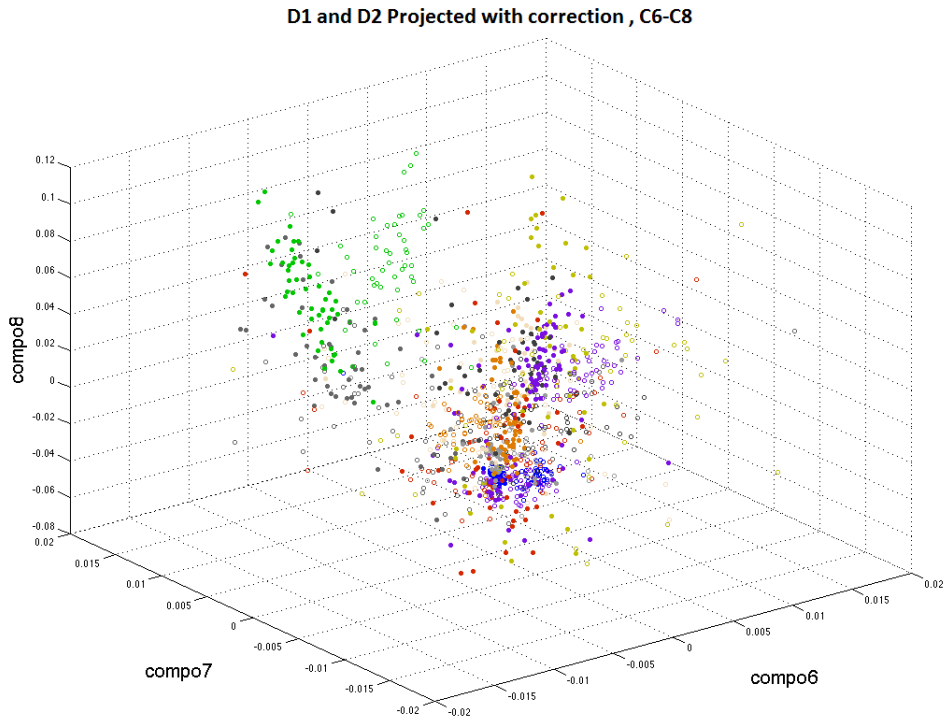


Figure 22: Projection of the two domains on the dimension 6-8 with correction

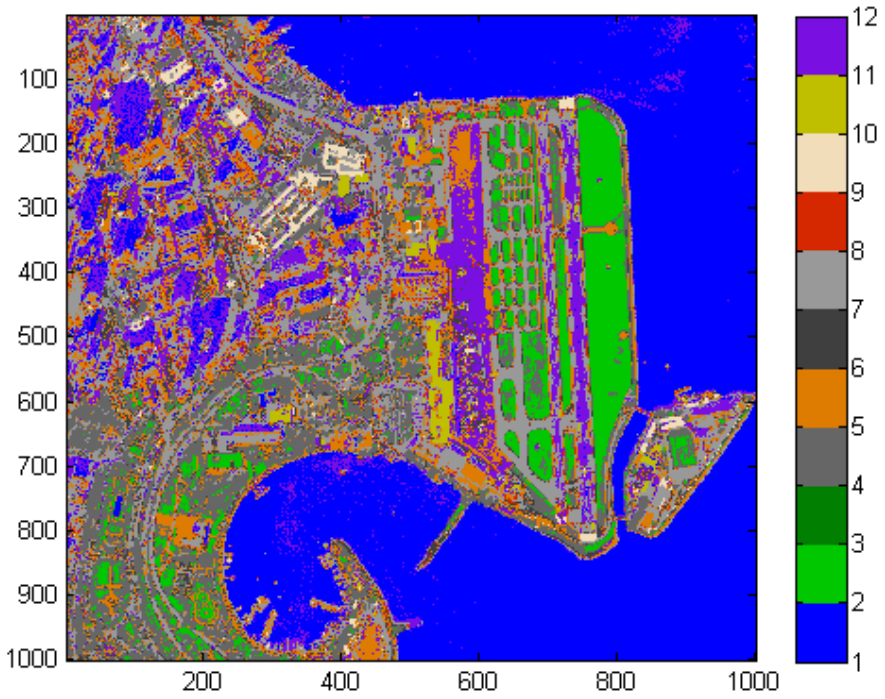


Figure 23: Classification of whole X_2 with NB, scenario D

A.2 CLASSIFICATION OF THE TWO IMAGES

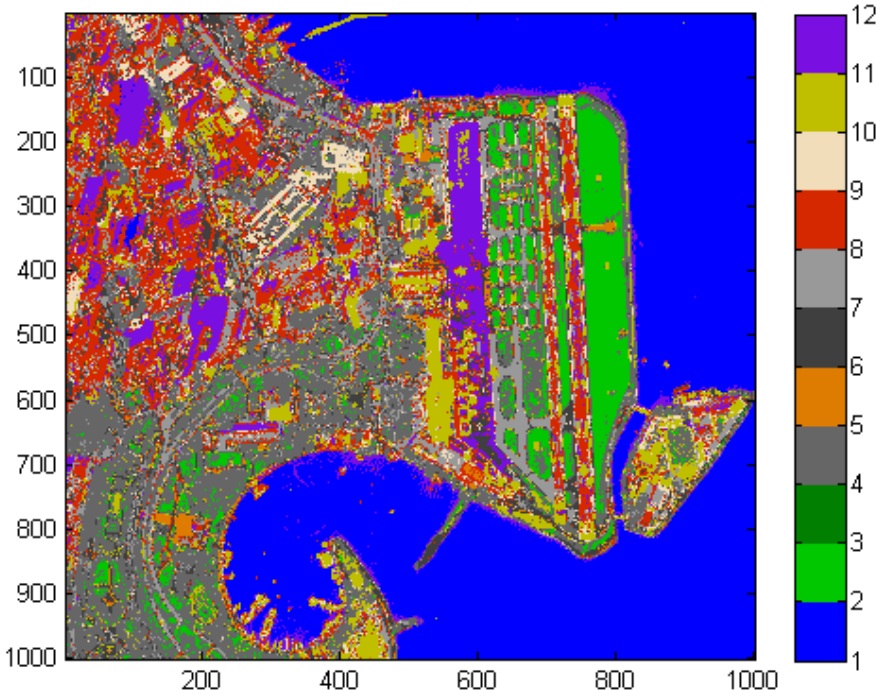


Figure 24: Classification of whole X_2 with RBF, scenario D

BIBLIOGRAPHY

[1] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford3. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[2] Chang Wang, Peter Krafft, and Sridhar Mahadevan. *Manifold Learning Theory and Applications*, chapter Manifold Alignment. CRC Press Inc, 2011.

[3] Chang Wang and Sidhar Mahadevan. Heterogenous domain adaptation using manifold alignment. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2:1541–1546, 2011.

[4] Fabio Pacifici, Jocelyn Chanussot, and Qian Du. 2011 grss data fusion contest : Exploiting worldview-2 multi-angular acquisitions. In *IGARSS 2011*, 2011.

[5] Clement Chatelain. Les support vector machine (svm). Technical report, 2003.

[6] Gong Jianya, Sui Haigang, Ma Guorui, and Zhou Qiming. A review of multi-temporal remote sensing data change detection algorithms. *ISPRS Congress Beijing*, 27:757–762, 2008.

