Sparse Multiple Kernel Learning for Signal Processing Applications

Niranjan Subrahmanya and Yung C. Shin

Abstract—In many signal processing applications, grouping of features during model development and the selection of a small number of relevant groups can be useful to improve the interpretability of the learned parameters. While a lot of work based on linear models has been reported to solve this problem, in the last few years, multiple kernel learning has come up as a candidate to solve this problem in nonlinear models. Since all of the multiple kernel learning algorithms to date use convex primal problem formulations, the kernel weights selected by these algorithms are not strictly the sparsest possible solution. The main reason for using a convex primal formulation is that efficient implementations of kernel-based methods invariably rely on solving the dual problem. This work proposes the use of an additional log-based concave penalty term in the primal problem to induce sparsity in terms of groups of parameters. A generalized iterative learning algorithm, which can be used with a linear combination of this concave penalty term with other penalty terms, is given for model parameter estimation in the primal space. It is then shown that a natural extension of the method to nonlinear models using the "kernel trick" results in a new algorithm, called Sparse Multiple Kernel Learning (SMKL), which generalizes group-feature selection to kernel selection. SMKL is capable of exploiting existing efficient single kernel algorithms while providing a sparser solution in terms of the number of kernels used as compared to the existing multiple kernel learning framework. A number of signal processing examples based on the use of mass spectra for cancer detection, hyperspectral imagery for land cover classification, and NIR spectra from wheat, fescue grass, and diesel are given to highlight the ability of SMKL to achieve a very high accuracy with a very few kernels.

Index Terms—Composite kernel learning, feature group selection, heterogeneous data fusion, sensor selection.

1 Introduction

Fusion of data obtained from multiple sources to improve prediction accuracy and reliability has been widely recognized as an important problem in machine learning. Grouping of features belonging to the same source during data fusion allows an easy interpretation of the importance of various sources using parameters of the learned model. For example, "feature-level sensor fusion" involves the extraction and integration of high level features from raw sensor data [1] and hence feature grouping allows the estimation of the importance of different sensors. Feature grouping is also useful for wavelength selection in spectral chemometric applications [2], [3]. In such applications, grouping of contiguous wavelengths into bands and selection of a small number of bands provide information about the regions of the spectrum which are sensitive to the target being monitored, whereas selection of individual wavelengths could result in the selected wavelengths being spread throughout the spectrum, thus losing interpretability. As the technology related to sensor development and data acquisition is improving, many other applications of feature-grouping-based interpretable model construction are constantly coming up, such as selection of a group of genes belonging to a specific pathway in microarray analysis or the analysis of the importance of spectral and spatial variance for hyperspectral image classification to name a few.

Sparsity or the selection of a small number of feature groups during data fusion is a very critical aspect because of the following reasons.

1. In applications like multisensor data fusion, each group selected corresponds to the use of another sensor and hence requires additional cost for installation and maintenance.
2. For spectral chemometric applications, although it may be possible to obtain accurate models using the entire feature set using techniques such as Partial Least Squares Regression (PLSR) [4] or Support Vector Machines (SVM) [5], the selection of the smallest number of bands is essential to retain interpretability of the model in terms of the sensitivity of the spectrum to the target being monitored.
3. Exclusion of irrelevant feature groups is expected to increase prediction accuracy.
4. A reduction in time required for online implementation is important for applications where this is critical. For example, in some spectroscopic applications, there may be a time associated with the acquisition of data for each band.

