Multitask Generalized Eigenvalue Program

Boyu Wang and **Joelle Pineau**
School of Computer Science
McGill University, Montreal, Canada
boyu.wang@mail.mcgill.ca, jpineau@cs.mcgill.ca

Borja Balle
Department of Mathematics and Statistics
Lancaster University, Lancaster, UK
b.deballepigem@lancaster.ac.uk

Abstract

We present a novel multitask learning framework called *multitask generalized eigenvalue program* (MTGEP), which jointly solves multiple related generalized eigenvalue problems (GEPs). This framework is quite general and can be applied to many eigenvalue problems in machine learning and pattern recognition, ranging from supervised learning to unsupervised learning, such as principal component analysis (PCA), Fisher discriminant analysis (FDA), common spatial pattern (CSP), and so on. The core assumption of our approach is that the leading eigenvectors of related GEPs lie in some subspace that can be approximated by a sparse linear combination of basis vectors. As a result, these GEPs can be jointly solved by a sparse coding approach. Empirical evaluation with both synthetic and benchmark real world datasets validates the efficacy and efficiency of the proposed techniques, especially for grouped multitask GEPs.

Introduction

The generalized eigenvalue problem (GEP) requires finding the solution of a system of equations:

$$Aw = \lambda Bw, \tag{1}$$

with respect to the pair (λ, w), where λ is the generalized eigenvalue, $w \in \mathbb{R}^d$, $w \neq 0$, is the corresponding generalized eigenvector, and $A, B \in \mathbb{R}^{d\times d}$. The GEP is useful as it provides an efficient approach to optimize the Rayleigh quotient

$$\max_{w\neq 0} \frac{w^\top Aw}{w^\top Bw}, \tag{2}$$

which arises in several pattern recognition and machine learning tasks. For example, both principal component analysis (PCA) (Jolliffe 2002) and Fisher discriminant analysis (FDA) (Bishop 2006), can be formulated as special cases of this problem. In most machine learning applications, A and B are estimated from data; in PCA, $B = I$, the identity matrix, and A is the covariance matrix estimated from data.

Although the GEP has been well studied over the years (Bie, Cristianini, and Rosipal 2005), to the best of our knowledge no one has tackled the problem of how to jointly solve *multiple* related GEPs, by sharing the common knowledge so that learning performance is better than independently solving each single GEP. This issue is especially important when the data for each GEP is insufficient, resulting in unreliable estimates of A and B, and therefore a poor estimate of the eigenvector w . Such a scenario may arise in many machine learning applications, including but not limited to:

- *Supervised learning*: perform multitask classification using FDA (Bishop 2006).
- *Unsupervised learning*: find principal components for multiple related datasets, yet each dataset consists of very few instances (Jolliffe 2002).
- *Spatial filter for signal processing*: design subject-specific spatial filters with a few electroencephalogram (EEG) data using common spatial pattern (CSP) algorithm (Ramoser, Müller-Gerking, and Pfurtscheller 2000), which is one of the most popular algorithms in brain-computer interface (BCI) research (Wolpaw et al. 2002). More recently, this technique has also been applied for discriminative feature construction (Karampatziakis and Mineiro 2014).

Some of these problems can be handled using existing multitask learning techniques (Caruana 1997). However, most previous work on multitask learning focuses only on supervised learning (Evgeniou, Micchelli, and Pontil 2005; Argyriou, Evgeniou, and Pontil 2008; Xue et al. 2007), has not been extended to the GEP setting.

On the practical side, our work is motivated by the need to improve EEG signal classification for Brain-Computer Interface (BCI) applications. Let $X \in \mathbb{R}^{d\times T}$ be a segment of multichannel EEG signals, where d is the number of channels and T is the number of sampled time points. The objective of a CSP algorithm is to design a series of spatial filters W by simultaneous diagonalization of two covariance matrices of classes of EEG patterns for each subject (task):

$$\begin{aligned} W^\top \Sigma^{(+)} W &= \Lambda^{(+)}, \\ W^\top \Sigma^{(-)} W &= \Lambda^{(-)}, \end{aligned} \tag{3}$$

such that $\Lambda^{(+)}$ and $\Lambda^{(-)}$ are diagonal matrices and $\Lambda^{(+)} + \Lambda^{(-)} = I$, where $\Sigma^{(c)} = \frac{1}{|\mathcal{I}_c|}\sum_{i\in\mathcal{I}_c} X_i X_i^\top$ and $\mathcal{I}_c$ ($c \in \{+,-\}$) is the set of indices of two classes of EEG patterns (e.g., left/right hand motor imagery). It can be shown that

Eq. 3 can be solved by finding a series of eigenvectors of Eq. 1, where $A = \Lambda^{(+)}$ and $B = \Lambda^{(-)}$ (Blankertz et al. 2008). After designing the spatial filters, the log-variances of the spatially filtered EEG signals are classified by FDA, an efficient and popular classifier for BCI (Lotte et al. 2007), which is also a GEP. In most BCI applications, however, the EEG signals of each subject are very limited, and therefore the learned spatial filter w can be unreliable. On the other hand, the EEG signals of different subjects may have some common information that can be shared. To address this issue, we propose the multitask generalized eigenvalue program (MTGEP) algorithm, which jointly solves K related GEPs. By leveraging knowledge of other GEPs, we expect the eigenvectors found by MTGEP is more reliable than the ones found by solving individual GEPs.

