MULTI-KERNEL REGRESSION FOR GRAPH SIGNAL PROCESSING

Arun Venkitaraman, Saikat Chatterjee, Peter Händel

Department of Information Science and Engineering, School of Electrical Engineering
KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
arunv@kth.se, sach@kth.se, ph@kth.se

ABSTRACT

We develop a multi-kernel based regression method for graph signal processing where the target signal is assumed to be smooth over a graph. In multi-kernel regression, an effective kernel function is expressed as a linear combination of many basis kernel functions. We estimate the linear weights to learn the effective kernel function by appropriate regularization based on graph smoothness. We show that the resulting optimization problem is shown to be convex and propose an accelerated projected gradient descent based solution. Simulation results using real-world graph signals show efficiency of the multi-kernel based approach over a standard kernel based approach.

Index Terms— Graph signal processing, kernel regression, convex optimization.

1. INTRODUCTION

Kernel regression is a nonlinear learning approach used extensively in classification, regression, and clustering problems [1, 2, 3]. Typically, kernel regression needs training with a significant amount of reliable data to ensure a good prediction performance. In the context of limited and noisy data availability, we recently proposed kernel regression for target signals by incorporating an additional structure that the target signal is smooth over a graph [4]. There are prior works in the literature on using graph structures for kernel regression [5, 6, 7, 8]. These existing works do not use the direct approach of predicting the complete target signal by enforcing smoothness constraint over a graph. The prior works consider kernels defined among nodes of the same graph signal for signal completion or recovery.

Performance of kernel-based approaches varies significantly with the choice of the kernel functions and parameters. The optimal choice of the kernel parameter is a difficult problem to solve analytically or through cross-validation. A natural approach is to express a general kernel function that is a linear combination of many known 'basis' kernels and find their optimal combination adapted to the given data [9]. This is referred to as multi-kernel learning in the literature and has been shown to be useful in many applications [10, 11, 12, 13]. In this article, our contribution is to develop a multi-kernel learning approach for graph signal processing where we predict vector targets. We assume that the vector target signals that we predict are smooth graph signals over an underlying graph. The resulting multi-kernel learning approach is shown to be a convex optimization problem. We then develop a projected gradient descent algorithm to learn the multi-kernel parameters by using Nesterov's method [14]. There exist recent works on developing multi-kernel regression for graph signals in [15, 16], where the prediction is made on a subset of nodes of a graph from the signals at the remaining nodes through a kernel regression across nodes. Our proposed approach is different from these works by formulation. We explicitly consider that the target is a vector signal lying over a graph, and the input signal is not necessarily over a graph. Our goal is then to make prediction for the smooth graph signal target given a new input signal by learning the parameters of multi-kernel function.

The authors would like to acknowledge the support received from the Swedish Research Council.

1.1. Kernel regression for graph signals

We provide background in this section on recently proposed kernel regression for graph signal processing [16]. In graph signal processing, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ denotes a graph with M vertices indexed by set $\mathcal{V}$, edge set $\mathcal{E}$, and the adjacency matrix $\mathbf{A} = [a_{ij}]$, $a_{ij} > 0$. A graph signal over $\mathcal{G}$ is a vector in $\mathbb{R}^M$ whose components denote the values of the signal at the nodes indexed by $\mathcal{V}$ [17, 18]. We consider only undirected graphs in our analysis, which corrsponds to symmetric $\mathbf{A}$. The graph-Laplacian matrix $\mathbf{L}$ of $\mathcal{G}$ is then defined as [19] $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathrm{diag}(d_1, d_2 \cdots, d_M)$ denotes the diagonal degree matrix with $d_i = \sum_j a_{ij}$. By construction, $\mathbf{L}$ is symmetric and positive semi-definite. The smoothness of a graph signal $\mathbf{y} = [y(1), y(2), \cdots, y(M)]^\top$ is quantified by the following quadratic form:

$$l(\mathbf{y}) = \mathbf{y}^\top \mathbf{L} \mathbf{y} = \sum_{(ij)\in\mathcal{E}} a_{ij}(y(i) - y(j))^2.$$

A small $l(\mathbf{y})$ implies that $\mathbf{y}$ is smooth since its value varies smoothly over connected nodes.

We next briefy discuss a standard kernel regression. Consider a set of N training observations of input $\mathbf{x}_n \in \mathbb{R}^L$ and the corresponding target $\mathbf{t}_n \in \mathbb{R}^M$. Let $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^K$ denotes a vector function of the input $\mathbf{x}$. Then, kernel regression makes prediction of output $\mathbf{y}$ for a new input $\mathbf{x}$ of the form [2]

$$\mathbf{y} = \mathbf{W}^\top \boldsymbol{\phi}(\mathbf{x}),$$

where the regression coefficient matrix is $\mathbf{W}$, found by

$$\mathbf{W} = \arg\min_{\mathbf{W}} \sum_{n=1}^{N} \|\mathbf{t}_n - \mathbf{y}_n\|_2^2 + \alpha\, \mathrm{tr}(\mathbf{W}^\top \mathbf{W}),\ \alpha \geq 0. \quad (1)$$

