Kernel Learning for Structured Multi-View Data

BY

SEPIDEH ESMAEILPOUR CHARANDABI
B.Sc., Islamic Azad University of Tabriz, Iran, 2009

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2017

Chicago, Illinois

Defense Committee:

Philip S. Yu, Chair and Advisor
Besma Smida
Ashkan Sharabiani, Mechanical and Industrial Engineering

To my dear husband, Roohollah who has always been the source of unconditional love and encouragement for me. Of course without his incredible understanding and support it would not be easy to finish what I had started.

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor, Prof. Philip S. Yu, who despite the incredibly busy schedule accepted me as a master's student and guided me through my thesis. Many thanks to Dr. Lifang He, who guided me with her ideas and suggestions and supported me during this work. Thanks to my committee members, Prof. Smida and Prof. Sharabiani for their time and attendance.

My deepest appreciation goes to my father-in-law and mother-in-law, Rezayat and Fatemeh, without whose support and warm encouragement, I would not be here at UIC writing this thesis. At the end, I would like to thank my parents, Hassan and Nahideh, for their unconditional love and heartfelt prayers that paved the way of success for me.

SEC

TABLE OF CONTENTS

CHAPTER			PAGE
1	**INTRODUCTION**		1
	1.1	Thesis Organization	3
2	**BACKGROUND AND NOTATION**		5
	2.1	Notation	5
	2.2	Support Vector Machines	6
	2.2.1	Kernel methods for SVM problems	9
	2.3	Structured Data	10
	2.3.1	Generalization Bounds for Tensor Factorizations	11
	2.4	Multi-view Data	13
	2.4.1	Consensus Principle and Complementary Principle	14
	2.4.2	Multi-view Acquisition	15
	2.5	Multi-View Learning	15
	2.5.1	Co-training	16
	2.5.2	Subspace Learning	16
	2.5.3	Multiple Kernel Learning	16
	2.5.3.1	Multiple kernels for SVM Classifiers	17
	2.6	Tensor Analysis	19
	2.6.1	Basic Definitions	19
	2.6.2	Tensor Factorization	21
	2.6.2.1	Tucker Decomposition	21
	2.6.2.2	CP Decomposition	22
	2.6.3	Tensor based learning	22
3	**METHODOLOGY**		25
	3.1	Dual Structure Preserving Kernels (DuSK)	25
	3.2	DuSK$_{RBF}$ and DuSK$_{Polynomial}$	28
	3.3	Soft Margin Multiple Kernel Learning	29
	3.3.1	Solving the hinge loss SM-MKL (SM1MKL)	31
4	**EXPERIMENT EVALUATION**		34
	4.1	Experimnets	34
	4.1.1	Dusk in the framework of SM1MKL	34
	4.1.2	Dusk on the concatenated data	39
5	**SUMMARY AND CONCLUSION**		42

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

APPENDIX 43

CITED LITERATURE 45

VITA 49

LIST OF TABLES

TABLE		PAGE
I	COMPARED ACCURACY RESULTS ON BP DATA	38

LIST OF FIGURES

FIGURE		PAGE
1	Dual tensorial mapping	27
2	Proposed algorithm	35
3	Diagram of the proposed method	36
4	Diagram of the baseline method	39
5	Parameter sensitivity of compared methods	40

LIST OF ABBREVIATIONS

MKL	Multiple Kernel Learning
SM1MKL	Hinge loss Soft Margin Multiple Kernel Learning
SVM	Support Vector Machines
DuSK	Dual Structure Preserving Kernel
DuSK$_{RBF}$	Dusk kernel based on radial basis function
DuSK$_{Poly}$	Dusk kernel based on polynomial function
OPRS	Office for Protection of Research Subjects

SUMMARY

It frequently happens in machine learning problems that the information explaining the subject of interest to be obtained from different sources or modalities and many well-studied algorithms are proposed for multi-view problems. The choice of the algorithms however, should adapt to the particular properties of each multi-view problem.

Inspecting brain connectivity networks represented in brain images is a prevalent way to determine subjects affected by the neurological disorders. From a machine learning perspective, brain images and their corresponding class are labeled data instances which can be used for learning purposes. Medical images obtained from different imaging techniques are considered to be multi-view structured instances. Tensorial representation is the method we use in this study to preserve the natural structure of these images.

Since the connectivity information corresponding to each brain image is embedded in tensorial structure, it is highly important that the implemented learning algorithm to preserve the multi-way structure. we use a structure preserving kernel representation of the tensor instances for all views of the data which eventually is exploited in a multiple kernel learning algorithm that is an extension of support vector machines for multi-view data.

CHAPTER 1

INTRODUCTION

Multi-view sources have recently been a great subject of interest in classification problems for their capability of revealing different aspects of the same subject and accordingly increasing the accuracy of predictions. Multi-view data does not necessarily have to be from different domains as they could just be feature sets divided to different groups. Regardless of how it is generated, searching for effective algorithms that could make the most of provided information is an active area of research. Generally, three major methods are applied to solve multi-view learning problems depending on the requirements and characteristics of the specific problem. The co-training style algorithms are holding to consensus principle which implies the necessity of mutual agreement of views to improve the prediction results (Dasgupta et al., 2002). The second widely used approach is multiple kernel learning method that a variety of its formulations can be find in (Lanckriet et al., 2004), (Bach et al., 2004), (Cortes et al., 2009), (Kloft et al., 2011). This method is basically representing the multi-view data according to the various predefined kernel functions and searches for the best possible combination of predefined kernels that results in better predictions. The third method is subspace learning that assumes all views are generated from a shared subspace and searches for this hidden subspace; in other words this approach is based on finding minimal principal components representing multiple views. (Hotelling, 1936), (Akaho, 2006). One of the widely used multi-view sources for medical diagnosis is the brain images acquired by two main medical imaging techniques, functional Magnetic Resonance Imaging (fMRI) and Diffusion Tensor Imaging (DTI).These two methods are vastly used to explore the gray matter functions and white

matter microstructures in neuroscience. In addition; due to the close relationship between the nature of the information they represent, it is believed that educated integration of these two sources can facilitate the outcome of predictions, estimations or classification problems.

From a machine learning perspective, this requires an optimal algorithm for combination of multiple views. Moreover; it is highly important that the multi-way property of brain images that actually contains the inherent network structure to be preserved. fMRI and DTI instances can be regarded as multi-dimensional arrays or tensors with grayscale values for each voxel. Therefore, the problem is to efficiently represent the tensorial data while maintaining its structure and using an optimal multi-view learning approach.(He et al., 2014) proposes a structure preserving kernel for tensorial data that achieves higher classification accuracies compared to other conventional kernels that initially perform flattening on the data.

Moreover; Many of the works in the literature hold to the fact that the brain connectivity networks can naturally be modeled as graph representations and choose graph classification methods for the analysis (Richiardi et al., 2011), (Cao et al., 2015).

A motivation for the presented work in this thesis is that brain network connectivities can also be modeled as tensors while remaining the original data structure. Afterwards; a set of experiments are conducted to explore the preferred multi-view learning algorithms on fMRI and DTI sources.

The main contribution of this work is to explore whether multiple kernel learning method in conjunction with an structure preserving kernel can improve the accuracy results for classification of multi-view structured data.

1.1 Thesis Organization

This thesis is organized into 5 chapters. Descriptions of each chapter are as follows:

Chapter 1 First chapter introduces the problem in hand and gives a brief explanation about the motivation for the current study and the specific approach we chose to solve the problem.

Chapter 2 This chapter provides the reader with the basic knowledge about SVM theory, sufficient representation for structured data and common approaches to deal with multi-view problems. Tensor analysis and factorization methods are also being reviewed along with some conventional tensor based learning methods. The frequently used notation is also included at the beginning.

Chapter 3 The implemented approach is explained in this chapter. First, the formulation for DuSK$_{Poly}$ is derived to be used along with DuSK$_{RBF}$ as the basekernels of MKL algorithm. Second, the SM1MKL algorithm as a preferred MKL approach is explained to deal with multi-view structured data.

Chapter 4 The implemented experiments and their corresponding results are included in this chapter to evaluate the efficiency of proposed method compared to a baseline method on a given multi-view brain network data.

Chapter 5 The conclusion out of the conducted experiments along with some cases for future study is summarized in this chapter.

CHAPTER 2

BACKGROUND AND NOTATION

2.1 Notation

In this section we list the variables that we will keep referring to in the following chapters:

$\mathbf{a}, \mathbf{b}, \mathbf{c}, ...$	Boldface lowercases show vectors
$\mathbf{A}, \mathbf{B}, \mathbf{C}, ...$	Boldface uppercases show matrices
$\mathscr{A}, \mathscr{B}, \mathscr{C}, ...$	Calligraphic uppercases show sets
$\mathcal{A}, \mathcal{B}, \mathcal{C}, ...$	Euler script uppercases show tensors
$\mathcal{A}^{(m)}$	shows m'th mode matricization of A
$\langle a, b \rangle$	shows inner product of vectors a and b
$a \otimes b$	shows outer product between a and b
$\mathcal{A} \times_m \mathcal{B}$	shows m-mode tensor product between $\mathcal{A}$ and $\mathcal{B}$

Before presenting our methodology, in this chapter we first give some basic definitions in machine learning area that will be used as the foundation of this study and then we briefly review some works mostly related to multi-view learning on structured data.

2.2 Support Vector Machines

Support vector machines as the baseline of learning methods that we use in this study and are briefly reviewed in this section (Murphy, 2012).