Method

We begin by formulating the multitask GEP (MTGEP), then present the algorithm for finding the leading eigenvector (MTGEP-L), followed by an extension that solves the entire spectrum of Eq. 1.

Problem Formulation

Let $\mathcal{S} = \{(A_1, B_1), \ldots, (A_K, B_K)\} \in \mathbb{R}^{d\times d}$ be the matrix pairs of K related GEPs. In our application, we assume that $\{A_k\} \in \mathbb{S}_+^d$ and $\{B_k\} \in \mathbb{S}_{++}^d, \forall k = \{1, \ldots, K\}$, where $\mathbb{S}_+^d$ ($\mathbb{S}_{++}^d$) denotes the set of symmetric positive semidefinite (definite) $d \times d$ matrices defined over $\mathbb{R}$. The objective is to maximize the summation of K Rayleigh quotients:

$$\max_{w_1,\ldots,w_K} \frac{1}{K}\sum_{k=1}^{K} \frac{w_k^\top A_k w_k}{w_k^\top B_k w_k}. \tag{4}$$

As Eq. 4 is decoupled with respect to w_k, it can be maximized by solving K GEPs individually. However if the data available for each task is small compared to its dimension, the estimates of A and B will be unreliable. In the PCA problem for example, where $B = I$ and A is the estimated covariance matrix, if the number of data points $N_k \ll d$ for each task, A_k cannot represent the covariance of each task properly, it is unlikely that the leading eigenvector solved by GEP will correctly maximize the variance of the data.

We tackle this problem by assuming that the K GEPs are related in a way such that their eigenvectors lie in some subspace that can be approximated by a sparse linear combination of a number of basis vectors. More formally, we assume that there is a dictionary $D \in \mathbb{R}^{d\times M}$ ($M < K$), and the leading eigenvector of each task can be represented by a subset of the basis vectors of D. In other words, let $\gamma_k \in \mathbb{R}^M$ be the sparse representation of the kth task with respect to D, then the objective function Eq. 4 can be formulated as:

$$\max_{D} \frac{1}{K}\sum_{k=1}^{K} \max_{\gamma_k} \frac{\gamma_k^\top D^\top A_k D\gamma_k}{\gamma_k^\top D^\top B_k D\gamma_k} - \rho||\gamma_k||_0, \tag{5}$$
$$\text{s.t.} \quad ||D||_F \le \mu,$$

where $||\gamma||_0$ is the ℓ_0-norm of γ, denoting the number of nonzero elements of γ, $||D||_F = (tr(DD^\top))^{1/2}$ is the Frobenius norm of matrix D. The ℓ_0 regularizer encourages γ to be sparse so that the knowledge embedded in D can be selectively shared. The norm constraint on D prevents the dictionary from being too large and overfitting the available data.

We see that in Eq. 5 that the K GEPs are coupled via the dictionary D that is shared across tasks, and therefore the K GEPs can be jointly learned in the context of multitask learning.

Algorithm 1 MTGEP for Leading Eigenvector (MTGEP-L)

Input: $\{(A_1, B_1), \ldots, (A_K, B_K)\}$, $maxIter$, # basis vectors M, regularization param. ρ

Solve each GEP to obtain $\{w^1, \ldots, w^K\}$
Initialize $t = 0$, $W^{(0)} = [w^1; \ldots; w^K]$
Initialize $D^{(0)}$ to the first M columns of U, where U is the obtained by singular value decomposition of $W^{(0)}$: $W^{(0)} = USV^\top$.
while $t < maxIter$ **do**
  for $k = 1, \ldots, K$ **do**
    Solve the kth SGEP (Eq. 6) to obtain $\gamma_k^{(t)}$.
  end for
  Update $D^{(t)}$ by solving Eq. 8.
  Normalize $D^{(t)}$ such that $||D^{(t)}||_F = M$.
  $t = t + 1$
  if converge **then**
    break
  end if
end while

Output: $D = D^{(t)}$, $\Gamma = [\gamma_1^{(t)}, \ldots, \gamma_K^{(t)}]$, $W = D\Gamma$

MTGEP for Leading Eigenvector

The objective function (Eq. 5) is not concave therefore we adopt the alternating optimization approach to obtain a local maximum (Bezdek and Hathaway 2003). We apply the following two optimization step alternately:

1. *Sparse coding*: given a fixed dictionary D, update sparse representation γ_k for each task.
2. *Dictionary update*: given fixed $\Gamma = [\gamma_1; \ldots; \gamma_K]$, update the dictionary D.