The optimal $\mathbf{W}$ depends on inner products of form $\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ [2]. On generalizing the inner product to a kernel function $k(\mathbf{x}, \mathbf{x}')$, the predicted output takes the form:

$$\mathbf{y} = \boldsymbol{\Psi}^\top \mathbf{k}(\mathbf{x}),$$

where $\boldsymbol{\Psi} = (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{T}$, $\mathbf{K} \in \mathbb{R}^{N\times N}$ denotes the kernel matrix with (m, n)th entry is equal to $k(\mathbf{x}_m, \mathbf{x}_n)$, $\mathbf{T} = [\mathbf{t}_1\, \mathbf{t}_2 \cdots \mathbf{t}_N]^\top$ is the target matrix, and $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \cdots, k(\mathbf{x}_N, \mathbf{x})]^\top$.

We finally discuss kernel regression for graph signal processing, proposed in [4]. It is assumed that the entire target signal is smooth over the underlying graph $\mathcal{G}$. In kernel regression for graph signal processing, the following optimization problem is solved [4]:

$$\min_{\mathbf{W}} \sum_{n=1}^{N} \|\mathbf{t}_n - \mathbf{y}_n\|_2^2 + \alpha \,\mathrm{tr}(\mathbf{W}^\top \mathbf{W}) + \beta \sum_{n=1}^{N} l(\mathbf{y}_n),\ \alpha, \beta \geq 0, \quad (2)$$

where the predicted signal $\mathbf{y}_n = \mathbf{W}^\top \boldsymbol{\phi}(\mathbf{x}_n)$ is enforced to be smooth over $\mathcal{G}$ by using the regularization term $\sum_{n=1}^{N} l(\mathbf{y}_n)$. The optimization problem (2) is equivalent to the following optimization problem [4]

$$\begin{aligned} \min_{\boldsymbol{\Psi}} (\mathrm{tr}(\mathbf{T}^\top \mathbf{T}) - 2\mathrm{tr}\left(\mathbf{T}^\top \mathbf{K} \boldsymbol{\Psi}\right) + \mathrm{tr}(\boldsymbol{\Psi}^\top \mathbf{K} \mathbf{K} \boldsymbol{\Psi}) \\ + \alpha \,\mathrm{tr}(\boldsymbol{\Psi}^\top \mathbf{K} \boldsymbol{\Psi}) + \beta \,\mathrm{tr}\left(\boldsymbol{\Psi}^\top \mathbf{K} \mathbf{K} \boldsymbol{\Psi} \mathbf{L}\right)). \end{aligned} \quad (3)$$

The optimal prediction $\mathbf{y} = \boldsymbol{\Psi}^\top \mathbf{k}(\mathbf{x})$ is associated with the optimal $\boldsymbol{\Psi}$ matrix with the following form

$$\mathrm{vec}(\boldsymbol{\Psi}) = [(\mathbf{I}_M \otimes (\mathbf{K} + \alpha \mathbf{I}_N)) + (\beta \mathbf{L} \otimes \mathbf{K})]^{-1} \,\mathrm{vec}(\mathbf{T}), \quad (4)$$

where $\mathrm{vec}(\cdot)$ and $\otimes$ denote the vectorization and Kronecker product operations, respectively. We note that, if there is no graph smoothness penalty ($\beta = 0$) then (2) reduces to standard kernel regression.

2. MULTI-KERNEL REGRESSION FOR GRAPH SIGNALS

We now develop multi-kernel regression for graph signal processing. In multi-kernel regression we use a general kernel that is a linear combination of multiple pre-specified kernels, as follows

$$k(\mathbf{x}, \mathbf{x}') = \sum_{s=1}^{S} \rho_s k_s(\mathbf{x}, \mathbf{x}'), \quad (5)$$

where $k_s(\mathbf{x}, \mathbf{x}')$ denotes the s'th pre-specified kernel function. The weight vector $\boldsymbol{\rho} = [\rho_1, \rho_2, \cdots, \rho_S]$ is unknown. We assume that $\boldsymbol{\rho}$ has a bounded q-norm $\|\boldsymbol{\rho}\|_q$, where q may be 1 or 2. Let us use $\mathbf{K}_s$ to denote the kernel matrix associated with the s'th kernel function $k_s(\mathbf{x}_m, \mathbf{x}_n)$. We then have the general kernel matrix as $\mathbf{K} = \sum_{i=s}^{S} \rho_s \mathbf{K}_s$. Now, generalizing the kernel regression for graph signal processing optimization problem (3), we learn the optimal weight vector and regression coefficients by the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \min_{\boldsymbol{\Psi}} -2\mathrm{tr}\left(\mathbf{T}^\top \mathbf{K} \boldsymbol{\Psi}\right) + \mathrm{tr}\left(\boldsymbol{\Psi}^\top \mathbf{K} \mathbf{K} \boldsymbol{\Psi}\right) \\ + \alpha \,\mathrm{tr}(\boldsymbol{\Psi}^\top \mathbf{K} \boldsymbol{\Psi}) + \beta \,\mathrm{tr}\left(\boldsymbol{\Psi}^\top \mathbf{K} \mathbf{K} \boldsymbol{\Psi} \mathbf{L}\right),\ \alpha, \beta \geq \mathbf{0} \\ \text{subject to } \boldsymbol{\rho} \succeq \mathbf{0}, \|\boldsymbol{\rho}\|_q \leq R,\ \mathbf{K} = \textstyle\sum_{i=s}^{S} \rho_s \mathbf{K}_s \end{aligned} \quad (6)$$