Among all other machine learning algorithms for classification, SVM's or support vector machines have proved to yield considerably good results in practice and therefore have been an interesting subject of study. The main idea behind the SVM learning theory is to find the maximum separating margin of different classes. Maximizing the margin is important in the sense that in case some interference was added to data samples, the algorithm would still be able to classify the samples correctly. Maximizing the margin is equivalent to solve the following problem:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}+b) \geq 1 \tag{2.1}$$

Where $\|\mathbf{w}\|$ defines the separating hyperplane and the constraint assures that all data samples lie on the correct side of the boundary while the margin is maximized. The Lagrangian of this minimization problem is given by:

$$\mathscr{L}(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}+b) - 1) \tag{2.2}$$

Normally solving the dual from of the 2.2 is easier than solving the primal form and when *Kurush-Kuhn-Tucker* conditions are met dual form can be solved instead of primal problem and is given by:

$$\max_{\alpha \geq 0}(\min_{\mathbf{w},b} \mathscr{L}(\mathbf{w}, b, \alpha)) \tag{2.3}$$

This is a quadratic optimization problem that is solved by quadratic programming. $\mathbf{w}$ is the weight vector corresponding to the separating hyperplane, b is the bias value and α is the vector of coefficients for all data points. These set of coefficients are shown to be zero for the points away from the margin and usually positive for those on the margin of each class. The data points with positive α_i values actually determine the margin and are called support vectors.

$$\begin{aligned} &\max_{\alpha}(\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle \mathbf{x^{(i)}}, \mathbf{x^{(j)}} \rangle) \\ &\text{subject to} \qquad \sum_{i=1}^{m} y^{(i)} \alpha_i = 0, \quad \alpha_i \geq 0 \end{aligned} \tag{2.4}$$

$\mathbf{w}$, α and b are obtained by solving 2.4, a decision function is derived that can classify any given point $\mathbf{x}$ according to the sgn function $h(\mathbf{x})$:

$$h(\mathbf{x}) = sgn(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x_i} \cdot \mathbf{x}) + b) \tag{2.5}$$

Where negative $h(\mathbf{x})$ corresponds to the negative class and positive $h(\mathbf{x})$ corresponds to the positive class. Equations 2.4 and 2.5 clearly show that the pairwise inner product of all training samples and

also the pairwise inner product of all training and testing samples is required for learning and labeling respectively. However it is proven that the α_i's are zero for all other points except the support vectors and accordingly the computation cost decreases significantly.

The Dual objective function in 2.4 and accordingly the functional margin of SVM in 2.5 are derived based on the assumption that the samples belonged to each class are linearly separable. In other words, this formulation searches for hypothesis that can strictly separate the different classes with no errors. If the assumption is violated there will be no solution for 2.4. Since for the real world data it is inevitable to encounter with not linearly separable problems and to be able to use the desirable properties of SVM formulation for these cases, 2.1 is changed in a way that allows a budgeted amount of error for misclassified samples. The error is modeled by a slack variable called ξ_i for any given training sample. For those of samples which are in the correct side of the margin $\xi_i = 0$, for those inside the margin $0 \leq \xi_i \leq 1$ and the ones that are completely in the wrong side

$xi_i \geq 1$. To control the budget of error tolerance the trade-off parameter C is introduced that balances between complexity and the number of errors. The frequently used formulation for this case (Murphy, 2012) is called C-SVM and is given by :

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \quad \text{subject to} \quad \xi_i \geq 0 \quad , y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \xi, \tag{2.6}$$

A variant of C-SVM penalizes the errors according to the tolerable number of misclassified samples. The ratio of misclassified samples to all samples is represented by parameter ν where it holds that $C = \frac{1}{N\nu}$ (Chen et al., 2005).

2.2.1 Kernel methods for SVM problems

The SVM derivation in previous section assumes that data points are linearly separable in the feature space, in other words different classes can be separated according to the straight lines in the original space. However; the fact is that usually the real world data are not linearly separable in the original feature space. Therefore, the kernel methods were introduced to overcome this issue. The main idea behind kernel method is that there existis a higher dimensional feature space ,where if the data mapped to, can solve the non-separability problem. In addition to this desirable property, kernel methods are popular due to the fact that they decrease the computational complexity of learning when inner product of samples is required to be calculated. This property which is referred as kernel trick paves the way for learning algorithms since it does not need the detailed coordinates of the samples in a possibly infinite dimensional space. Instead, it can calculate the desired inner product in the higher dimensional space based on their coordinates in the raw feature space. Equation 2.7 illustrates how the kernel trick works:

$$h(\mathbf{x}) = sgn(\sum_{i=1}^{m} \alpha_i y_i (\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x})) + b) \tag{2.7}$$

$$K(x_i, x) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x})$$

where $\phi(\mathbf{x_i})$ and $\phi(\mathbf{x})$ are the mapped samples and $k(x_i, x)$ is the corresponding kernel function. Radial basis function (RBF) kernel and Polynomial kernel are two frequently applied kernels due to their good performance on a wide range of datasets. For given sample $\mathbf{x}$ and $\mathbf{y}$, RBF and Polynomial kernels are defined as follows:

$$K_{RBF} = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}) \tag{2.8}$$

$$K_{Polynomial} = (\mathbf{x}\cdot\mathbf{y}+c)^d \tag{2.9}$$

The bandwidth parameter of RBF kernel (σ) and the degree parameter of polynomial kernel are adjusted according to the requirements of the learning problem.

2.3 Structured Data

According to the prolific advances in data acquisition technologies, the representation methods of gathered data is grown to be a matter of great importance. Arbitrary types of features or attributes explain the data instances that may not necessarily be related to each other; however, for some of the datasets the features are bonded together in different ways and a meaningful interpretation of the observed data cannot be acquired without taking the inherent structure into account which requires an expressive model of the data.

One of the most common methods to efficiently represent the structured data is to utilize various matrix factorization methods that use the tensor product as their common principle and make a compact representation of the data possible. (Srebro, 2004) comprehensively studied the matrix factorization models

as common tools of analyzing the machine learning problems such as matrix compression, matrix prediction or extracting the structure of a matrix. (Nickel and Tresp, 2013) expands the results to the higher order tensor case and conducts a set of experiments to analyze the consequences of unfit representation of tensorial data. For a given tensor of order 3 called $\mathcal{X}$, matrices A,B,C and the tensor $\mathbf{W}$ are called factorization matrices and the core tensor respectively if they can express $\mathcal{X}$ by a set of tensor products:

$$X = W \times_1 A \times_2 B \times_3 C \tag{2.10}$$

A,B,C and $\mathbf{W}$ are actually the elements of multi-linear or tucker3 decomposition explained in 2.6. Tucker3 is the true factorization of $\mathcal{X}$ since it can take all the dimensions of $\mathcal{X}$ into the account. However, other factorizations that assume higher or lower order for $\mathcal{X}$ are referred to be overstructured or understructured representations. For instance tucker4 which assumes fourth mode for the data is considered overstructured for $\mathcal{X}$. Overstructured representation is created when we assume unreal slicing along extra modes. Analogously, an understructured representation for X can be a SVD decomposition for matrixes where the structure along one mode of $\mathcal{X}$ is ignored by flattening methods.

2.3.1 Generalization Bounds for Tensor Factorizations

Tensor factorizations are a non dispensable part of tensor learning methods as they can give efficient representation of structured data. In the other hand, factorizations enforce constraints that may affect the generalization ability of a representation in many learning problems such as link prediction or item recommendation.

Given a tensor prediction problem with the zero-one loss function which is the commonly used function for classification purposes, for each element of $\mathcal{X}$ and the target labels of $y_i \in \{\pm 1\}$:

$$\text{Loss}(x_i, y_i) = \begin{cases} 0, & \text{if } sgn(x_i) = sgn(y_i) \\ 1, & \text{otherwise} \end{cases} \tag{2.11}$$

The value of loss function clearly just depends on the sign of x_i and not its magnitude. Therefore, the equivalence classes of tensors are those with the same sign patterns. The set of all possible sign pattern for a given tensor of arbitrary order is given by(Nickel and Tresp, 2013):

$$sgn(\mathbf{X}) \in \{-1, 0, 1\}^{\prod n} \tag{2.12}$$

Therefore, a constrained factorization of $\mathbf{X}$ such as Tucker decomposition actually limits the sign patterns in 2.12. This is the reason why the discrepancy of any future prediction $\mathbf{X}$ is bounded by the observed discrepancy between $\mathbf{X}$ and the target tensor $\mathbf{Y}$ plus another term(Nickel and Tresp, 2013):

$$\forall \mathbf{X} \in \mathscr{X}_{\mathbf{r}} : Prob(D(\mathbf{X}, \mathbf{Y}) \leq D_{\Omega}(\mathbf{X}, \mathbf{Y}) + \sqrt{\frac{\log |S_{n,r}| - \log \delta}{2|\Omega|}}) > 1 - \delta \tag{2.13}$$

$$S_{n,r} = \{sgn(\mathbf{X}) \in \{-1, 0, 1\}^{\prod n} | \mathbf{X} \in \mathbb{R}^{\prod n}, \text{n-rank}(\mathbf{X}) \leq \mathbf{r}\} \tag{2.14}$$

$$\text{where} \quad |S_{n,r}| \leq (\frac{4e(\text{ord}(\mathbf{X}) + 1)\text{pol}(\mathbf{X})}{\text{Var}(\mathbf{X})})^{\text{Var}(\mathbf{X})}$$

This is a Probably approximately correct(PAC) bound for discrepancy and the second term clearly shows that how the structure of a tensor, determined by $\text{ord}(\mathbf{X})$ and $\text{Pol}(\mathbf{X})$, and the factorization con-

straints, determined by var($\mathbf{X}$) affect the upper bound on the generalization ability of the model.