The proposed MTGEP-L for optimizing the leading eigenvector according to this approach is outlined in Algorithm 1, with details of each optimization step described next.

Sparse Coding Given a fixed dictionary D, Eq. 5 is decoupled and can be optimized by solving K individual GEPs:

$$\gamma_k = \arg\max_{\gamma} \frac{\gamma^\top P_k \gamma}{\gamma^\top Q_k \gamma} - \rho||\gamma||_0, \tag{6}$$

where $P_k = D^\top A_k D$ and $Q_k = D^\top B_k D$. Eq. 6 is called a sparse generalized eigenvalue problem (SGEP) and has been studied in (Moghaddam, Weiss, and Avidan 2006; Sriperumbudur, Torres, and Lanckriet 2007; Song, Babu, and Palomar 2014). In this work, we adopt the *bi-directional* search (Moghaddam, Weiss, and Avidan 2006) and *iteratively reweighed quadratic minorization* (IRQM) algorithm (Song, Babu, and Palomar 2014) to solve Eq. 6, and the

better empirical results between these two are reported in our experimental section.

Dictionary Update We initialize D using the approach proposed by (Kumar and Daumé III 2012). We first solve each GEP individually to obtain K leading eigenvectors $\{w^1, \ldots, w^K\}$, one for each task. Then the dictionary D is initialized as the first M left singular vectors of $W^{(0)} \in \mathbb{R}^{d \times K}$, where $W^{(0)}$ is constructed by $\{w^1, \ldots, w^K\}$, one for each column.

Given a fixed $\Gamma = [\gamma_1, \ldots, \gamma_K]$, the optimization problem (Eq. 5) becomes

$$D^{(t)} = \arg\max_D \sum_{k=1}^{K} \frac{\gamma_k^{(t)\top} D^\top A_k D \gamma_k^{(t)}}{\gamma_k^{(t)\top} D^\top B_k D \gamma_k^{(t)}}. \tag{7}$$

By applying the property of vectorization operator that $\gamma^\top D^\top \Sigma D \gamma = \text{vec}(D)^\top (\Sigma \otimes \gamma\gamma^\top)\text{vec}(D)$ to Eq. 7, we have the following equivalent objective function:

$$D^{(t)} = \arg\max_D \sum_{k=1}^{K} \frac{\text{vec}(D)^\top \left(A_k \otimes \gamma_k^{(t)} \gamma_k^{(t)\top}\right) \text{vec}(D)}{\text{vec}(D)^\top \left(B_k \otimes \gamma_k^{(t)} \gamma_k^{(t)\top}\right) \text{vec}(D)}, \tag{8}$$

where $\text{vec}(\cdot)$ is the vectorization operator and $\otimes$ is Kronecker product. Eq. 8 is a nonconave unconstrained optimization problem, but a local maximum can be found by standard gradient based algorithm, using $D^{(t-1)}$ as a warm start for computing $D^{(t)}$. As the Rayleigh quotient is invariant with respect to its argument scaling, we simply normalize D after each update step such that $||D||_F = \mu$ with $\mu = M$: $\text{vec}(D) = \frac{M \text{vec}(D)}{||D||_F}$.

Convergence Analysis

Let $\mathcal{L}(D, \Gamma) = \frac{1}{K} \sum_{k=1}^{K} \frac{\gamma_k^\top D^\top A_k D \gamma_k}{\gamma_k^\top D^\top B_k D \gamma_k} - \rho ||\gamma_k||_0$, the following lemma states the convergence of Algorithm 1.

Lemma 1. *Updating Γ and D by optimizing Eq. 6 using IRQM and Eq. 8 will monotonically increase the value of $\mathcal{L}(D, \Gamma)$, hence Algorithm 1 converges.*

Proof. By the convergence property of IRQM, the value sequence generated by IRQM is non-decreasing and converges to a stationary point of a equivalent problem of Eq. 6 (Song, Babu, and Palomar 2014). Therefore, we have $\mathcal{L}(D^{(t)}, \Gamma^{(t)}) \leq \mathcal{L}(D^{(t)}, \Gamma^{(t+1)})$. In addition, when using $D^{(t)}$ as a warm start for each dictionary update step, we have $\mathcal{L}(D^{(t)}, \Gamma^{(t+1)}) \leq \mathcal{L}(D^{(t+1)}, \Gamma^{(t+1)})$, hence $\mathcal{L}(D^{(t)}, \Gamma^{(t)}) \leq \mathcal{L}(D^{(t+1)}, \Gamma^{(t+1)})$. As we also assume that $\{B_k\} \in \mathbb{S}_{++}^d, \forall k = \{1, \ldots, K\}$, then $\mathcal{L}(D, \Gamma)$ is upper bounded, and the lemma holds. □

MTGEP for Entire Spectrum

The algorithm presented in above section only finds the largest eigenvalues (one per task) and corresponding eigenvectors. In this section, we show how to apply a *deflation* method based on the Lagrange multiplier algorithm (Bertsekas 1982) to solve the entire spectrum of multitask GEP.