where $\|\boldsymbol{\rho}\|_q \leq R$ is a regularization constraint. We note that for a fixed $\boldsymbol{\rho}$, the general kernel function is known. Therefore, for a given $\boldsymbol{\rho}$, the optimization problem becomes same as the problem in (4). In order to solve (6), we thus substitute $\boldsymbol{\Psi}$ for a given $\boldsymbol{\rho}$ and simplify the problem in the following equivalent form

$$\min_{\boldsymbol{\rho}} \mathrm{vec}(\mathbf{T})^\top \mathbf{B}(\boldsymbol{\rho}) \mathrm{vec}(\mathbf{T}) \text{ subject to } \boldsymbol{\rho} \succeq \mathbf{0}, \|\boldsymbol{\rho}\|_q \leq R, \quad (7)$$

where

$$\mathbf{B}(\boldsymbol{\rho}) = -(\mathbf{I}_M \otimes \mathbf{K}) \left[\mathbf{I}_M \otimes (\mathbf{K} + \alpha \mathbf{I}_N) + \beta \mathbf{L} \otimes \mathbf{K}\right]^{-1}. \quad (8)$$

The steps in the simplification of (6) to (7) is shown in the appendix.

Algorithm 1 Multi-kernel regression for graph signal processing

Initialize $\mu(0)$, $\boldsymbol{\rho}(0) = 0$, $\lambda(0) = 1$
i_{max} is the maximum number of iterations
while $i < i_{max}$ OR $\|\boldsymbol{\rho}(i) - \boldsymbol{\rho}(i-1)\|_2^2 > \epsilon$ **do**
  $\mu(i) = \mu(0)/i$
  $\mathbf{s}(i) = \boldsymbol{\rho}(i-1) - \mu(i) \nabla \gamma(\boldsymbol{\rho}(i-1))$
  $\mathbf{z}(i) = \arg\min_z \|\mathbf{s}(i) - \mathbf{z}\|_2^2$,
  subject to $\|\mathbf{z}\|_1 \leq R$, $\mathbf{z} \succeq 0$
  Set $\nu(i) = \frac{\lambda(i-1)-1}{\lambda(i+1)}$, $\lambda(i) = \frac{1+\sqrt{1+4\lambda(i-1)^2}}{2}$ (Nesterov)
  $\boldsymbol{\rho}(i) = (1 - \nu(i))\mathbf{z}(i) + \nu(i)\boldsymbol{\rho}(i-1)$ (Nesterov)
  $i \rightarrow i + 1$

2.1. Theoretical analysis

Proposition 1. *The optimization problem* (7) *is convex in* $\boldsymbol{\rho}$.

Proposition 2. *Let us define* $\gamma(\boldsymbol{\rho}) \triangleq vec(\mathbf{T})^\top \mathbf{B}(\boldsymbol{\rho}) vec(\mathbf{T})$. *We state that* $\gamma(\boldsymbol{\rho})$ *is a decreasing function of* $\boldsymbol{\rho}$ *and takes the maximum value of* 0 *at* $\boldsymbol{\rho} = \mathbf{0}$.

Proof: Proofs of propositions 1 and 2 are provided in the appendix.

Proposition 3. *The solution of* (7) *satisfies the boundary condition* $\|\boldsymbol{\rho}\|_q = R$.

Proof. This follows from Proposition 2 that $\gamma(\boldsymbol{\rho})$ is a decreasing function for $\boldsymbol{\rho} \succeq \mathbf{0}$ and takes the maximum value at $\mathbf{0}$. □

Proposition 4. *The gradient* $\nabla\gamma(\boldsymbol{\rho})$ *of* $\gamma(\boldsymbol{\rho})$ *is given by* $\nabla\gamma(\boldsymbol{\rho}) = [\nabla\gamma_1, \nabla\gamma_2, \cdots, \nabla\gamma_S]$ *where*

$$\begin{aligned} \nabla\gamma_s &= -\mathrm{vec}(\boldsymbol{\Psi})^\top (\mathbf{I}_M \otimes \mathbf{K}_s) \mathrm{vec}(\boldsymbol{\Psi}), \\ \mathrm{vec}(\boldsymbol{\Psi}) &= [(\mathbf{I}_M \otimes (\mathbf{K} + \alpha \mathbf{I}_N)) + (\beta \mathbf{L} \otimes \mathbf{K})]^{-1} \mathrm{vec}(\mathbf{T}) \end{aligned}$$

Also, $\nabla\gamma(\boldsymbol{\rho}) \preceq \mathbf{0}$.