(Srebro, 2004) compares the generalization ability of overstructered and understructured factorizations of $\mathbf{X}$ with its true model factorization. The generalization ability of the overstructured model is entirely low compared to the true model and therefore, the overstructured model may seem not to be expressive enough but at the same time it is more persistent for different ratio's of missing data. On the other hand, the understructured model shows good generalization abilities compared to the true models for low amounts of missing data which is an evidence on its expressiveness but it degenerates when the missing data ratio increases.

The effects of factorization constraints are studied in (Srebro, 2004) and the results suggest that although due to the limited sign patterns of a model with more constraints, the overall classification results are worst than the true model, However; it also does not suffers from the overfitting problem like the true model with no constraints.

2.4 Multi-view Data

The main difference between the single-view and multi-view data is the availability of extra information that actually represents the data from other points of view. In many of the data analytic problems, there are often different domains that the data can be collected from. Although groups of collected data may be heterogeneous, they are actually representing the subjects of the same source. Therefore, it is important to be able to sufficiently exploit and combine all views to improve prediction or classification performance. For example, the information used for medical diagnosis may be acquired by medical imaging techniques along with some clinical reports about subjects. To obtain better classification performance an efficient integration of these information is necessary.

A multi-view algorithm is naturally expected to perform better compared to a single-view algorithm; however, if the learning method cannot adapt to the multi-view data, the performance measures may even get worse. There are two main principles to assure the compatibility of the algorithm to the multi-view data. These principles are as follows

2.4.1 Consensus Principle and Complementary Principle

Assuming two different views for the dataset X that are shown by X^1,X^2 and the label Y associated with X, the consensus principle maximizes the acknowledgement between multiple views.Under some assumptions (Dasgupta et al., 2002)the relation between the probability of error for single view and multi-view acknowledgement is proved to be as follows:

$$P(f^1 \neq f^2) \geq \max\{P_{err}(f^1), P_{err}(f^2)\} \tag{2.15}$$

2.15 shows that the probability of disagreement upper bounds the probability of error for both of the single-view classifiers or equivalently it is inferred that minimizing the disagreement for multi-view problem also minimizes the error rate of both single-view hypothesis. Many algorithms such as co-training or co-regularization algorithms formulate their objective functions holding to this principle.
The complementary principle assumes each view containing supportive or complementary information that is used to boost the performance of the other views. For example, the predicted labels of a hypothesis learned on one view can be used as the labeled training set of other views which is indeed a method to exchange useful information contained in only one view (Nigam and Ghani, 2000). Different views of the data can even be generated by differently set the configurations of the same hypothesis such as

its bias and (Wang and Zhou, 2007) shows that when the diversity of these different configurations is greater than each learner's error, better performance can be expected.

2.4.2 Multi-view Acquisition

Sometimes different views are gathered directly from different sources describing the same data and the adaptation analysis is required to prove their adequacy for multi-view learning. However; sometimes the multiple views are generated manually according to a given feature space. One way of generating multiple views is dividing the set of features to as subsets of features. If the data samples reside in an n dimensional feature space, a total of 2^n possible feature subspaces can be generated that can be considered as different views of the data. This method is called random subspace method (Ho, 1998) and employs the bootstrapping and aggregation approach to choose the feature subsets. There are also other methods that try to construct a low-dimensional feature space representing both views by taking advantage of the complementary principle.(Sigal et al., 2009). Also to evaluate the proficiency of the constructed views, various methods are proposed to analyze and predict their adequacy.(Muslea et al., 2002)

2.5 Multi-View Learning

In section 2.3 different methods of gathering or even generating the multi-view versions of the data were explained. however; it is equivalently important that the algorithms used for learning on the multi-view data can adapt to the multi-view characteristics to be able to extract the underlying distribution or pattern of the represented data. Various multi-view learning algorithms are explained in this chapter. In particular, the multiple kernel learning method as a promising approach in multi-view problems will be studied.

A variant of approaches are used to deal with multi-view data which could mainly be categorized ito three groups.

2.5.1 Co-training

These kind of algorithms first learn the classifier on each view similar to the learning on single-view data and then according to the consensus principle the disagreement between multiple views is forced to be minimized by possibly propagating back the inconsistency. In other words, each learned hypothesis according to the labeled data, is first used to predict the labels on a new unlabeled set of data called validation set and the prediction results of this hypothesis is fed back to the other hypothesis as training set and therefore, each hypothesis uses the information in other views to improve its accuracy. A formulated version of co-training method is co-regularization which induces the agreement on unlabeled data for all hypothesis and minimizes the loss resulting form misclassification of the labeled data.

2.5.2 Subspace Learning

Despite the Co-training method that postpones the combination procedure of learners to the last stage of multi-view learning, in subspace learning method, the first step is to define an expressive subspace that is assumed to be the foundation for all produced views. One of the most common learning methods that falls into this category is Canonical correlation analysis (CCA) which is considered to be the multi-view version of principle component analysis(PCA).

2.5.3 Multiple Kernel Learning

Kernel methods are considered as powerful tools to overcome the nonlinearity problem of the real world data by benefiting from kernel trick that highly decreases the computational complexity of kernel matrix calculation.

kernels may be calculated according to the data samples in the original feature space where each kernel value represents the measure of similarity between corresponding mapped data points in the new space called Hilbert space. Every kernel matrix is characterized by its generating kernel function. In the case of multi-view data, a kernel matrix is generated according to a particular kernel function for each view. The second step is to combine the calculated kernels to be used in one of the common kernel-based learning methods. Therefore, the classifier is neither learned separately for each view like co-training method nor is learned on a shared representation like subspace learning method. Multiple kernel learning method can be considered as an in-between approach. The transitional stage of combining the calculated kernel matrixes as a single kernel can be done as a linear or non-linear combination; however, the performance results of non-linear combination were shown to be inconsistent (Varma and Babu, 2009) and the linear kernel combination is desired in terms of performance.

2.5.3.1 Multiple kernels for SVM Classifiers

Different kernels are representing various aspects of the data and not all of these features are necessarily relevant or useful for learning purpose. For example, some of the features may only be the result of some noise or interference in the data. To reduce the result of irrelevant kernels, a set of weights associated with each kernel are introduced that can be considered as the importance measure of a given kernel. A primitive way of combining a set of predefined kernels is through calculating the average kernel that basically sums a set of equally weighted kernels:

$$K(x_i, x_j) = \frac{1}{M} \sum_{k=1}^{M} K_k(x_i, x_j) \tag{2.16}$$

Average kernel method has shown(Cortes et al., 2009) to obtain better performance results on some applications compared to the more sophisticated methods like $L1$MKL method; however, it is not ideal when a lot of noisy features are included in the data.

To be able to efficiently control the effect of each basekernel in MKL algorithm, a set of L_p norm constraints may be enforced on their importance coefficients. Generally L_1MKL method is known for its sparsity property because the kernel with minimum objective takes all the weight. For this reason L_1MKL is sometimes referred as the hard margin MKL. MKL-SVM tries to simultaneously learn the optimal kernel combination and α coefficients of the SVM classifier.The dual objective function of a SVM classifier for L_1 constrained MKL is given by:

$$\max_{\alpha\in\mathscr{A}} \min_{\mu\in\mathscr{M}} \sum_{m=1}^{M} \mu_m \left(\frac{-1}{2}(\alpha\odot\mathbf{y})^T \mathbf{K_m}(\alpha\odot\mathbf{y})\right) \tag{2.17}$$

$$\text{where} \quad \begin{cases} \mathscr{A} = \{\alpha | \alpha^T\mathbf{1} = 1, \alpha^T\mathbf{y} = 0, \mathbf{0} \le \alpha \le C\mathbf{1}\} \\ \mathscr{M} = \{\mu | \mathbf{0} \le \mu, \sum_{m=1}^{M} \mu_m = 1\} \end{cases}$$

It can be seen in 2.17 that the constraints on the kernel weights for L_1MKL are just simplex constraints that eventually lead to sparsity. The difference of L_1MKL with L_pMKL for $p \ge 2$ is that the kernel coefficient constraint term in 2.17 is replaced by L_p norm ball constraint:

$$\mu \in \{\mu | \mathbf{0} \le \mu, \sum_{m=1}^{M} \mu_m^P \le 1\} \tag{2.18}$$

Due to the 2.18 the kernel importance coefficients of L_pMKL will not be zero for any of the basekernels in SVM problem which leads to diversity and is the main disadvantage of L_pMKL while irrelevant

kernels are included in the kernel set.

Considering the shortages of L_1MKL and L_pMKL methods, an intermediate approach with the capability of choosing the optimal kernels without including the irrelevant ones is desired. The next section explains the proposed method in (Xu et al., 2013b) to overcome this problem.

2.6 Tensor Analysis

Tensor representations for the real world data are proven to be an efficient way of preserving informative relations between data samples where neglecting these inherent structures results in non-optimal solutions for the learning algorithms. structural representation in the form of multi-dimensional arrays is specifically important when the learning process is done according to a small number of samples because learning algorithm needs to be fed the most informative representation of these scarce samples. The number of dimensions determines the order or mode of the tensor. Scalars, vectors and matrices are 0 order, first order and second order tensors respectively.

2.6.1 Basic Definitions

A set of essential definitions for tensor analysis are reviewed in this section that are explained comprehensively in (Kolda and Bader, 2009). These definitions are the principle for various tensor factorization methods.

Each element of a given M'th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ is specified with $x_{i_1,i_2,\ldots,i_M}$. The inner product of two same-sized tensors is, similar to the vector case, the summation of element-wise product between two tensors.