Algorithm 2 Multitask Generalized Eigenvalue Program (MTGEP)

Input: $\{(A_1, B_1), \ldots, (A_K, B_K)\}$, number of generalized eigenvectors r,

1: $A_k^{(1)} = A_k, \forall k = \{1, \ldots, K\}$
2: **for** $i = 1, \ldots, r$ **do**
3: Solve $D^{(i)}$, $\Gamma^{(i)}$ and $W^{(i)}$ for $\{(A_1^{(i)}, B_1), \ldots, (A_K^{(i)}, B_K)\}$ using MTGEP-L algorithm.
4: Deflate $\{A_1^{(i)}, \ldots, A_K^{(i)}\}$ using Eq. 11.
5: **end for**

Output: $\mathbf{D} = \{D^{(1)}, \ldots, D^{(r)}\}$, $\mathbf{\Gamma} = \{\Gamma^{(1)}, \ldots, \Gamma^{(r)}\}$ and $\mathbf{W} = \{W^{(1)}, \ldots, W^{(r)}\}$.

Suppose we have already obtained $r-1$ eigenvectors $\{w_1, \ldots, w_{r-1}\}$, then the rth eigenvector of Eq. 1 can be obtained by solving the following constrained optimization problem:

$$\begin{aligned} w_r = \arg\max_w & \; w^\top A w \\ \text{s.t.} \quad & w_r^\top B w_r = 1, \\ & w_r^\top B w_i = 0, \quad \forall i = \{1, \ldots, r-1\}. \end{aligned} \tag{9}$$

By applying the method of Lagrange multiplier, Eq. 9 can be reformulated as the following GEP:

$$(I - B W_{r-1} W_{r-1}^\top) A w = \lambda B w, \tag{10}$$

where $W_{r-1} = [w_1, \ldots, w_{r-1}] \in \mathbb{R}^{d \times (r-1)}$. Let $A^{(1)} = A$, then Eq. 10 leads to the following deflation technique for A^r, $r = \{2, 3, \ldots\}$:

$$A^{(r)} = (I - B W_{r-1} W_{r-1}^\top) A, \tag{11}$$

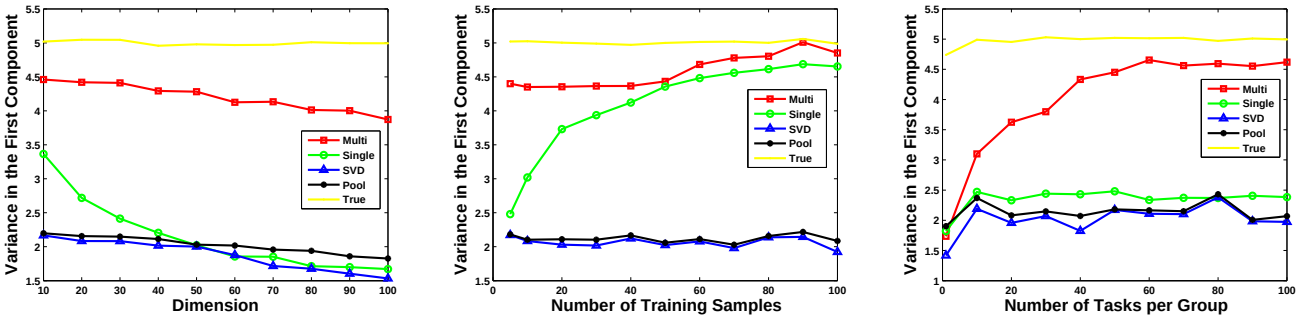
By the property of the Lagrange multiplier method, we immediately have the following proposition:

Proposition 1. *Let $\lambda_1 \leq \ldots \lambda_{r-1}$ be the $(r-1)$ largest eigenvalues of Eq. 1, and $w_1, \ldots, w_{r-1}$ be the corresponding eigenvectors, then the leading eigenvalue-eigenvector pair of Eq. 10 is the rth largest eigenvalue and corresponding eigenvector of Eq. 1. In addition, Eq. 10 has $(r-1)$ eigenvalues of zero, and the correspond eigenvectors are W_{r-1}.*

For the multitask GEP on round i, $i \in \{1, \ldots, r\}$, we assign a new dictionary D^i, based on the fact that the eigenvectors corresponding to different eigenvalues seldom lie in the same subspace, which is especially true for the case of PCA, where the eigenvectors are orthogonal to each other. Therefore, it is not necessary to force the eigenvectors corresponding to different eigenvalues to share the dictionary. In addition, this approach requires less computation for sparse coding and dictionary update at each iteration and can also avoid overfitting. The complete MTGEP algorithm is given in Algorithm 2.