The problem of group-feature selection is especially tricky when only a few data points are available and a large number of features, which may be highly correlated to each other, are present in each group. This is usually the case in many sensor fusion and spectral chemometric applications. An extension of the popularly used filter techniques for feature selection based on correlation [6] or mutual information [6] to group-feature selection is severely affected in such

- *The authors are with the School of Mechanical Engineering, Purdue University, 585 Purdue Mall, West Lafayette, IN 47907. E-mail: {nsubrahm, shin}@purdue.edu.*

Manuscript received 20 Oct. 2008; revised 2 Apr. 2009; accepted 22 Apr. 2009; published online 29 Apr. 2009.
Recommended for acceptance by N. Lawrence.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-10-0726.
Digital Object Identifier no. 10.1109/TPAMI.2009.98.

cases. Moreover, these methods generally require manual tuning of the optimal number of groups in addition to the tuning of hyperparameters for model training. Hence, this work draws on some of the insights gained with embedded feature selection methods, in particular the effect of sparsity promoting penalty terms, to provide a systematic solution targeted at selecting an optimal number of feature groups with minimal manual intervention.

Embedded selection methods have been successfully used for the problem of combined feature selection and model learning by exploiting information about specific learning methods [7], [8]. For linear models (models of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$), assuming the feature values to be normalized, the model parameters $\mathbf{w}$ may be considered as scaling factors as each feature is multiplied by the corresponding component of $\mathbf{w}$ to get the term $\mathbf{w}^T\mathbf{x}$, making the magnitude of $\mathbf{w}_i$ indicative of the importance of the ith feature. Using the so called "one norm" or $\ell_1(\mathbf{w})$, which is equal to the sum of the absolute values of the elements of $\mathbf{w}$, for regularization, has been shown to result in sparse solutions for $\mathbf{w}$ while ensuring that the resulting optimization problem remains convex [9], [10], [11]. The use of concave regularization terms, such as the ones used in the Feature Selection Concave (FSV) [10], AROM [12], Relevance Vector Machines (RVM) [13], and the approach using Jeffrey's prior in [11], on the other hand, have been observed to give sparser solutions in spite of the presence of many local minima [10], [12].

Recently, there have been some attempts to extend feature selection methods to handle feature group selection. Extensions of filter-based methods would have the same disadvantage as they do for feature selection, namely, they do not exploit the characteristics of the learning algorithm used and they require an additional round of cross validation. Hence, we focus on extensions of embedded selection methods in this survey. Many such approaches, which are suitable for linear models, have been proposed to date in the machine learning as well as signal processing literature [14], [15], [16], [17], [18], [19], [20], [21]. A particular approach that has been gaining popularity lately is the "group lasso," which generalizes the lasso penalty [9] for feature selection to group-feature selection. This was originally proposed in [18] as a solution to variable selection for regression when categorical variables also need to be considered. It has since been used for grouping of variables in many applications such as feature selection for multi-output regression [16], micorarray data analysis [22], and for logistic regression [23]. It has been observed that, although the group lasso results in setting the weights of many groups to zero, it does not return the smallest possible set of groups that are sufficient to obtain an accurate model [17]. The M-FOCUSS algorithm [21], which covers the group lasso as a special case, does consider concave penalty terms using ℓ_p, $0 \leq p \leq 1$, norms for the groups and has been reported to give sparser solutions than the group lasso. We acknowledge here that there are a number of other, similar works in the areas of machine learning, sparse signal approximation, and statistics which have not been mentioned here but can be found in the references of the works mentioned above.

However, most of the methods mentioned so far deal with obtaining sparse solutions based on linear models. While it is possible to extend many of the concepts used in the above mentioned methods to nonlinear models, ad hoc combinations of grouping strategies and nonlinear models would not result in efficient implementations. In this work, we focus on a particular class of nonlinear models based on the framework of Multiple Kernel Learning (MKL) [24], which provides a more generic method for selecting groups of features than linear-model-based techniques while simultaneously retaining computational feasibility. MKL has been shown to be effective for fusion of heterogeneous data from different sources. MKL generalizes feature/feature-group selection to kernel selection in kernel-based methods, which make use of a so-called kernel function that defines the similarity between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$. The framework of MKL proposed by Lanckriet et al. [24] suggests a way of learning this kernel function from data wherein positive-semidefinite composite kernels of the form $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{K} \beta_k \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$ are obtained from primitive kernels, $\mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$. It is easy to see that if the primitive kernels are calculated using different groups of features, then their corresponding weights β_k may be considered as being indicative of the importance of that feature group. Algorithms for training such models with a specific regularization term, which turns out to be equivalent to the use of a group lasso in the linear setting, have been proposed in [25], [26], [27], [28]. All of the methods to date have restricted themselves to this convex regularizer (or equivalent ones) as kernel-based methods generally solve the dual problem by exploiting the convexity of the primal problem, but, as expected based on the results from the linear methods, these regularizers do not select the sparsest possible solution for the kernel weights.