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=2}^{I_2} \ldots \sum_{i_M=1}^{I_M} x_{i_1,i_2,\ldots,i_M} y_{i_1,i_2,\ldots,i_M} \tag{2.19}$$

Outer product of given M'th order tensor $\mathcal{X}$ and N'th order tensor $\mathcal{Y}$, also called tensor product, is a tensor of order $M+N$ and is defined by:

$$(\mathcal{X} \otimes \mathcal{Y})_{i_1,i_2,\ldots,i_M,i'_1,i'_2,\ldots,i'_M} = x_{i_1,i_2,\ldots,i_M} y_{i'_1,i'_2,\ldots,i'_M} \tag{2.20}$$

Rank of a given tensor $\mathcal{X}$ is determined by the minimum number of rank one tensors that could exactly fit in $\mathcal{X}$ when summed together. A rank-one tensor of order M, similar to the matrix case, is a tensor that can be expressed as the tensor product of M vectors.

when tensor multiplication is required, given a high order tensor and a matrix, the problem is that there are different ways of multiplying tensor fibers with a given matrix, i.e. it is important to specify the mode that the the fibers are chosen from. To solve this issue the m-mode product is defined. For a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_m}$, the m-mode product is calculated as follows:

$$(\mathcal{X} \times_m \mathbf{A})_{x_{i_1,i_2,\ldots,i_{m-1},j,i_{m+1},\ldots,i_M}} = \sum_{i_m=1}^{I_M} x_{i_1,i_2,\ldots,i_M} a_{j,i_m} \tag{2.21}$$

m-mode product can equivalently be expressed by matricization of tensor $\mathcal{X}$ along mode m:

$$\mathcal{Y} = (\mathcal{X} \times_m \mathbf{A}) \Leftrightarrow \mathbf{Y}_m = (A)\mathbf{X}_m \tag{2.22}$$

Where the subscripts in the right term indicates the matricization along m th mode.

2.6.2 Tensor Factorization

Recently, tensor methods have been used in attempts to better understand the success of structured analysis. One class of broadly useful techniques within tensor methods are tensor decompositions. Considering the favorable properties of SVD decomposition for second order tensors (matrices) in extracting and ordering the basis components of a given matrix, higher order generalization of SVD decomposition or HOSVD is used for efficient representation of tensorial data. Two main tensor decompositions that are widely used in tensorial data analysis are Tucker decomposition (Tucker, 1963) and CANDECOMP/PARAFAC referred as CP (Carroll and Chang, 1970) (Harshman, 1970). CP can be considered as a special case of Tucker decomposition. The details and properties of both decompositions is given below.

2.6.2.1 Tucker Decomposition

Given a third order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the tucker decomposition is defined by factor matrices $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$ and the core tensor $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ where:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{p,q,r} \mathbf{a}_p \otimes \mathbf{b}_q \otimes \mathbf{c}_r = [\![\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{2.23}$$

An element indexed by x_{ijk} in tensor $\mathcal{X}$ is :

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr} \tag{2.24}$$

Tucker decomposition can also be computed for higher order tensors.

2.6.2.2 CP Decomposition

CP decomposition is a special case of Tucker decomposition where $P = Q = R$ and $x_{i_1 i_2 \ldots i_M} \neq 0$ only if $i_1 = i_2 = \ldots = i_M$. Given a third order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the CP decomposition is defined by factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$ where:

$$\mathcal{X} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r \tag{2.25}$$

An element indexed by x_{ijk} in tensor $\mathcal{X}$ is :

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \qquad \text{for all } \; i, j, k \tag{2.26}$$

2.6.3 Tensor based learning

Kernel methods are popular for their flexibility in alleviating the non separability of samples belonged to different classes. For data samples that reside in vector spaces, mapping to a higher dimensional space is done according to the conventional kernel functions that take sample feature vectors as their input and regardless of the description of mapped sample in high dimensional feature space, provide a similarity measure between all pairs of data samples which is fed to the classifier as data representation.

Although tensors are proven to efficiently capture the multi-way structure, since the suggested models for tensors are affine on the data (Signoretto, 2011) they are unable to cope with nonlinearities presented by real world data. Many studies are conducted to integrate the favorable properties of kernels to tensorial data; however, most of these methods inevitably need to perform flattening tasks that give

a vectorized or matricized representation of data(Signoretto et al., 2012)(Zhao et al., 2013a). However, all or parts of the structural information is lost in the cost of using the desirable properties of kernel methods.

Variations of tensor kernels are studied in (Signoretto, 2011) with their specific characteristics. The most naive type of kernels for tensorial data is the kernel that operates on vectorized version of tensor as follows:

$$k(\mathcal{X},\mathcal{Y}) = \langle \text{vec}(\mathcal{X}), \text{vec}(\mathcal{Y}) \rangle \tag{2.27}$$

Where for the well-known Gaussian-RBF kernel, it is given by:

$$k(\mathcal{X},\mathcal{Y}) = \prod_{p \in I_1 \times I_2 \times \ldots \times I_M} exp(-\frac{1}{2\sigma^2}(x_p - y_p)^2) \tag{2.28}$$

Regardless of data-type, this kernel neglects the structure along all dimensions which is referred as an invariance property.

Another type of structure exploiting kernels that is based on kernel products uses matrix unfoldings of a given tensor as inputs to each factor kernel.

$$k(\mathcal{X},\mathcal{Y}) = k^1(\mathcal{X},\mathcal{Y})k^2(\mathcal{X}\mathcal{Y})...k^M(\mathcal{X},\mathcal{Y}) \tag{2.29}$$

where $k^m(\mathcal{X},\mathcal{Y})$ is the kernel calculated according to the m'th mode tensor unfolding. Distance measure for product kernel with Gaussian-RBF factor kernels is different from Euclidean distance. It is concluded from 2.29 that the distance measure between $\mathcal{X}$ and $\mathcal{Y}$ for RBF kernel is given by:

$$d_T(\mathcal{X},\mathcal{Y}) = \sqrt{\sum_{m=1}^{M} d(\mathcal{X}_{(m)},\mathcal{Y}_{(m)})} \tag{2.30}$$

where $d(\mathcal{X}_{(m)},\mathcal{Y}_{(m)})$ is the distance between m'th mode unfolding of tensors $\mathcal{X}$ and $\mathcal{Y}$. Therefore, the product RBF kernel in 2.29 is:

$$k(\mathcal{X},\mathcal{Y}) = exp(-\frac{1}{2\sigma^2} d_T(\mathcal{X},\mathcal{Y})^2) \tag{2.31}$$

2.29 and 2.31 clearly show that flattening of the tensor is a non dispensable part of these methods.

CHAPTER 3

METHODOLOGY

The proposed algorithm in this study uses structure preserving kernels to represent multi-way data coming from different views. Dual structure preserving kernel or briefly DuSK is a specific variant of these kind of kernels that is previously proposed by (He et al., 2014) with an application of radial basis function DuSK to third dimensional brain images. We use two variants of DuSK, namely DuSK-RBF and DuSK-Polynomial, in the framework of a multiple kernel learning algorithm that has the ability of tuning the kernel weights according to their importance. Multiple kernel learning method that we choose for the multi-view problem is soft margin MKL that has some advantages over its counterpart formulations. In this chapter, derivation of DuSK kernels and solution to the soft margin MKL are briefly reviewed.

3.1 Dual Structure Preserving Kernels (DuSK)

A training set of tensorial data $\{\mathcal{X}_i\}_{i=1}^{l}$ and their corresponding binary labels $y_i = \pm 1$ are given. Similar to the derivation of non-linear SVM, where sample vectors are mapped to the higher dimensional space, this method generalizes vector mapping to tensor mapping in non-linear SVM formulation:

$$\min_{\mathscr{W}_F^2, b, \xi_i} \frac{1}{2}\|\mathscr{W}\| + C\sum_{i=1}^{l}\xi_i \tag{3.1}$$

$$s.t. \quad y_i(\langle \mathscr{W}, \phi(\mathcal{X}_i)\rangle) \geq 1-\xi_i \quad , \quad \xi_i \geq 0 \quad \forall i = 1,...,l$$

Using the Lagrangian method, the decision function for tensorial SVM is derived:

$$f(\mathcal{X}) = sign(\sum_{i=1}^{l} \alpha_i y_i k(\mathcal{X}, \mathcal{X}_i) + b) \tag{3.2}$$

It is shown in 3.2 that the tensorial SVM problem is only different from standard SVM by the kernel function $k(\mathcal{X}, \mathcal{X}_i)$ to be defined.