Proof. The expression for the derivative is obtained using standard matrix derivative relations and the proof is omitted here for brevity. Each kernel matrix is positive semidefinite by definition, and hence $\mathbf{I}_M \otimes \mathbf{K}_s \succeq 0$, $\forall s$. Since each component of the gradient is a negative quadratic form with a positive semidefinite matrix, $\nabla\gamma_s \leq 0$ $\forall s$. Hence, $\nabla\gamma(\boldsymbol{\rho}) \preceq \mathbf{0}$. □

2.2. Estimating kernel parameters

We have shown that the optimization problem in (7) is convex. However, we find that it is non-trivial to express the optimization problem into a standard optimization form such that a convex optimization toolbox, like CVX [20], can be used. Hence, we use a projected gradient descent approach to solve the optimization problem. This is possible due to the property that $\gamma(\boldsymbol{\rho})$ has a gradient in closed-form (see Proposition 4). In order to improve the rate of convergence over that of the projected gradient descent, we use Nesterov's accelerated gradient approach [14]. For successive iterations of the gradient descent, we use a step-size inversely proportional to the number of iterations. The resulting steps of the gradient decent search are described in Algorithm 2.2.

Table 1. List of regularization parameter values

N	Linear regression (α)	Kernel regression on graph ($\sigma = 1$) (α, β)	R	$\mu(0)$
4	4.3	(0.02,5.5)	5	0.01
8	4.3	(0.03,5.5)	5	0.01
16	4.3	(0.06,5.5)	5	0.01
30	4.3	(0.1,5.5)	5	0.01

3. EXPERIMENTS

We apply our approach for target prediction on daily temperature measurements over 45 largest cities in Sweden [1]. At a particular time instant, temperature data over cities of a country is a graph signal. We consider 60 temperature measurements during the time period of October to December 2015 [21]. Let $\mathbf{t}_{o,n} \in \mathbb{R}^{45}$ is the vector that contains temperatures of 45 cities at a particular day. The corresponding input vector $\mathbf{x}_n \in \mathbb{R}^{45}$ is true temperature values from the previous day. Our interest is to predict the temperature of cities for the next day from the readings of current day. For the sake of experiments, we assume that in many applications the true target values are unknown or can not be observed. We only have access to noisy targets. To emulate such kind of situations, we deliberately corrupt true target temperature signal $\mathbf{t}_{o,n}$ and use the corrupted (noisy) targets for training the learning algorithms. Then, at the time of testing, we try to predict the true target and check the robustness of learning algorithms. In this experiment, we generated noisy targets $\mathbf{t}_n$ as follows:

$$\mathbf{t}_n = \mathbf{t}_{o,n} + \mathbf{e}_n,$$

where $\mathbf{e}_n$ denotes the white Gaussian noise at a signal-to-noise ratio (SNR) of 0 dB. Let d_{ij} denote the geodesic distance between i'th and j'th cities. We construct the adjacency matrix of the graph by setting

$$a_{ij} = \exp\left(-\frac{d_{ij}^2}{\sum_{i,j} d_{ij}^2}\right).$$

We randomly partition the total dataset into two subsets, each subset contains 30 samples. Then we use one dataset for training and the other for testing. We consider $S = 100$ Gaussian kernels where the parameters (variances) are picked in uniform step size from the span $[0.01, 10]$. Then we predict $\mathbf{t}_{o,n}$ using the output $\mathbf{y}_n$ and evaluate normalized mean-squared error (NMSE) performance. The NMSE is obtained by averaging over 100 different noise realizations and partitions of the data. We find that Algorithm 1 typically converges after around 50 steps with $\epsilon = 10^{-4}$. We compare the NMSE of our approach with that of the conventional regression using linear kernel $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, and kernel regression over graphs [4] with Gaussian kernel with parameter equal to 1. For the conventional regression and kernel regression with single Gaussian kernel, α and β resulting in the smallest NMSE for training data are found by exhaustive grid search. For multi-kernel regression, we set α and β to be the same as those obtained for the single Gaussian kernel. We consider the case of $q = 1$ in our experiments in this paper. We also performed experiments with $q = 2$, though they are not reported here for brevity. $R = 5$ is experimentally found to be a good choice for all training sample sizes. The parameters used in the experiments are shown in Table 1. The NMSE for testing data for different approaches is shown in Figure 1. We observe that the multi-kernel approach clearly outperforms the other two approaches particularly at low training sample sizes. An instance of $\boldsymbol{\rho}$ learned for a random data partition is shown in Figure 2, which demonstrates how the kernels that best explain the data are selected by the algorithm, since we have used $q = 1$ which is known to promote sparsity.