In this paper, the development of a novel multiple kernel learning scheme is approached by starting with a linear model-based embedded method for selecting groups of features by using a suitable penalty term called a Group Selection Term (GST). This term is allowed to be a linear combination of a certain allowable set of regularization terms. A simple method for deriving upper bounds with a diagonal quadratic form for the GST is presented and a bound optimization approach [29] based on surrogate functions employing a combination of these quadratic bounds and the loss function is suggested for parameter estimation. A natural extension of this method using the "kernel trick" results in a generalized composite kernel learning algorithm, which is termed SMKL and is applicable to a wide range of loss functions, such as squared loss, hinge loss [5], epsilon-insensitive loss [5], and logistic loss [30]. SMKL gives notably sparser solutions than the existing MKL framework in terms of the number of kernels used while maintaining or improving prediction accuracy. It may be noted that quadratic bounds, which may not always be diagonal, have been used successfully in other works to optimize over nonsmooth functions or approximate intractable integrals when dealing with $\ell_1(\mathbf{w})$ penalty [30], [31], log penalty [11], and logistic loss [30], [32]. A significant advantage of using diagonal quadratic upper bounds in this work is that it not only makes a nonsmooth optimization problem simpler, but also decouples the interaction between parameters that is introduced by the GST, which allows the application of the "kernel trick." The significant contributions of this work include the extension of the use of concave penalty terms in nonlinear kernel-based methods and the development of a novel kernel-based algorithm (SMKL) for learning the kernel and model parameters with various loss functions. This has significant implications in allowing the use of nonlinear kernel-based models for sparse signal reconstruction and knowledge discovery.

Section 2 presents the idea of GSTs and gives a generalized parameter estimation algorithm for linear models, i.e., in the primal space. In Section 3, an extension of the grouping technique to a class of nonlinear models using the "kernel trick" is developed and the SMKL algorithm is given. Section 4 presents some experimental results while the conclusions are given in Section 5.

2 Group Selection Terms and Parameter Estimation

Given a training set $\mathbf{D} = \{(\mathbf{x}_i, y_i) \in \chi \times \{-1, 1\} : i = 1, \ldots, n\}$ for binary classification or $\mathbf{D} = \{(\mathbf{x}_i, y_i) \in \chi \times \mathbb{R} : i = 1, \ldots, n\}$ for regression with $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]^T \in \chi \subset \mathbb{R}^d$, the goal of supervised learning is to learn a function $y = f(\mathbf{x})$ which not only recalls this information but also generalizes well. Assume that the features of the data set are generated by different sources. Multiple features may be extracted from each source and each source could be a sensor, a specific band from a spectrum or a predefined group of features. Assume that a total of d features are extracted from m sources such that each feature belongs to one and only one source. This is not a strict requirement but makes the notation simpler. Let $\mathbf{s}_k \in \mathbb{N}^{d_k}$ denote the feature set of source k. Therefore, $\sum_{k=1}^m d_k = d$ and $\mathbf{s}_{k1} \cap \mathbf{s}_{k2} = \emptyset \; \forall k1 \neq k2$. Let $S \subset \{1, \ldots, m\}$ denote a subset of the sources. Therefore, the goal is to minimize the cardinality of S while retaining good generalization capability for $y = f(\mathbf{x}^S)$, where $\mathbf{x}^S$ is a vector with features extracted from sources belonging to S. GSTs that induce this goal are now introduced in the next section.