The specific method presented in (He et al., 2014) exploits CP factorization to express the data as a sum of low rank (rank one) tensors. CP is desirable in a sense that the factorization extracts the principle features of a tensor in the form of component vectors that are easier to operate on. For a factorized rank one tensor $\mathcal{X}$, the mapping ϕ is:

$$\phi : \prod_{m=1}^{M} \otimes \mathbf{x}^{(m)} \rightarrow \prod_{m=1}^{M} \otimes \phi(\mathbf{x}^{(m)}) \in \mathbb{R}^{H_1 \times H_2 \times \ldots \times H_M} \tag{3.3}$$

Therefore, the kernel function as an inner product in Hilbert space is given by:

$$k(\mathcal{X}, \mathcal{Y}) = \prod_{m=1}^{M} k(\mathbf{x}^{(m)}, \mathbf{y}^{(m)}) \tag{3.4}$$

Where 3.4 is resulted from the fact that the inner product of two factorized rank-one tensors is the multiplication of the inner products of corresponding modes. When the rank of the tensor is more than one, a summation is added to the formulation; however, since the Hilbert space is just a higher dimensional version of the original space, mapping of R summed tensors is equivalent to a summation between mapped factor tensors in CP. In other words, this kernel preserves the structure of tensorial data regardless of the

used mapping function. The dual tensorial mapping of a second order tensor is illustrated in *Figure* 1. For a M'th order tensor of rank R, the dual tensorial mapping is given by:

$$\phi : \sum_{r=1}^{R} \prod_{m=1}^{M} \otimes \mathbf{x}^{(m)} \rightarrow \sum_{r=1}^{R} \prod_{m=1}^{M} \otimes \phi(\mathbf{x}^{(m)}) \tag{3.5}$$

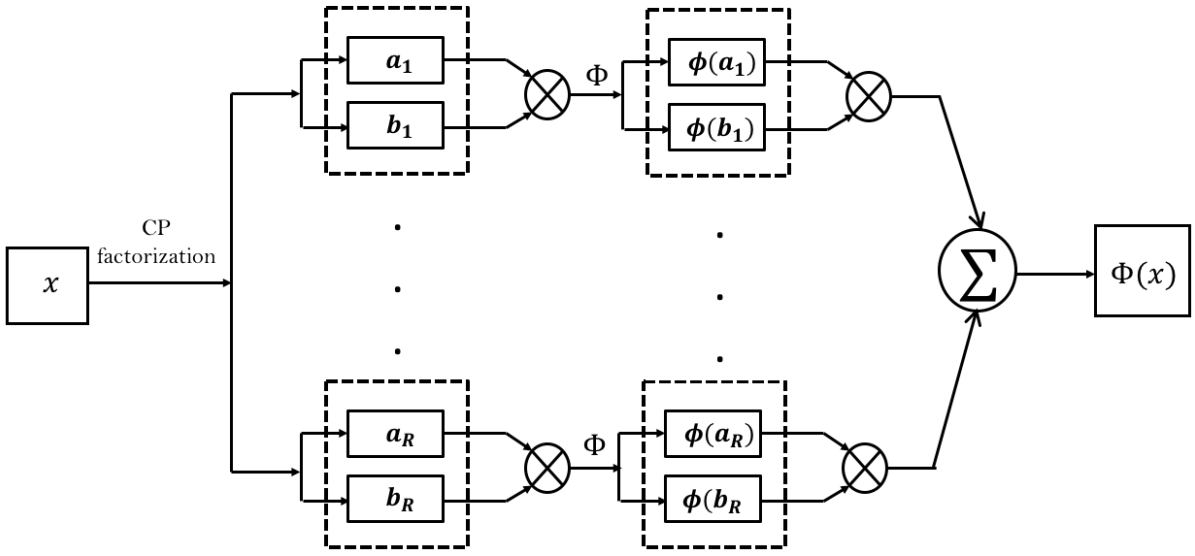


Figure 1: Dual tensorial mapping has applied to a 2D tensor of Rank R

Accordingly, DuSK kernel that is a measure of similarity between two structured samples of order M and rank R is derived as follows:

$$k(\sum_{r=1}^{R}\prod_{m=1}^{M}\otimes\mathbf{x}_r^{(m)},\sum_{r=1}^{R}\prod_{m=1}^{M}\otimes\mathbf{y}_r^{(m)})=\sum_{i=1}^{R}\sum_{j=1}^{R}\prod_{m=1}^{M}k(\mathbf{x}_i^{(m)},\mathbf{y}_j^{(m)}) \tag{3.6}$$

3.2 DuSK$_{RBF}$ and DuSK$_{Polynomial}$

The DuSK version of the well-known RBF kernel with parameter σ is formulated by:

$$k(\mathcal{X},\mathcal{Y})=\sum_{i=1}^{R}\sum_{j=1}^{R}exp(-\sigma\sum_{m=1}^{M}\|\mathbf{x}_i^{(m)}-\mathbf{y}_j^{(m)}\|^2) \tag{3.7}$$

As it is shown in 3.7, DuSK$_{RBF}$ is a summation between a set of product kernels.

Similar to the DuSK$_{RBF}$, polynomial DuSK is derived by choosing the core kernel function for 3.6 to be polynomial. The polynomial kernel function for two given vectors $\mathbf{x}$ and $\mathbf{y}$ is:

$$k(\mathbf{x},\mathbf{y})=(\mathbf{x}\cdot\mathbf{y}+c)^d \tag{3.8}$$

Where C is a constant value and d is called the degree of polynomial kernel. Therefore DuSK$_{Poly}$ is formulated as follows:

$$k(\mathcal{X},\mathcal{Y})=\sum_{i=1}^{R}\sum_{j=1}^{R}\prod_{m=1}^{M}(\mathbf{x}_i^{(m)}\cdot\mathbf{y}_j^{(m)}+c)^d \tag{3.9}$$

Now the kernel matrices for a given tensorial data instance can be calculated according to 3.7 and 3.9. A set of experiments conducted in (He et al., 2014) illustrates the preference of DuSK over other conventional kernels or those which initially use flattening methods. Normally there is no trivial way to determine the best combination of decomposition rank, kernel parameter and the trade-off parameter of SVM classifier; therefore, a grid search is preformed for a reasonable range of parameters. In multi-

view learning problems where information carried out by all views are considered equally important, the chosen learning methods should be capable of highlighting the beneficial features of each view. The effect of irrelevant or harmful features also needs to be attenuated.

3.3 Soft Margin Multiple Kernel Learning

C-SVM method tries to alleviate the problem of not linearly separable data by introducing the penalty parameter C that tries to balance the number of outliers and the complexity of the separating hyperplane. In other words, it adds a soft margin property to the standard SVM problem that allows a particular number of training errors determined by C. Soft margin multiple kernel learning or shortly SM-MKL that is studied extensively in (Xu et al., 2013b) is an approach that uses the notion of slack variable for SVM problem.

For a given dataset $S = \{(x_i, y_i)|i = 1, ..., l\}$, if we consider a set of M basekernels $K = \{\mathbf{K_1}, ..., \mathbf{K_M}\}$ generated according to S, the slack variable corresponding to each basekernel can be represented as:

$$\zeta_m = \tau - (\frac{-1}{2}(\alpha \odot \mathbf{y})^T \mathbf{K_m}(\alpha \odot \mathbf{y})) \tag{3.10}$$

It can be seen from 3.10 that ζ_m is the difference between target margin and the SVM dual objective for each kernel. Various loss functions can be defined for ζ where the most common loss function also used

for C-SVM is hinge-loss. Subsequently, the objective formulation for the soft margin multiple kernel learning algorithm with hinge-loss for kernel slack variables is defined as follows:

$$\min_{\tau,\alpha\in A,\zeta_m} -\tau + \theta \sum_{m=1}^{M} \zeta_m \tag{3.11}$$
$$\text{s.t.} \quad -1/2(\alpha \odot \mathbf{y})^T \mathbf{K_m} (\alpha \odot \mathbf{y}) \geq \tau - \zeta_m, \quad \zeta_m \geq 0, \quad m = 1, ..., M$$

Equation 3.11 clearly shows the role of parameter θ as the penalty for kernel slack variables defined in 3.10. It is proven (Xu et al., 2013b) that the solution to the problem in 3.11 is equivalent to solving the following problem:

$$\min_{\mu\in\mathscr{M}_1} \max_{\alpha\in A} -1/2 \sum_{m=1}^{M} \mu_m (\alpha \odot \mathbf{y}) \mathbf{K}_m (\alpha \odot \mathbf{y}) \tag{3.12}$$
$$\text{where:} \quad \mathscr{M}_1 = \{\mu | \mathbf{0} \leq \mu \leq \theta_1, \sum_{m=1}^{M} \mu_m = 1\}$$

By comparing 2.17 and 3.12 it is concluded that the only difference in formulation is the upper bound θ on the kernel combination coefficients for SM1MKL. In other words, this new formulation guarantees that each kernel is given a limited weight and therefore it is less likely that extreme cases like L_1MKL happens. The constraint in 3.12 ensures that θ could not be less than $1/M$ because the coefficients should sum to one. It can also be inferred that for $\theta = 1/M$ the same result as the average kernel is obtained. However; when $\theta \geq 1$ the new formulation for SM1MKL is the same as L_1MKL.

3.3.1 Solving the hinge loss SM-MKL (SM1MKL)

Although other losses can be defined for kernel slack variables, we stick to the hinge loss within our experiments. The Dual form of the objective function in 3.11 represents the role of the kernel slack variables ζ_m and the regularization parameter θ in the SM1MKL problem. However; the block-wise coordinate descent algorithm suggested in (Xu et al., 2013b) actually exploits the primal form that is convex on the objective function (Rakotomamonjy et al., 2008) and linear on the constraints and is given by:

$$\begin{aligned} &\min_{\mu\in M_1, f_m, b, \rho, \xi_i} \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|^2_{H_m}}{\mu_m} + C\sum_{i=1}^{l}\xi_i - \rho \\ &\text{s.t.}\;\; y_i\Big(\sum_{m=1}^{M} f_m(x_i) + b\Big) \geq \rho - \xi_i, \;\; \xi_i \geq 0 \end{aligned} \tag{3.13}$$

The idea behind a coordinate descent procedure is based on optimizing the objective function for a group of variables at a time. Starting from some initial values for a set of variables, the algorithm solves the problem for the other set of variables. Subsequently, the obtained values are substituted in the objective function and the problem is solved for the second set of variables. The procedure iterates between these two steps until convergence or until the maximum iteration limit is reached.