[1]The codes used for experiments may be found at: https://www.kth.se/en/ees/omskolan/organisation/avdelningar/information-science-and-engineering/research/reproducibleresearch

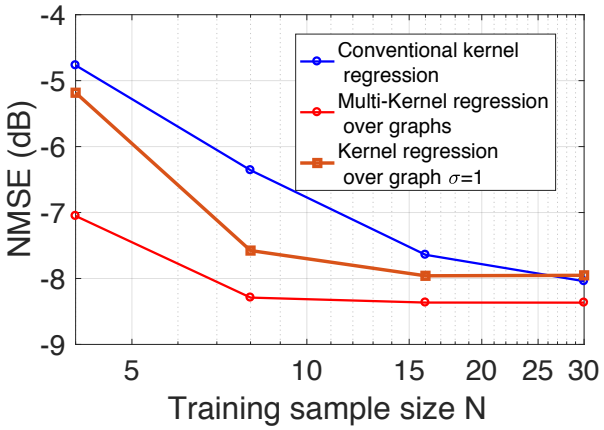

Fig. 1. NMSE as function of training sample size at SNR of 0dB.

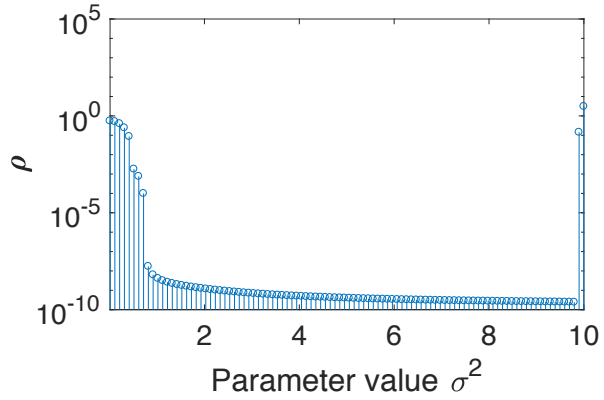

Fig. 2. An instance of $\boldsymbol{\rho}$ for the choice $q = 1$.

4. CONCUSIONS

We proposed a multi-kernel regression for targets that are smooth over a given graph. This was built on our earlier work of kernel regression for smooth signals over graphs which was shown to outperform conventional kernel regression for small training sample sizes and under noisy training. Multi-kernel regression was shown to be a convex optimization problem and some of its properties were studied. We then proposed an accelerated projected descent for evaluating the optimal kernel coefficients. Experiments on real-data demonstrated the potential of our approach for small training sizes and low SNR levels.

5. APPENDIX

We derive the steps for simplification of (6) to (7). We know that for fixed $\boldsymbol{\rho}$ the optimal $\boldsymbol{\Psi}$ is given by (4):

$$\text{vec}(\boldsymbol{\Psi}) = [(\mathbf{I}_M \otimes (\mathbf{K} + \alpha\mathbf{I}_N)) + (\beta\mathbf{L} \otimes \mathbf{K})]^{-1}\,\text{vec}(\mathbf{T}), \quad (9)$$

Let $\mathbf{B} = \mathbf{I}_M \otimes (\mathbf{K} + \alpha\mathbf{I}_N)$ and $\mathbf{C} = (\beta\mathbf{L} \otimes \mathbf{K})$. Then, $\boldsymbol{\Psi} = (\mathbf{B} + \mathbf{C})^{-1}\text{vec}(\mathbf{T})$. On using the property that $\text{tr}(\mathbf{a}_1^\top \mathbf{a}_2) = \text{vec}(\mathbf{a}_1)^\top \text{vec}(\mathbf{a}_2)$ and substituting optimal $\boldsymbol{\Psi}$ for a fixed $\boldsymbol{\rho}$, the objective function is given by