Experiments

We first evaluate MTGEP in the context of multitask PCA (MultiPCA) using three synthetic data sets. We then test MTGEP in the FDA setting (MultiFDA) using three multitask



(a) (b) (c)

Figure 1: Learning performances with different settings of Synth. 1. (a): with different feature dimensions, (b): different number of training samples, (c): different number of tasks for each group.

classification benchmarks. Finally, we apply MTGEP to spatial filter design and propose MultiCSP for multi-subject EEG classification in the BCI application. In all experiments, the hyper-parameters (e.g., M, ρ) are selected by grid search and cross-validation.

MultiPCA for Multitask Dimensionality Reduction

Synth. 1: In the first synthetic dataset, we generate $G = 5$ disjoint groups of d-dimensional Gaussian distributed random variables. Within each group, we generate J tasks, each with 500 instances (N_{train} instances for training, the rest for testing). For all tasks, the leading eigenvalue of the covariance matrix is 5; remaining eigenvalues are sampled from a one-side normal distribution. Eigenvectors of the covariance matrices are randomly generated, and are the same within each group. We compare the average variance explained by leading eigenvectors found by MTGEP-L to these:

- *SinglePCA*: apply traditional PCA on each individual task.
- *PoolPCA*: apply traditional PCA jointly on all tasks.
- *SVDPCA*: the first column of the initial dictionary $D^{(0)}$ found by MTGEP.

We first set $J = 50$, $d = 30$, $N_{train} = 5$, and the variances explained by first principal component obtained by different approaches are reported in Table 1. We observe that MultiPCA significantly outperforms the other approaches. Fig. 1 shows how the learning performances of different algorithms vary with different settings. In Fig. 1(a) we set $J = 50$ and $N_{train} = 5$, and vary the dimension d of the samples. We observe that the performance of SinglePCA decreases as the feature dimension increases due to less reliable estimates of the principal component, while MultiPCA is robust to the increase in dimension. In Fig. 1(b) we set $J = 50$ and $d = 30$, and vary the amount of training data, N_{train}. We observe that MultiPCA significantly benefits from other tasks when the number of training instances for each task is small. Finally, we consider the performances of MultiPCA with different tasks for each group. We set $J = 50$, $N_{train} = 5$, $d = 30$, and vary J from 1 to 100. Fig. 1(c) shows that MultiPCA does not improve the learning performance when $J = 1$, since in this case there is no common knowledge to be shared among tasks. As the number of tasks per group increases, the performance of MultiPCA improves by leveraging the knowledge from other tasks within each group.

Table 1: Variance explained by each algorithm on synthetic data sets, with $d = 30, G = 5, J = 50, N_{train} = 5$ for Synth.1 and Synth.3, and $d = 20, G = 3, J = 50, N_{train} = 10$ for Synth.2.

	Synth. 1	Synth. 2	Synth. 3			
			1st	2nd	3rd	Total
Single	2.480	3.406	5.423	3.445	2.512	11.380
Pool	2.181	2.291	4.096	3.455	3.058	10.609
SVD	2.172	2.125	3.058	2.740	2.287	8.085
Multi	4.437	4.586	7.529	5.137	4.522	17.188

Synth. 2: The second dataset is generated using the approach proposed by (Kumar and Daumé III 2012). The goal is to investigate the effectiveness of MultiPCA on grouped tasks with shared structure. The dataset consists of $G = 3$ groups of datasets with $d = 20$ features, $J = 50$ for each group, and $N_{train} = 10$ per task. The leading eigenvectors are generated from 4 latent vectors randomly drawn from a Gaussian distribution with zero mean and identity covariance matrix. The leading eigenvectors of the first group are generated by linearly combining the first two latent vectors, with the coefficients combination for each task i.i.d. sampled from a normal distribution. Similarly, the leading eigenvectors of the second group are linear combinations of second and third latent vectors and the leading eigenvectors of the third group are linear combinations of the last two latent vectors.

The results shown in Table 1 confirm that MultiPCA works well on the tasks with overlapped structure. Fig. 2 illustrates the sparsity structure recovered by MultiPCA for Sythn. 1 and 2, showing that MultiPCA recovers most of the task structure for both disjoint and overlapped sparsity patterns.