For the problem in 3.12 we are trying to optimize the kernel combination coefficients μ at one hand and to maximize the margin of a standard C-SVM problem by adjusting f_m, b, ρ, ξ_i variables on the other hand. Therefore, the procedure is as follows:

According to (Xu et al., 2013b) the algorithm begins by initializing equal kernel weights, i.e. $\mu = [1/M, ..., 1/M]$, and the resulting single kernel that is a summation between same-sized and weighted

kernels can be calculated. Afterwards, the problem reduces to a classic kernel SVM problem and can be solved as a quadratic programming problem to obtain α values of SVM classifier.

$$\max_{\alpha \in \mathscr{A}} -1/2 \sum_{m=1}^{M} \mu_m (\alpha \odot \mathbf{y}) \mathbf{K}_m (\alpha \odot \mathbf{y}) \tag{3.14}$$

After obtaining α values, the primal variables f_m, b, ρ, ξ_i are retrieved. Now with fixed values for f_m, b, ρ, ξ_i there are M fixed SVM values shown by a_m and the algorithm solves the following convex problem for μ_m:

$$\min_{\mu \in \mathscr{M}_1} \sum_{m=1}^{M} \frac{a_m}{\mu_m} \tag{3.15}$$

$$\text{where} \quad a_m = 1/2 \mu_m^2 (\alpha \odot \mathbf{y}) \mathbf{K}_m (\alpha \odot \mathbf{y})$$

Algorithm 1 SM1MKL Algorithm

1: Set initial values for kernel weigths $\mu^t = [1/M, 1/M, ..., 1/M]$ ▷ Initialize by equal weights

2: **while** Maximum iteration is not reached **do**

3: Obtain α^t by solving 3.14 using the Standard QP solver with μ^t

4: Calculate a_m and update $\mu^{\mathbf{t+1}}$ by solving the convex problem in 3.15

5: $t = t + 1$

6: **end while**

7: **return** Prediction results

We suppose all the basekernels in 3.15 are positive definite during implementation. This assumption imposes all $a_m \geq 0$. Without loss of generality If all a_m values are also supposed to be ordered in a descending manner ($a_1 \geq a_2 \geq .... \geq a_M$) by using the Lagrangian multipliers for the constraints in 3.12 we arrive (Xu et al., 2013b) at the following solution to the problem in 3.15:

$$\mu_m = \begin{cases} \theta & m \leq \omega \\ \frac{(1-\omega\theta)\sqrt{a_m}}{\sum_{p=\omega+1}^{M}\sqrt{a_p}} & m \geq \omega \end{cases} \tag{3.16}$$

Where ω is the number of kernels that their coefficient equals to the upper bound θ and is calculated according to the following criteria:

$$\min\{p \in \{0,1,...,M-1\} | \frac{\sqrt{a_{p+1}(1-p\theta)}}{\sum_{m=p+1}^{M}\sqrt{a_m}} \leq \theta\} \tag{3.17}$$

It is also proven that (Xu et al., 2013b) for the optimal values of μ^*: $\forall\, a_p > a_q$, if $\mu_q^* = \theta \Rightarrow \mu_p^* = \theta$. Algorithm 1 demonstrates the underlying procedure. The final step is to use Dusk's as input representation of data to the SM1MKL algorithm which is explained in *chapter* 4.

CHAPTER 4

EXPERIMENT EVALUATION

The application of DuSK kernels for solving multi view problem is the baseline of our experiments. Two learning methods that use DuSK kernels are examined to tackle the classification problem of multi view brain network images. As explained in appendix **??** the used dataset consists of two sets of fMRI and DTI brain images for 52 subjects affected by bipolar disease along with brain images of 47 healthy controls. The problem is to train a classifier that can efficiently discriminate between these two classes. In addition, data are collected and preprocessed to highlight the brain network connectivity information in fMRI and DTI images corresponding to each of 97 subjects. All experiments are implemented in Matlab. CP factorization is obtained using Tensorlab toolbox for Matlab (Sorber et al., 2014). The factor matrices are set to be initialized randomly and the alternating least square method is chosen as optimization approach for CP decomposition. However, for higher order tensors ($order \geq 3$) there is only bounds for the true rank of a tensor; therefore, a set of rank values $R \in \{1,2,...,8\}$ are all examined during each experiment to find the true rank resulting in better accuracies. These are common ranks that result in better accuracy values in brain images according to(He et al., 2014).

4.1 Experimnets

4.1.1 Dusk in the framework of SM1MKL

One of the promising methods to deal with multi-view data is multiple kernel learning method that can represent each view according to a kernel matrix and then simultaneously search for kernel

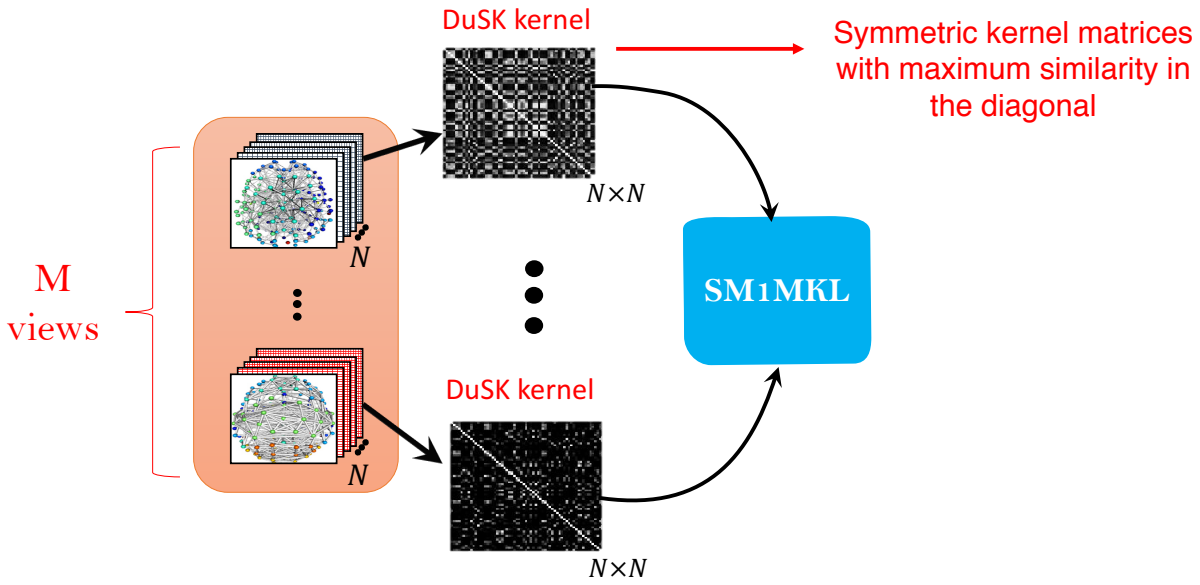


Figure 2: An overview of the proposed algorithm

combination coefficients and α parameters of SVM classifier. The specific variant of MKL that we use for multi-view structured data is the soft margin multiple kernel learning method that penalizes very low or very high weights for predefined kernels by adjusting the trade-off parameter θ for kernel slack variables. Therefore; it can be considered as a desired approach for our multi-view problem where fMRI and DTI both contain useful information. The diagram illustrated in *figure* 3 is an overview of the steps for the proposed algorithm. The inputs of the SM1MKL algorithm are changed in each experiment to evaluate the best choice of DuSK basekernels.

DuSK$_{RBF}$ and DuSK$_{Poly}$ kernels are used as basekernels of this method. The range of Bandwidth parameter for DuSK$_{RBF}$, degree parameter for DuSK$_{Poly}$ and trade off parameter of SVM are chosen

from $\sigma \in \{-5,...,5\}$, $d \in \{1,2,3\}$ and $C \in \{-5,...,5\}$ respectively. There is no trivial way to find the optimal combination of these parameters and a grid search is conducted that performs the learning process for all possible combinations of these parameters. The implemented grid search is illustrated in *Algorithm* 2. The parameter optimization part for a given decomposition is done a total of 10 times by putting 20% of data aside to avoid overfitting of parameters to the data. Then optimized parameters are used as fixed parameters of SM1MKL and the total reported accuracy on whole data is obtained by averaging the accuracy of 5-fold cross validation over 10 repeats. The whole procedure is repeated for the all ranks to achieve better accuracy results corresponding to a specific CP decomposition rank. *Figure* 2 illustrates the procedure for parameter optimization and the training and testing process for all ranks.