$$\begin{aligned}
C &= -2\text{tr}\left(\mathbf{T}^\top\mathbf{K}\boldsymbol{\Psi}\right) + \text{tr}\left(\boldsymbol{\Psi}^\top\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\right) \\
&\quad + \alpha\,\text{tr}(\boldsymbol{\Psi}^\top\mathbf{K}\boldsymbol{\Psi}) + \beta\,\text{tr}\left(\boldsymbol{\Psi}^\top\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\mathbf{L}\right) \\
&= -2\text{vec}(\mathbf{T})^\top\text{vec}(\mathbf{K}\boldsymbol{\Psi}) + \text{vec}\boldsymbol{\Psi})^\top\text{vec}(\mathbf{K}\mathbf{K}\boldsymbol{\Psi}) \\
&\quad + \alpha\,\text{vec}(\boldsymbol{\Psi})^\top\text{vec}(\mathbf{K}\boldsymbol{\Psi}) + \beta\,\text{vec}(\boldsymbol{\Psi})^\top\text{vec}(\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\mathbf{L}) \\
&= -2\text{vec}(\mathbf{T})^\top(\mathbf{I}\otimes\mathbf{K})\text{vec}(\boldsymbol{\Psi}) + \text{vec}\boldsymbol{\Psi})^\top(\mathbf{I}\otimes\mathbf{K}\mathbf{K})\text{vec}(\boldsymbol{\Psi}) \\
&\quad + \alpha\,\text{vec}(\boldsymbol{\Psi})^\top(\mathbf{I}\otimes\mathbf{K})\text{vec}(\boldsymbol{\Psi}) + \beta\,\text{vec}(\boldsymbol{\Psi})^\top(\mathbf{L}\otimes\mathbf{K}\mathbf{K})\text{vec}(\boldsymbol{\Psi}) \\
&= -2\text{vec}(\mathbf{T})^\top(\mathbf{I}\otimes\mathbf{K})(\mathbf{B}+\mathbf{C})^{-1}\text{vec}(\mathbf{T}) \\
&\quad + \text{vec}(\mathbf{T})^\top(\mathbf{B}+\mathbf{C})^{-1}(\mathbf{I}\otimes\mathbf{K}\mathbf{K})(\mathbf{B}+\mathbf{C})^{-\mathbf{1}}\text{vec}(\mathbf{T}) \\
&\quad + \alpha\,\text{vec}(\mathbf{T})^\top(\mathbf{B}+\mathbf{C})^{-1}(\mathbf{I}\otimes\mathbf{K})(\mathbf{B}+\mathbf{C})^{-\mathbf{1}}\text{vec}(\mathbf{T}) \\
&\quad + \beta\,\text{vec}(\mathbf{T})^\top(\mathbf{B}+\mathbf{C})^{-1}(\mathbf{L}\otimes\mathbf{K}\mathbf{K})(\mathbf{B}+\mathbf{C})^{-\mathbf{1}}\text{vec}(\mathbf{T}) \\
&= \text{vec}(\mathbf{T})^\top[-2(\mathbf{I}\otimes\mathbf{K}) + (\mathbf{B}+\mathbf{C})^{-1}\mathbf{H}](\mathbf{B}+\mathbf{C})^{-1}\text{vec}(\mathbf{T}).
\end{aligned}$$

where $\mathbf{H} = [(\mathbf{I}\otimes(\mathbf{I}+\alpha\mathbf{K})\mathbf{K}) + \beta\,(\mathbf{L}\otimes\mathbf{K}\mathbf{K})]$, and we have used properties of the $\text{tr}(\cdot)$ and $\text{vec}(\cdot)$ operators in the different steps. Using distributivity of Kronecker product, we have that

$$\mathbf{H} = [(\mathbf{I}\otimes(\mathbf{I}+\alpha\mathbf{K})\mathbf{K}) + \beta\,(\mathbf{L}\otimes\mathbf{K}\mathbf{K})] = (\mathbf{B}+\mathbf{C})(\mathbf{I}\otimes\mathbf{K}).$$

Substituting $\mathbf{H}$ in C, we get that

$$\begin{aligned}
C &= \text{vec}(\mathbf{T})^\top[-2(\mathbf{I}\otimes\mathbf{K}) + (\mathbf{I}\otimes\mathbf{K})](\mathbf{B}+\mathbf{C})^{-1}\text{vec}(\mathbf{T}). \\
&= -\text{vec}(\mathbf{T})^\top(\mathbf{I}\otimes\mathbf{K})(\mathbf{B}+\mathbf{C})^{-1}\text{vec}(\mathbf{T}).
\end{aligned}$$

Proposition 1. We prove this by showing that the equivalent problem in (6) is convex. We note that the objective function of (6) is the minimum of functions of the form

$$\begin{aligned}
\xi(\boldsymbol{\rho}) \quad &= -2\text{tr}\left(\mathbf{T}^\top\mathbf{K}\boldsymbol{\Psi}\right) + \text{tr}\left(\boldsymbol{\Psi}^\top\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\right) \\
&\quad +\alpha\,\text{tr}(\boldsymbol{\Psi}^\top\mathbf{K}\boldsymbol{\Psi}) + \beta\,\text{tr}\left(\boldsymbol{\Psi}^\top\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\mathbf{L}\right)
\end{aligned}$$

Since minimum of convex functions is also convex, it suffices to prove that $\xi(\boldsymbol{\rho})$ is a convex function of $\boldsymbol{\rho}$. We have that

$$\begin{aligned}
&\xi(\boldsymbol{\rho}) = -2\text{tr}\left(\mathbf{T}^\top\mathbf{K}\boldsymbol{\Psi}\right) + \text{tr}\left(\boldsymbol{\Psi}^\top\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\right) \\
&+ \alpha\,\text{tr}(\boldsymbol{\Psi}^\top\mathbf{K}\boldsymbol{\Psi}) + \beta\,\text{tr}\left(\boldsymbol{\Psi}^\top\mathbf{K}\mathbf{K}\boldsymbol{\Psi}\mathbf{L}\right) \\
&= -2\text{tr}\left(\mathbf{T}^\top\sum_{s=1}^{S}\rho_s\mathbf{K}_s\boldsymbol{\Psi}\right) \\
&+ \text{tr}\left(\boldsymbol{\Psi}^\top\sum_{r=1}^{S}\sum_{s=1}^{S}\rho_r\rho_s\mathbf{K}_r\mathbf{K}_s\boldsymbol{\Psi}\right) + \alpha\,\text{tr}\left(\boldsymbol{\Psi}^\top\sum_{s=1}^{S}\rho_s\mathbf{K}_s\boldsymbol{\Psi}\right) \\
&+ \beta\,\text{tr}\left(\boldsymbol{\Psi}^\top\sum_{r=1}^{S}\sum_{s=1}^{S}\rho_r\rho_s\mathbf{K}_r\mathbf{K}_s\boldsymbol{\Psi}\mathbf{L}\right) \quad (10)
\end{aligned}$$