2.1 Group Selection Terms

Let $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_d]^T \in \mathbb{R}^d$ represent a vector of scaling factors for the d features. Assume that the grouping information based on the sources can be introduced by dividing $\mathbf{w}$ into m groups, $\mathbf{w} = [\mathbf{w}^1, \ldots \mathbf{w}^k, \ldots \mathbf{w}^m]$, $\mathbf{w}^k = \{\mathbf{w}_j : j \in \mathbf{s}_k\}$ and using a suitable regularization term that exploits the grouping information. Some of the terms which have already been used for this purpose include the group lasso [18]

$$\left(\sum_{k=1}^{m} \eta_k \sqrt{\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right)$$

and a more generalized version of this used in the M-FOCUSS [21]

$$\left(\sum_{k=1}^{m} \eta_k \left(\sqrt{\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right)^p, 0 \leq p \leq 1 \right),$$

where η_k is a term that has been introduced to control for varying group sizes. Since the use of a log-based penalty in place of $\ell_p(\mathbf{W})$ norms has been shown to be effective in the feature selection scenario [12], we consider the use of a similar regularization term over the groups,

$$\sum_{k=1}^{m} \eta_k \log \left(\sqrt{\varepsilon + \sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right),$$

in order to increase sparsity of the obtained solution. Here, ε is a small, positive nonzero term used to prevent the log term from becoming unbounded below as $\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2 \to 0$. A similar term for feature selection has been found to give good results over a broad range of applications [12], [33]; it is felt that choosing this term is a better option than trying to tune the parameter p for the generic M-FOCUSS algorithm (using this term is actually equivalent to fixing p at zero in the M-FOCUSS algorithm [34]). Finally, recognizing the fact that the use of concave terms alone in certain scenarios can reduce prediction accuracy by overemphasizing sparseness and noting that using a combination of convex and concave penalty terms helps to retain the desirable properties of sparseness and good generalization [35], [36], the GST is allowed to be a linear combination of terms from Table 1.

TABLE 1
Sparsity Promoting Terms and Corresponding Parameters for Quadratic Bounds

Sl No.	Name	$g(\mathbf{w})$	$C_j(\mathbf{w}) = \frac{1}{\mathbf{w}_j} \frac{\partial g}{\partial \mathbf{w}_j}$, $\mathbf{w}_j > 0, j \in \mathbf{s}_l$
1.	$GST_{Log}(\mathbf{w})$	$\sum_{k=1}^{m} \eta_k \log \left(\sqrt{\varepsilon + \sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right)$	$\frac{\eta_l}{\left(\varepsilon + \sum_{j \in \mathbf{s}_l} \mathbf{w}_j^2 \right)}$
2.	$GST_{GL}(\mathbf{w})$ [18]	$\sum_{k=1}^{m} \eta_k \sqrt{\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2}$	$\eta_l \Big/ \sqrt{\sum_{j \in \mathbf{s}_l} \mathbf{w}_j^2}$
3.	$GST_{MF}(\mathbf{w})$ [21]	$\sum_{k=1}^{m} \eta_k \left(\sqrt{\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right)^p$	$\eta_l p \Big/ \left(\sum_{j \in \mathbf{s}_l} \mathbf{w}_j^2 \right)^{\frac{p}{2}-1}$
4.	$GST_{MKL}(\mathbf{w})$ [25]	$\frac{1}{2} \left(\sum_{k=1}^{m} \eta_k \sqrt{\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right)^2$	$\eta_l \left(\sum_{k=1}^{m} \eta_k \sqrt{\sum_{j \in \mathbf{s}_k} \mathbf{w}_j^2} \right) \Big/ \sqrt{\sum_{j \in \mathbf{s}_l} \mathbf{w}_j^2}$

$g(\mathbf{w})$ gives the functional form of the penalty term. C_j are the parameters used to get a diagonal quadratic upper bound for $g(\mathbf{w})$. Assume that the $C_j(\mathbf{w})$ being calculated is for the lth group. The subscripts of the GSTs denote their source: GL refers to the Group Lasso, MF refers to the M-FOCUSS, and MKL refers to the original Multiple Kernel Learning formulation.