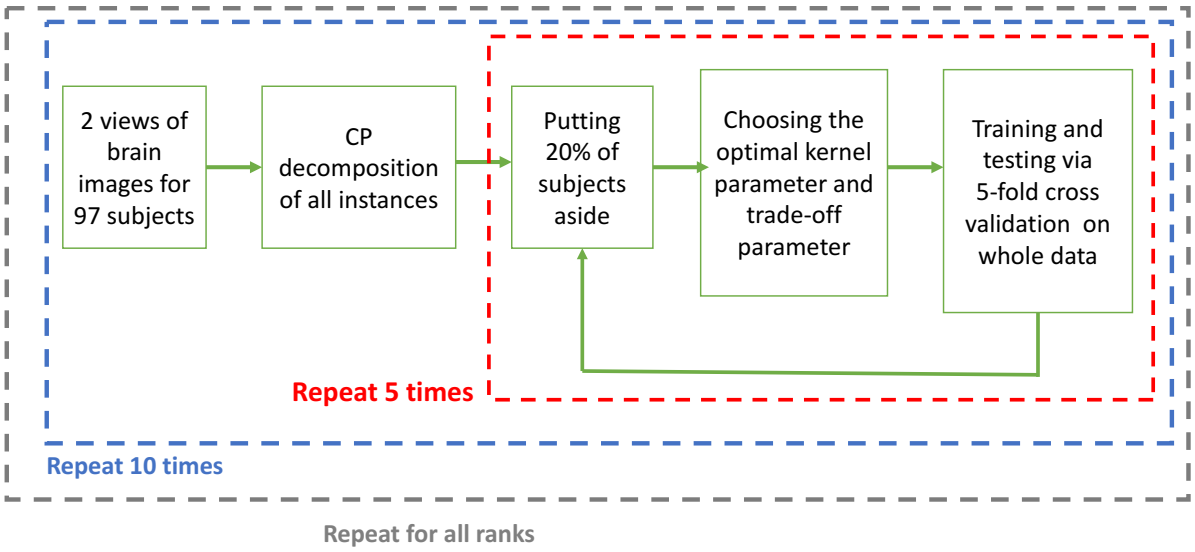


Figure 3: An overview of the proposed method

The first experiment is inspired by the success of DuSK$_{RBF}$ kernels on 3D brain images (He et al., 2014). Two DuSK$_{RBF}$ kernels are used to represent both set of brain images where the bandwidth parameters are σ_1 and σ_2 for each of the kernels respectively. The second experiment is the first application of DuSK$_{Poly}$ on brain network images where the degree parameters are d_1 and d_2 for each of the kernels. Eventually, the third experiment uses a combination of DuSK$_{RBF}$ and DuSK$_{Poly}$ in an attempt to evaluate whether different basekernels could capture the sufficient measure of similarity for our purpose. The bandwidth parameter σ and degree parameter d are the kernel parameters in this experiment. Training and testing through soft margin MKL, which iteratively optimizez the SVM parameters α and kernel weights μ, is implemented using *LIBMKL* package introduced in (Xu et al., 2013b) for SM1MKL algorithm. The algorithm 2 shows the procedure for the third experiment.

As it is illustrated in emphTable I the soft margin multiple kernel learning method with DuSK$_{R}BF$ and DuSK$_{P}oly$ kernels outperforms other two settings where the same type of kernel is used for both of the views.

Note that there is no trivial way to determine which kernel is a better choice for each of views and the reported accuracy is the choice with higher accuracy. In other words, the kernel that performs best on one view is not necessarily a good choice for the other view. When we introduce an algorithm that can learn an optimized combination, the accuracy increases. The parameter sensitivity for this method is demonstrated in *Figure* 5. This plot illustrates how the obtained accuracy values are affected by the CP decomposition rank.

Algorithm 2 SM1MKL algorithm with DuSK-RBF and DuSK-Poly basekernels

for $d = d1 : d2$ **do**
  for $g = g1 : g2$ **do**
    for $c = c1 : c2$ **do**
      calculate Dusk-RBF and Dusk-Poly kernels for test and train
      Train and test via SM1MKL and store accuracy values
    end for
  end for
end for
return the optimal values of c, d, g corresponding to highest accuracy value

	DuSK_{RBF}	DuSK_{Poly}	DuSK_{RBF} & DuSK_{Poly}
DuSK-SM1MKL on Multi-view	(R=3) 0.635 ± 0.034	(R=7) 0.584 ± 0.031	(R=2) 0.640 ± 0.015
DuSK-SVM on concatenated fMRI & DTI	(R=3) 0.608 ± 0.018	(R=5) 0.604 ± 0.029	N/A

TABLE I: Accuracy comparison of proposed DuSK-SM1MKL on multi-view data and the baseline DuSK-SVM on concatenated data

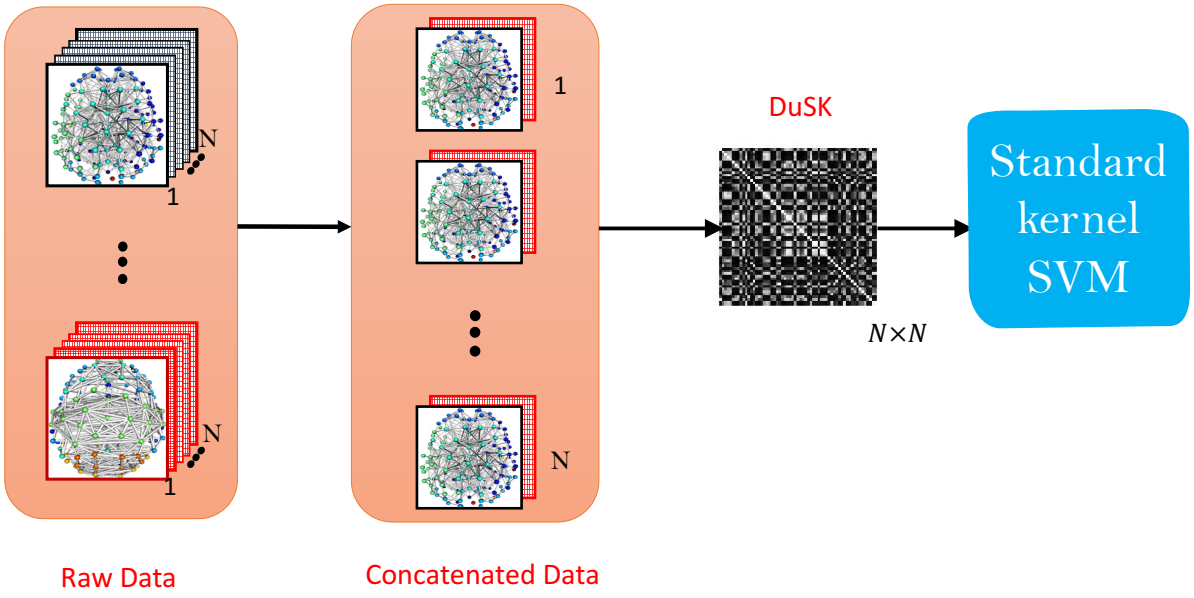


Figure 4: Summary of the general procedure repeated in all of the experiment

4.1.2 Dusk on the concatenated data

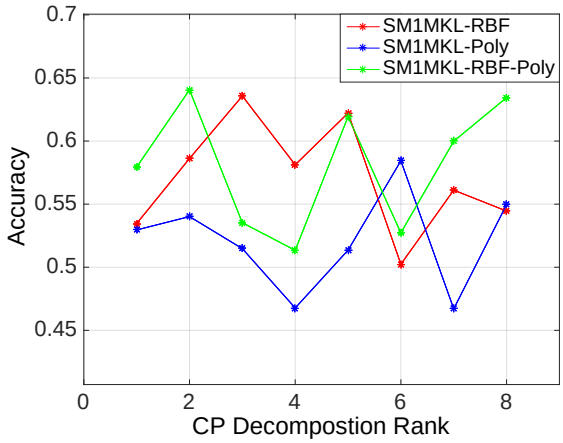
A conventional method for learning on multi-view homogeneous data is to concatenate multiple views to obtain a single representation. This is considered to be a naive way of early combination for multi-view data since it is known to cause overfitting. This method is used as a baseline for our proposed method where a single view is constructed from multiple views and then a single DuSK is calculated for concatenated data. The overview of this baseline approach is shown in *figure* 4.

Despite the second order CP factorization that was required for the proposed method, the concatenated single version is a third order tensor that requires third order CP factorization. Both of DuSK_{RBF}

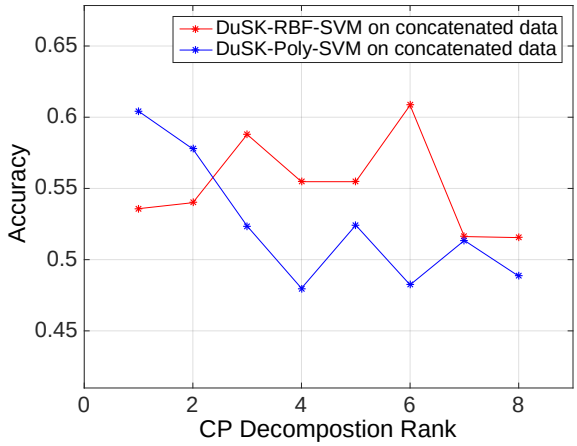
and DuSK_{Poly} are tested on the concatenated data to determine the best representing kernel that results in higher accuracy.

Figure 5: Sensitivity of accuracy to the CP rank for DuSK-SM1MKL and DuSK-SVM methods

(a) DuSK-SM1MKL



(b) DuSK-SVM



The learning method for this experiment is standard C-SVM method and is implemented by the well-known *libsvm* library for support vector machines (Chang and Lin, 2011). According to the *Table* I concatenation method cannot beat the proposed multiple kernel learning with two kernels of different types. As it can be seen in *Figure* 5, the accuracy results are less affected by decomposition rank of CP for concatenated data.

CHAPTER 5

SUMMARY AND CONCLUSION

To the best of our knowledge, this work is the first effort to apply tensor learning method to the brain network images. We used structure preserving kernels in the framework of multiple kernel learning algorithm to solve the problem of learning on the multi-view structured data. The results clearly show that MKL algorithm tunes the kernel weights to achieve better accuracy results. It is interesting to note that within the MKL method the approach that uses the same type of kernel for both views cannot beat the accuracy obtained by using different types of kernels. It can be inferred that each of the views are adapted more to one of the kernels than the other. The structure preserving MKL method performs better compared to a baseline approach that uses early concatenation of views to first obtain a single view. It is also concluded that the accuracy of proposed method is highly affected by the rank of CP decomposition.

The optimal kernel calculated in the proposed algorithm is calculated as a linear combination of basekernels; however, other types of combinations such as product kernels can be examined. In the other hand, the used dataset in this study consisted of two views with the same order and same size. An interesting extension to this work can be using views with different sizes and even different dimensions in the framework of multiple kernel learning.