Synth. 3: This is the same as Synth. 1, except that the first three leading eigenvalues for the covariance matrix of each

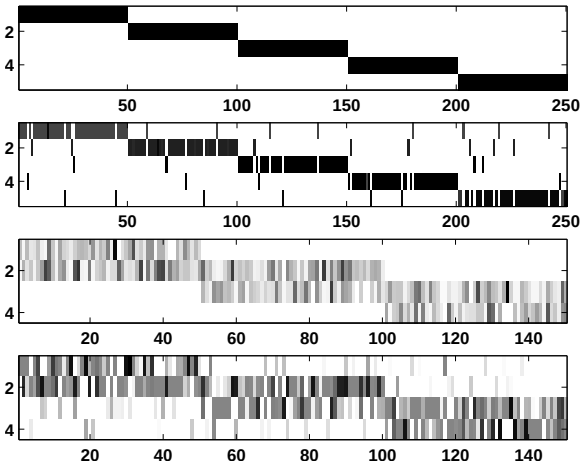


Figure 2: Sparsity recovery by MultiPCA. Top to bottom: True model of Synth. 1; recovered sparsity structure for Synth. 1; true model of Synth. 2; recovered sparsity structure for Synth. 2.

task are 9, 7, 5 respectively. We use the deflation approach in Algorithm 2 to find the first three principal components. Results are presented in Table 1. The sum of the first three variances explained by MultiPCA is 17.188, much larger than that of any other approaches, which highlights the effectiveness of MTGEP to extract multiple components.

MultiFDA for Multitask Classification

Next, we evaluate MultiFDA algorithm on three common multitask learning benchmarks: the *landmine* dataset (Xue et al. 2007), *USPS* and *MNIST* datasets (Kang, Grauman, and Sha 2011). For more detailed description of the datasets and experimental setting, see (Kang, Grauman, and Sha 2011; Ruvolo and Eaton 2013). Besides the baseline approach (SingleFDA), we also include comparison with the *grouping and overlapping multitask learning* (GO-MTL) algorithm (Kumar and Daumé III 2012), an existing multitask supervised learning approach. Table 2 summarizes the results. We see that MultiFDA outperforms single task learning and performs comparably to GO-MTL. We note that for the digit datasets, the improvements of the multitask learning approach over single task approach is not significant, which is consistent with previous analysis (Kang, Grauman, and Sha 2011; Kumar and Daumé III 2012).

MultiCSP for Multi-subject EEG Classification

Finally, we evaluate MTGEP for extracting multitask common spatial patterns (MultiCSP) in EEG signals. One benchmark dataset, dataset IIa from BCI competition IV[1] is used for performance evaluation. The dataset consists of EEG signals from 9 subjects who are instructed with visual cues to perform left hand, right hand, foot, and tongue motor imagery . In this study, only the EEG signals from the left hand and right hand motor imagery are used. The signals are recorded using 22 channels, sampled at 250 Hz, and bandpass-filtered in 0.5-100Hz. For each subject, the EEG signals consist of a training set and a test set, each containing 72 trials per EEG pattern. The main challenge of this problem is that the underlying task relatedness is unknown and the EEG data structure can be complex (Müller, Anderson, and Birch 2003). In our experiments, the EEG signals from 0.5 to 2.5 s after visual cue are used, and the data are further bandpass filtered to 5-30 Hz, since this time segment and frequency band include the signals involved in performing motor imagery. The first and last three eigenvectors of Eq. 3 are used as spatial filters, and then the logarithm of the variance of spatially filtered EEG signals are used as the input for FDA classification.

[1] http://www.bbci.de/competition/iv/.

Table 2: Results on multitask classification tasks: the area under the ROC curve (AUC) for the landmine dataset, and classification accuracy (%) for USPS and MNIST datasets.

	Landmine	USPS	MNIST
SingleFDA	74.9	90.9	89.1
MultiFDA	77.8	91.9	89.8
GO-MTL	78.0	92.8	86.6

The results in Table 3 show the that the multitask learning algorithms outperform single task learning approach for most subjects. In particular, the combination of MultiCSP and MultiFDA achieves the best performance, yielding an average improvement of 2.55% in classification accuracy. More important, it performs at least as well as single task learning approach for each subject. In other words, the learning performances of all the subject-specific spatial filters and classifiers benefit from knowledge shared between subjects.

We further investigate the effectiveness of MTGEP with insufficient data by varying the number of training samples per task, and the results are shown in Fig. 3. We can observe that the performance gap between single task learning approach and MTGEP is larger when fewer training samples are used. In other words, the estimation of covariance matrix, as well as between-class scatter matrix and within-class scatter matrix, suffer insufficient data problem, which deteriorates the learning performance. The less training samples are available, the more necessary it is to share the knowledge across the tasks, and the more significant the improvement of MTGEP is, as it alleviates the unreliable estimation problem, which justify the effectiveness of our algorithm.

Related Work

Multitask learning has been actively studied in recent years. While most previous work focuses on supervised learning, no existing work deals with the problem in the context of GEP. As the first attempt to solving multitask GEP, our work formulates this problem within the framework of sparse coding (Olshausen and Field 1996). In this section, we briefly review the literature that relates to sparse coding based transfer and multitask learning algorithms.