[7] Nathan Longbotham, Chuck Chaapel, Laurence Bleiler, Chris Padwick, William J. Emery, and Fabio Pacifici. Very high resolution multiangle urban classification analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 50:1155–1170, 2012.

[8] David J. Diner, Gregory P. Asner, Roger Davies, Yuri Knyazikhin, Jan-Peter Muller, Anne W. Nolin, Bernard Pinty, Crystal B. Schaaf, and Julienne Stroeve. New direction in earth observing scientific application of multiangle remote sensing. *Bulletin of the American Meteorological Society*, 1999.

[9] Sicong Liu and Peijun Du. Multi-angular satellite remote sensing and forest inventory data for carbon stock and sink capacity in the eastern united stats forest ecosystems. *ISPRS Journal of Photogrammetry & Remote Sensing*, 2004.

[10] Stavros Stagakis, Nikos Markos, Olga Sykioti, and Aris Kyparissis. Monitoring canopy biophysical and biochemical parameters in ecosystem scale using satellite hyperspectral imageery : An application on a phlomis fruiticosa mediterranean ecosystem using multiangular CHRIS/PROBA obersvations. *Remote Sensing of the Environment*, 114:977–994, 2010.

[11] Riccardo Duca and Fabio Del Frate. Hyperspectral and multiangle CHRIS/PROBA images for the generation of land cover

maps. *IEEE Transactions on Geoscience and Remote Sensing*, 46:2857–2866, 2008.