APPENDIX

DATA COLLECTION AND PREPROCCESSING

This dataset consists of the fMRI and DTI image data of 52 bipolar I subjects who are in euthymia and 45 healthy controls with matched age and gender. The resting-state fMRI scan was acquired on a 3T Siemens Trio scanner using a T2*-weighted echo planar imaging (EPI) gradient-echo pulse sequence with integrated parallel acquisition technique (IPAT), set with TR = 2 sec, TE = 25 msec, flip angle = 78, matrix = 64x64, FOV = 192 mm, in-plane voxel size = 3x3 mm, slice thickness = 3 mm, 0.75 mm gap, and 30 total interleaved slices. Two TRs at the beginning of the scan were discarded to allow for scanner equilibration. There are 208 volumes acquired for the total sequence time of 7 min and 2 sec. Diffusion weighted MRI data were acquired on a Siemens 3T Trio scanner. Sixty contiguous axial brain slices were collected using the following parameters: 64 diffusion-weighted (b = 1000s/mm2) and 1 non-diffusion weighted scan; field of view (FOV) 190x190 mm; voxel size 2x2x2 mm; TR = 8400 ms; TE = 93 ms. In addition, high-resolution structural images were acquired using T1-weighted magnetization-prepared rapid-acquisition gradient echo (MPRAGE; FOV 250x250 mm; voxel size: 1x1x1 mm; TR = 1900 ms, TE = 2.26 ms, flip angle = 9).

For the Bipolar dataset, the brain networks were constructed using the CONN3 toolbox (Whitfield-Gabrieli and Nieto-Castanon, 2012). The raw EPI images were first realigned and co-registered, after which we perform the normalization and smoothing. Then the confound effects from motion artifact, white matter, and CSF were regressed out of the signal. Finally, the brain networks were derived using

APPENDIX (Continued)

the pairwise signal correlations based on the 82 labeled Freesurfer-generated cortical/subcortical gray matter regions.

CITED LITERATURE

[Akaho, 2006] Akaho, S.: A kernel method for canonical correlation analysis. arXiv preprint cs/0609071, 2006.

[Bach et al. , 2004] Bach, F. R., Lanckriet, G. R., and Jordan, M. I.: Multiple kernel learning, conic duality, and the smo algorithm. In Proceedings of the twenty-first international conference on Machine learning, page 6. ACM, 2004.

[Blum and Mitchell, 1998] Blum, A. and Mitchell, T.: Combining labeled and unlabeled data with co-training. In Proceedings of the eleventh annual conference on Computational learning theory, pages 92–100. ACM, 1998.

[Cao et al. , 2015] Cao, B., Kong, X., Zhang, J., Philip, S. Y., and Ragin, A. B.: Identifying hiv-induced subgraph patterns in brain networks with side information. Brain informatics, 2(4):211–223, 2015.

[Carroll and Chang, 1970] Carroll, J. D. and Chang, J.-J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of ?eckart-young? decomposition. Psychometrika, 35(3):283–319, 1970.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J.: Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3):27, 2011.

[Chen et al. , 2005] Chen, P.-H., Lin, C.-J., and Schölkopf, B.: A tutorial on ν-support vector machines. Applied Stochastic Models in Business and Industry, 21(2):111–136, 2005.

[Cortes et al. , 2009] Cortes, C., Mohri, M., and Rostamizadeh, A.: L 2 regularization for learning kernels. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pages 109–116. AUAI Press, 2009.

[Dasgupta et al. , 2002] Dasgupta, S., Littman, M. L., and McAllester, D. A.: Pac generalization bounds for co-training. In Advances in neural information processing systems, pages 375–382, 2002.

[Harshman, 1970] Harshman, R. A.: Foundations of the parafac procedure: models and conditions for an" explanatory" multimodal factor analysis. 1970.

[He et al. , 2014] He, L., Kong, X., Yu, P. S., Yang, X., Ragin, A. B., and Hao, Z.: Dusk: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages. In Proceedings of the 2014 SIAM International Conference on Data Mining, pages 127–135. SIAM, 2014.

[Ho, 1998] Ho, T. K.: The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8):832–844, 1998.

[Hotelling, 1936] Hotelling, H.: Relations between two sets of variates. Biometrika, 28(3/4):321–377, 1936.

[Kloft et al. , 2011] Kloft, M., Brefeld, U., Sonnenburg, S., and Zien, A.: Lp-norm multiple kernel learning. Journal of Machine Learning Research, 12(Mar):953–997, 2011.

[Kolda and Bader, 2009] Kolda, T. G. and Bader, B. W.: Tensor decompositions and applications. SIAM review, 51(3):455–500, 2009.

[Lanckriet et al. , 2004] Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I.: Learning the kernel matrix with semidefinite programming. Journal of Machine learning research, 5(Jan):27–72, 2004.

[Murphy, 2012] Murphy, K. P.: Machine learning: a probabilistic perspective. MIT press, 2012.

[Muslea et al. , 2002] Muslea, I., Minton, S., and Knoblock, C. A.: Active+ semi-supervised learning= robust multi-view learning. In ICML, volume 2, pages 435–442, 2002.

[Nickel and Tresp, 2013] Nickel, M. and Tresp, V.: An analysis of tensor models for learning on structured data. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 272–287. Springer, 2013.

[Nigam and Ghani, 2000] Nigam, K. and Ghani, R.: Analyzing the effectiveness and applicability of co-training. In Proceedings of the ninth international conference on Information and knowledge management, pages 86–93. ACM, 2000.

[Ragin et al. , 2012] Ragin, A. B., Du, H., Ochs, R., Wu, Y., Sammet, C. L., Shoukry, A., and Epstein, L. G.: Structural brain alterations can be detected early in hiv infection. Neurology, 79(24):2328–2334, 2012.

[Rakotomamonjy et al. , 2008] Rakotomamonjy, A., Bach, F. R., Canu, S., and Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research, 9(Nov):2491–2521, 2008.

[Richiardi et al. , 2011] Richiardi, J., Eryilmaz, H., Schwartz, S., Vuilleumier, P., and Van De Ville, D.: Decoding brain states from fmri connectivity graphs. Neuroimage, 56(2):616–626, 2011.

[Shon et al. , 2006] Shon, A., Grochow, K., Hertzmann, A., and Rao, R. P.: Learning shared latent structure for image synthesis and robotic imitation. In Advances in neural information processing systems, pages 1233–1240, 2006.

[Sigal et al. , 2009] Sigal, L., Memisevic, R., and Fleet, D. J.: Shared kernel information embedding for discriminative inference. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 2852–2859. IEEE, 2009.

[Signoretto, 2011] Signoretto, M.: Kernels and tensors for structured data modelling. Doctoral dissertation, Ph. D. thesis, 2011.

[Signoretto et al. , 2011] Signoretto, M., De Lathauwer, L., and Suykens, J. A.: A kernel-based framework to tensorial data analysis. Neural networks, 24(8):861–874, 2011.

[Signoretto et al. , 2012] Signoretto, M., Olivetti, E., De Lathauwer, L., and Suykens, J. A.: Classification of multichannel signals with cumulant-based kernels. IEEE Transactions on Signal Processing, 60(5):2304–2314, 2012.

[Sorber et al. , 2014] Sorber, L., Van Barel, M., and De Lathauwer, L.: Tensorlab v2. 0. Available online, January, 2014.

[Srebro, 2004] Srebro, N.: Learning with matrix factorizations. 2004.

[Tucker, 1963] Tucker, L. R.: Implications of factor analysis of three-way matrices for measurement of change. Problems in measuring change, 122137, 1963.

[Tzourio-Mazoyer et al. , 2002] Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., and Joliot, M.: Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. Neuroimage, 15(1):273–289, 2002.

[Varma and Babu, 2009] Varma, M. and Babu, B. R.: More generality in efficient multiple kernel learning. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 1065–1072. ACM, 2009.

[Wang and Zhou, 2007] Wang, W. and Zhou, Z.-H.: Analyzing co-training style algorithms. In European Conference on Machine Learning, pages 454–465. Springer, 2007.

[Whitfield-Gabrieli and Nieto-Castanon, 2012] Whitfield-Gabrieli, S. and Nieto-Castanon, A.: Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks. Brain connectivity, 2(3):125–141, 2012.

[Xu et al. , 2013a] Xu, C., Tao, D., and Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634, 2013.

[Xu et al. , 2013b] Xu, X., Tsang, I. W., and Xu, D.: Soft margin multiple kernel learning. IEEE transactions on neural networks and learning systems, 24(5):749–761, 2013.

[Zhao et al. , 2013a] Zhao, Q., Zhou, G., Adalı, T., Zhang, L., and Cichocki, A.: Kernel-based tensor partial least squares for reconstruction of limb movements. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 3577–3581. IEEE, 2013.

[Zhao et al. , 2013b] Zhao, Q., Zhou, G., Adali, T., Zhang, L., and Cichocki, A.: Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data. IEEE Signal Processing Magazine, 30(4):137–148, 2013.

VITA

Sepideh Esmaeilpour Charandabi

Education	M.S. Electrical Engineering University of Illinois at Chicago (GPA: 4/4)	2015 – 2017
	B.S. Electrical Engineering (Electronics) Islamic Azad University of Tabriz, Iran (GPA: 3.5/4)	2005 – 2009
Skills	Matlab and Simulink (Machine learning, Neural Network, Image analysis, Optimization, Estimation) Programming in C++ and Python Latex and Microsoft office Industrial automation (programming for PLC's and microcontrollers)	
Experience	Teaching assistant for ECE210 and ECE434 at UIC	2016
	Research on the round 8 of Yelp dataset challenge	
	Research on watermark detection and retrieval	