In the context of transfer learning, the self-taught learning

Table 3: Classification accuracy (%) of different algorithms for nine different subjects.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	Mean
CSP+FDA	90.28	52.08	93.75	65.28	51.39	62.50	79.17	89.58	88.19	74.69
CSP+MultiFDA	91.67	**59.03**	93.75	65.97	50.69	**65.97**	77.78	90.28	**88.89**	76.00
MultiCSP+FDA	**92.36**	56.25	93.75	65.97	50.69	63.89	81.25	92.36	88.19	76.08
MultiCSP+MultiFDA	**92.36**	56.94	93.75	**66.67**	**54.86**	65.28	**84.03**	**93.06**	88.19	**77.24**

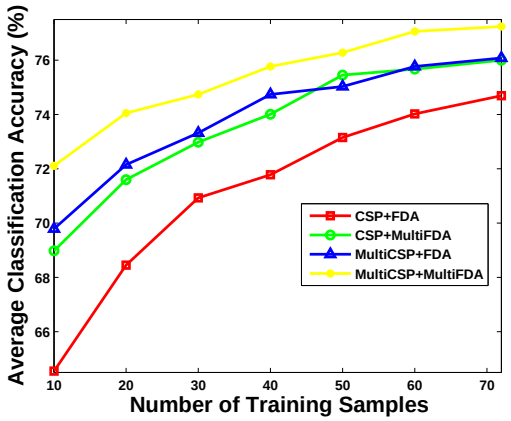


Figure 3: Learning performances with different number of training samples.

framework (Raina et al. 2007) applies sparse coding to construct higher level features using unlabeled data in source domain to improve the supervised learning performance in target domain. The seminal work encoding common knowledge into a dictionary via sparse coding for multitask learning is proposed in (Kumar and Daumé III 2012), where the model parameters of the multiple tasks are assumed to lie in a low dimensional subspace. Later, the generalization to transfer learning and the related theoretical analysis are presented in (Maurer, Pontil, and Romera-Paredes 2013; 2014). This method is applied for activity recognition (Zhou et al. 2013), with feature selection achieved by imposing ℓ_1 regularization on the dictionary. More recently, it is generalized to the context of lifelong learning (Ruvolo and Eaton 2013; 2014), and also applied to multitask reinforcement learning (Ammar et al. 2014).

Multitask learning for spatial filter and/or classifier design has also been studied in BCI community. The first attempt to utilizing the multitask learning framework to BCI design is presented in (Alamgir, Grosse-Wentrup, and Altun 2010), where the model parameters of each task are assumed to share a common Gaussian prior. By inferring the mean and covariance from all tasks jointly, the features extracted from brain signals of different subjects are interacted and the learning performances are improved. In (Devlaminck et al. 2011), the spatial filters of each subject are decomposed into the sum of a global and a subject-specific filter and the CSP algorithm is reformulated as a sum of regularized GEPs. More recently, (Samek, Kawanabe, and Muller 2014) has reformulated multitask CSP algorithm as a regularized divergence maximization problem, where the regularization term is the Kullback-Leibler (KL) divergence of different subjects. In (Kang and Choi 2014), a non-parametric Bayesian approach with Indian Buffet process priors is proposed for multitask CSP, where spatial patterns are modeled by an infinite latent feature model, assuming that a latent subspace is shared across subjects. While all of these methods are exclusively designed for the BCI application, MTGEP is a more general algorithm that can be applied to more scenarios.

While the structure we adopt to share knowledge across the tasks is similar to (Kumar and Daumé III 2012; Maurer, Pontil, and Romera-Paredes 2013), we emphasize that the learning contexts and optimization techniques are totally different. MTGEP is a new framework that jointly solves multiple generalized eigenvalue problems, rather than traditional multitask learning paradigm, which substantially enriches the possibilities for multitask learning.

Discussion

This paper introduces a new framework to solve multitask generalized eigenvalue problems, which can be used within a wide variety of machine learning approaches, such as multitask PCA, multitask FDA and more. The framework relies on the core assumption that the multitask problem can be captured by multiple GEPs whose eigenvectors lie in some subspace that can be represented by a set of shared basis vectors. We solve the resulting optimization problem via simultaneous sparse coding and dictionary learning. The proposed framework is validated within several task categories (both unsupervised and supervised) using both synthetic and real datasets. The empirical results show that solving related GEPs indeed benefits from our MTGEP approach, especially for GEPs with well grouped or overlapped structures.

The work opens up several avenues for future work. First, the MTGEP framework can also be extended for transfer learning and lifelong learning as in (Maurer, Pontil, and Romera-Paredes 2013; Ruvolo and Eaton 2013). Of particular interest is a deeper investigation of the use of MTGEP for lifelong EEG signal classification system in BCI research. In addition, there are promising directions for a theoretical analysis of the generalization bound of MTGEP based on Rademacher complexity (Bartlett and Mendelson 2002) and a recently proposed inequality for multitask dictionary learning (Maurer, Pontil, and Romera-Paredes 2014).