Since summation and trace operations commute, we have

$$\begin{aligned}
\xi(\boldsymbol{\rho}) &= \sum_{s=1}^{S}\rho_s\text{tr}\left(-2\mathbf{K}_s\boldsymbol{\Psi}\mathbf{T}^\top\right) + \alpha\text{tr}\left(\mathbf{K}_s\boldsymbol{\Psi}\boldsymbol{\Psi}^\top\right) \\
&\quad + \sum_{r=1}^{S}\sum_{s=1}^{S}\rho_r\rho_s\text{tr}(\boldsymbol{\Psi}^\top\mathbf{K}_r\mathbf{K}_s\boldsymbol{\Psi}(\mathbf{I}_M + \beta\mathbf{L})) \\
&= \sum_{s=1}^{S}\rho_s\text{tr}\left(-2\mathbf{K}_s\boldsymbol{\Psi}\mathbf{T}^\top\right) + \alpha\text{tr}\left(\mathbf{K}_s\boldsymbol{\Psi}\boldsymbol{\Psi}^\top\right) \\
&\quad + \sum_{r=1}^{S}\sum_{s=1}^{S}\rho_r\rho_s\text{tr}\left((\mathbf{I}_M + \beta\mathbf{L})^{-\frac{1}{2}}\boldsymbol{\Psi}^\top\mathbf{K}_r\mathbf{K}_s\boldsymbol{\Psi}(\mathbf{I}_M + \beta\mathbf{L})^{-\frac{1}{2}}\right) \\
&= \mathbf{b}^\top\boldsymbol{\rho} + \boldsymbol{\rho}^\top\mathbf{C}\boldsymbol{\rho} \quad (11)
\end{aligned}$$

where $\mathbf{b} \in \mathbb{R}^S$ and $\mathbf{C} \in \mathbb{R}^{S\times S}$ such that the sth component of $\mathbf{b}$ is equal to $\mathbf{b}(s) = (-2\mathbf{K}_s\boldsymbol{\Psi}\mathbf{T}^\top + \alpha\text{tr}(\mathbf{K}_s\boldsymbol{\Psi}\boldsymbol{\Psi}^\top)$ and the (r,s)th element of matrix $\mathbf{C}$ is given by

$$\begin{aligned}
&\mathbf{C}(r,s) \quad (12) \\
&= \text{tr}((\mathbf{I}_M + \beta\mathbf{L})^{-\frac{1}{2}}\boldsymbol{\Psi}^\top\mathbf{K}_r\mathbf{K}_s\boldsymbol{\Psi}(\mathbf{I}_M + \beta\mathbf{L})^{-\frac{1}{2}}) \\
&= \text{vec}(\mathbf{K}_r\boldsymbol{\Psi}(\mathbf{I}_M + \beta\mathbf{L})^{-\frac{1}{2}})^\top\text{vec}(\mathbf{K}_s\boldsymbol{\Psi}(\mathbf{I}_M + \beta\mathbf{L})^{-\frac{1}{2}}),
\end{aligned}$$

using the property that $\text{tr}(\mathbf{a}_1^\top\mathbf{a}_2) = \text{vec}(\mathbf{a}_1)^\top\text{vec}(\mathbf{a}_2)$. This shows that $\mathbf{C}$ can be expressed as the Grammian matrix $\mathbf{C} = \mathbf{D}^\top\mathbf{D}$ where $\mathbf{D} \in \mathbb{R}^{mn\times S}$ is the matrix whose sth column is given by $\mathbf{d}_s = \text{vec}(\mathbf{K}_s\boldsymbol{\Psi}(\mathbf{I}_M + \beta\mathbf{L})^{-1/2})$. Since Grammian matrices are positive semidefinite [22], we have that $\mathbf{C}$ is symmetric and positive semidefinite, and hence $\xi(\boldsymbol{\rho})$ is convex. This concludes the proof. □

Proposition 2. Let $\boldsymbol{\rho}_2 \succeq \boldsymbol{\rho}_1 \succ \mathbf{0}$. For the case when $\boldsymbol{\rho} \neq \mathbf{0}$, let us assume $\mathbf{K} \succeq$ is invertible. This is not an unreasnable requirement as $\mathbf{K}$ is a nonnegative sum ($\boldsymbol{\rho} \succeq \mathbf{0}$) of kernel matrices: if atleast one of the kernel matrices is positive definite (which is usually the case for most kernel matrices), then $\mathbf{K}$ is positive definite and hence, invertible for $\boldsymbol{\rho} \neq \mathbf{0}$. Then $\mathbf{B}(\boldsymbol{\rho})$ is expressible as

$$\mathbf{B}(\boldsymbol{\rho}) = -\left[\mathbf{I}_M \otimes (\mathbf{I}_N + \alpha\mathbf{K}^{-1}) + \beta\mathbf{L}\otimes\mathbf{I}_N\right]^{-1}$$