[12] Jochem Verrelst, Jan G. P. W. Clevers, and Michael E. Schaepman. Merging the minnaert-k parameter with spectral unmixing to map forest heterogeneity with chris/porba data. *IEEE Transactions on Geoscience and Remote Sensing*, 48:4014–4022, 2010.

[13] J. Verrelst, M.E. Schaepman, B. Koetzb, and J.G.P.W. Clevers. Spectrodirectional minnaert-k retrival using CHRIS/PROBA data. *Journal Canadien de teledetection*, pages 3016–3023, 2010.

[14] Farooq Ahmad. A review of remote sensing data change detection : Comparison of faisalabad and multan districts. *Jounral of Geography and Regional Planning*, 5:236–251, 2012.

[15] M. Pesaresi, K. Gutjahr, and E. Pagot. Moving targets velocity and direction estimation by using a single optical vhr satellite imagery. *International Journal of Remote Sensing*, 29:1221–1228, 2008.

[16] Hsien-Ting Chen and Hsuan Ren. Using multidimensional histogram equalization as relative radiometric calibration for change detection in remote sensing imagery. In *Asian Conference on Remote Sensing(ACRS)*, 2008.

[17] Zhengwei Yang and Rick Mueller. Unbiased histogram matching quality measure for optimal radiometric normalization. *ASPRS Annual Conference*, 2007.

[18] Shilpa Inamdar, Francesca Bovolo, Lorenzo Bruzzone, and Subhasis Chaudhuri. Multidimensional probability density function matching for preprocessing of multitemporal remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 46:1243–1252, 2008.

[19] Devis Tuia, Jordi Munoz-Mari, Luis Gomez-Chova, and Jesus Malo. Graph matching for adaptation in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–13, 2012.

[20] Suju Rajan, Joydeep Ghosh, and Melba M. Crawford. Exploiting class hierarchies for knowledge transfer in hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 44:3408–3417, 2006.

[21] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 11:487–493, 1998.

[22] Koji Tsuda, Motoaki Kawanabe, and Klaus Robert Muller. Clustering with the fisher score. *Advances in Neural Information Processing Systems*, 15:729–736, 2002.

[23] Lorenzo Bruzzone and Fernandez Prieto. Unsupervised retraining of a maximum likelyhood classifer for the analysis of multitemporal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 39:456–460, 2001.

[24] Luis Gomez-Chova, Gustavo Camps-Valls, Lorenzo Bruzzone, and Javier Calpe-Maravilla. Mean map kernel methods for semisupervised cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48:207–220, 2010.

[25] Claudio Persello and Lorenzo Bruzzone. A novel active learning strategy for domain adaptation in classification of remote sensing images. pages 3720–3723, 2011.

[26] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. 2007.

[27] Mikhail Belkin and Partha Niyogi. Towards a theoritical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74:1289–1308, 2008.

[28] Jihun Ham, Daniel Lee, and Lawrence Saul. Semisupervised alignment of manifolds. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 120–127, 2005.

[29] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[30] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395 – 416, 2007.

[31] Euisun Choi and Chulhee Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36:1703 –1709, 2003.

[32] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15:52–60, 1967.

[33] Congle Zhang. Web-scale classification with naive bayes. *Proceedings of the 18th international conference on World wide web (WWW)*, pages 1083–1084, 2009.

[34] Gustavo Camps-Valls and Lorenzo Bruzzone. *Kernel Mehtods for Remote Sensing Data Analysis*. Wiley, 2009.

[35] Christopher J.C Burges. A tutorial on support vector machine for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[36] Alexander Statnikov, Douglas Hardin, Isabelle Guyon, and Constantin F. Aliferis. Support vector machine without tears. *Bio-Medical and Health Informatics*, 2008.

[37] Simon Haykin. *Neural Networks, a comprehensive foundation*. Eastern Economy Edition, 2008.

[38] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition. *IEEE Transactions on Electronic Computers*, pages 326–334, 1965.

[39] Giles M. Foody. Thematic map comparison : Evaluating the statistical significance of differences in classification accuracy. *PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING*, 70:627–633, 2004.

[40] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, 2009.

[41] Zoubin Ghahramani. Unsupervised learning. Technical report, Gatsby Computational Neuroscience Unit, 2004.

[42] Bradley L. Whitehall and Stephen C-Y. Lu. Machine learning in engineering automation - the present and the futur. *Computer in Industry*, 17:91–100, 1991.