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) through the Discovery

Grants Program and the NSERC Canadian Field Robotics Network (NCFRN), as well as by the Fonds de Recherche du Quebec Nature et Technologies (FQRNT).

References

Alamgir, M.; Grosse-Wentrup, M.; and Altun, Y. 2010. Multitask learning for brain-computer interfaces. In *AISTATS*.

Ammar, H. B.; Eaton, E.; Ruvolo, P.; and Taylor, M. E. 2014. Online multi-task learning for policy gradient methods. In *ICML*.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Machine Learning* 73(3):243–272.

Bartlett, P. L., and Mendelson, S. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3:463–482.

Bertsekas, D. P. 1982. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press.

Bezdek, J. C., and Hathaway, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel and Scientific Computations* 11:351–368.

Bie, T. D.; Cristianini, N.; and Rosipal, R. 2005. Eigenproblems in pattern recognition. In *Handbook of Geometric Computing*. 129–170.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. New York: Springer.

Blankertz, B.; Tomioka, R.; Lemm, S.; Kawanabe, M.; and Müller, K.-R. 2008. Optimizing spatial filters for robust eeg single-trial analysis. *IEEE Signal Processing Magazine* 25(1):41–56.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Devlaminck, D.; Wyns, B.; Grosse-Wentrup, M.; Otte, G.; and Santens, P. 2011. Multisubject learning for common spatial patterns in motor-imagery BCI. *Computational Itelligence and Neuroscience* 2011:1–9.

Evgeniou, T.; Micchelli, C. A.; and Pontil, M. 2005. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research* 6:615–637.

Jolliffe, I. 2002. *Principal Component Analysis*. New York: Springer, 2nd edition.

Kang, H., and Choi, S. 2014. Bayesian common spatial patterns for multi-subject EEG classification. *Neural Networks* 57:39–50.

Kang, Z.; Grauman, K.; and Sha, F. 2011. Learning with whom to share in multi-task feature learning. In *ICML*.

Karampatziakis, N., and Mineiro, P. 2014. Discriminative features via generalized eigenvectors. In *ICML*.

Kumar, A., and Daumé III, H. 2012. Learning task grouping and overlap in multi-task learning. In *ICML*.

Lotte, F.; Congedo, M.; Lécuyer, A.; Lamarche, F.; and Arnaldi, B. 2007. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering* 4(2):R1–R13.

Maurer, A.; Pontil, M.; and Romera-Paredes, B. 2013. Sparse coding for multitask and transfer learning. In *ICML*.

Maurer, A.; Pontil, M.; and Romera-Paredes, B. 2014. An inequality with applications to structured sparsity and multitask dictionary learning. In *COLT*.

Moghaddam, B.; Weiss, Y.; and Avidan, S. 2006. Generalized spectral bounds for sparse LDA. In *ICML*.

Müller, K.-R.; Anderson, C. W.; and Birch, G. E. 2003. Linear and nonlinear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11(2):165–169.

Olshausen, B. A., and Field, D. J. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381(6583):607–609.

Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. Y. 2007. Self-taught learning: Transfer learning from unlabeled data. In *ICML*.

Ramoser, H.; Müller-Gerking, J.; and Pfurtscheller, G. 2000. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transaction on Rehabilitation Engineering* 8(4):441–446.

Ruvolo, P., and Eaton, E. 2013. ELLA: An efficient lifelong learning algorithm. In *ICML*.

Ruvolo, P., and Eaton, E. 2014. Online multi-task learning via sparse dictionary optimization. In *AAAI*.

Samek, W.; Kawanabe, M.; and Muller, K.-R. 2014. Divergence-based framework for common spatial patterns algorithms. *IEEE Reviews in Biomedical Engineering* 7:50–72.

Song, J.; Babu, P.; and Palomar, D. P. 2014. Sparse generalized eigenvalue problem via smooth optimization. *arXiv preprint arXiv:1408.6686*.

Sriperumbudur, B. K.; Torres, D. A.; and Lanckriet, G. R. G. 2007. Sparse eigen methods by d.c. programming. In *ICML*.

Wolpaw, J. R.; Birbaumer, N.; McFarland, D. J.; Pfurtscheller, G.; and Vaughan, T. M. 2002. Brain–computer interfaces for communication and control. *Clinical Neurophysiology* 113(6):767–791.

Xue, Y.; Liao, X.; Carin, L.; and Krishnapuram, B. 2007. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research* 8:35–63.

Zhou, Q.; Wang, G.; Jia, K.; and Zhao, Q. 2013. Learning to share latent tasks for action recognition. In *ICCV*.