Since $\boldsymbol{\rho}_2 \succeq \boldsymbol{\rho}_1$, we have that

$$\begin{aligned}
&\sum_{s=1}^{S}\rho_{1s}\mathbf{K}_s \preceq \sum_{s=1}^{S}\rho_{2s}\mathbf{K}_s, \text{ or} \\
&\left(\sum_{s=1}^{S}\rho_{1s}\mathbf{K}_s\right)^{-1} \succeq \left(\sum_{s=1}^{S}\rho_{2s}\mathbf{K}_s\right)^{-1}, \text{ or} \\
&\mathbf{I}_N + \alpha\left(\sum_{s=1}^{S}\rho_{1s}\mathbf{K}_s\right)^{-1} \succeq \mathbf{I}_N + \alpha\left(\sum_{s=1}^{S}\rho_{2s}\mathbf{K}_s\right)^{-1}, \text{ or} \\
&\left(\mathbf{I}_M \otimes (\mathbf{I}_N + \alpha\left(\sum_{s=1}^{S}\rho_{1s}\mathbf{K}_s\right)^{-1}\right) + \beta\mathbf{L}\otimes\mathbf{I}_N\right)^{-1} \\
&\preceq \left(\mathbf{I}_M \otimes (\mathbf{I}_N + \alpha\left(\sum_{s=1}^{S}\rho_{2s}\mathbf{K}_s\right)^{-1}\right) + \beta\mathbf{L}\otimes\mathbf{I}_N\right)^{-1}, \text{ or} \\
&\mathbf{B}(\boldsymbol{\rho}_1) \succeq \mathbf{B}(\boldsymbol{\rho}_2) \quad (13)
\end{aligned}$$

Since $\boldsymbol{\rho} = 0$ lies in the feasible region, we have that $\mathbf{B}(\boldsymbol{\rho})$ takes maximum value at $\boldsymbol{\rho} = \mathbf{0}$ and the correspondng value is given by setting $\boldsymbol{\rho} = \mathbf{0}$ in (8), we get that $\mathbf{B}(\mathbf{0}) = \mathbf{0}$ maximum value of $\text{vec}(\mathbf{T})^\top\mathbf{B}(\boldsymbol{\rho})\text{vec}(\mathbf{T}) = 0$. □

6. REFERENCES

[1] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. Int. Conf. Machine Learning*, 1998, pp. 515–521.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] Yo. Cho and L. K. Saul, "Kernel methods for deep learning," in *Adv. Neural Inform. Process. Syst.*, pp. 342–350. Curran Associates, Inc., 2009.

[4] A. Venkitaraman, S. Chatterjee, and P. Händel, "Kernel Regression for Signals over Graphs," *ArXiv e-prints*, June 2017.

[5] A. J. Smola and R. Kondor, *Kernels and Regularization on Graphs*, pp. 144–158, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[6] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," *Proc. ICML*, pp. 315–322, 2002.

[7] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, Feb 2017.

[8] V. N. Ioannidis, D. Romero, and G. B. Giannakis, "Kernel-based reconstruction of space-time functions via extended graphs," in *Asilomar Conf. Signals Syst. Comput.*, Nov 2016, pp. 1829–1833.

[9] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, July 2011.

[10] G. V. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March, pp. 6315–6319.

[11] G. Sonnenburg, S.and Rätsch and C. Schäfer, "A general and efficient multiple kernel learning algorithm," pp. 1273–1280, 2005.

[12] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *Proc. Conf. Uncertainty in Artificial Intelligence*, 2009, UAI '09, pp. 109–116.

[13] E. Castro, V. Gómez-Verdejo, M. Martínez-Ramón, K. A. Kiehl, and V. D. Calhoun, "A multiple kernel learning approach to perform classification of groups from complex-valued fmri data analysis: Application to schizophrenia," *NeuroImage*, vol. 87, no. Supplement C, pp. 1 – 17, 2014.

[14] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Inc., 1 edition, 2014.

[15] L. Zhang, D. Romero, and G. B. Giannakis, "Fast convergent algorithms for multi-kernel regression," in *IEEE Statistical Signal Processing Workshop (SSP)*, 2016, pp. 1–4.

[16] D. Romero, M. Ma, and G. B. Giannakis, "Estimating signals over graphs via multi-kernel learning," in *IEEE Statistical Signal Processing Workshop (SSP)*, June, pp. 1–5.

[17] D. I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[18] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[19] F. R. K. Chung, *Spectral Graph Theory*, AMS, 1996.

[20] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[21] Swedish Meteorological and Hydrological Institute (SMHI), "http://opendata-download-metobs.smhi.se/," .

[22] R. A. Horn and C. R. Johnson, Eds., *Matrix Analysis*, Cambridge University Press, New York, NY, USA, 1986.