An immediate concern when looking at the proposed log-based concave penalty term (first entry in Table 1) might be that although $GST_{Log}(\mathbf{w})$ has been prevented from becoming unbounded below when any $\mathbf{w}^k = 0$ (by using the additional positive term ε), $\mathbf{w} = 0$ could still be the globally optimal trivial solution to the parameter estimation problem irrespective of the loss function used. We would like to remind the reader that, since the optimization problem becomes concave in the presence of $GST_{Log}(\mathbf{w})$, there are a number of other valid locally optimal solutions. Our goal is to arrive at one such solution and our algorithm does this by using an iterative local search method, which uses a "good" starting point. In the future, when referring to parameter estimates with $GST_{Log}(\mathbf{w})$ as the penalty term, it is this locally optimal nontrivial solution that is being referred to and not the possibly trivial solution $\mathbf{w} = 0$.

While the theoretical possibility of converging to a "bad" or trivial local minimum remains, in practice it was observed that good solutions were always obtained on the first run starting from an equally weighted initial composite kernel.

2.2 Parameter Estimation Using Surrogate Function Minimization

In this section, an algorithm for parameter estimation using the bound optimization approach [29] is described. It may be noted that all of the entries in Table 1, except for $GST_{MKL}(\mathbf{w})$, satisfy the following conditions: $GST_{MKL}(\mathbf{w})$ does not satisfy condition 3.

1. $g(\mathbf{w})$ may be written as $G(\mathbf{r})$ where $\mathbf{r}$ is an m-dimensional vector with $\mathbf{r}_k = \sum_{j\in \mathbf{s}_k} \mathbf{w}_j^2$.
2. $g(\mathbf{w}) = g(|\mathbf{w}|)$ where $|\mathbf{w}|$ is obtained by taking the absolute value of each element of $\mathbf{w}$.
3. $G(\mathbf{r})$ is smooth, concave, and monotonically increasing in the positive "quadrant."

For any $g(\mathbf{w})$ which satisfies the above conditions, a diagonal quadratic function of the form shown in (1) can be constructed at any estimate $\hat{\mathbf{w}}$ belonging to the domain of $g(\mathbf{w})$. Here, $g^u(\mathbf{w} \mid \hat{\mathbf{w}})$ denotes a function of $\mathbf{w}$ whose parameters depend on the current estimate ($\hat{\mathbf{w}}$) as shown in (1). $g^u(\mathbf{w} \mid \hat{\mathbf{w}})$ gives an upper bound for $g(\mathbf{w})$, i.e., $g(\mathbf{w}) \le g^u(\mathbf{w} \mid \hat{\mathbf{w}})$, and also satisfies $g(\hat{\mathbf{w}}) = g^u(\hat{\mathbf{w}} \mid \hat{\mathbf{w}})$ and $\frac{\partial g(\mathbf{w})}{\partial \mathbf{w}}|_{\hat{\mathbf{w}}} = \frac{\partial g^u(\mathbf{w}|\hat{\mathbf{w}})}{\partial \mathbf{w}}|_{\hat{\mathbf{w}}}$ (can be verified directly from (1)).

$$\begin{aligned} g^u(\mathbf{w} \mid \hat{\mathbf{w}}) &= \sum_{j=1}^{d} \left(\frac{1}{2\mathbf{w}_j} \frac{\partial g}{\partial \mathbf{w}_j}\bigg|_{\hat{\mathbf{w}}} \right) \left(\mathbf{w}_j^2 - \hat{\mathbf{w}}_j^2 \right) + g(\hat{\mathbf{w}}) \\ &= C_0 + \frac{1}{2} \sum_{j=1}^{d} C_j \mathbf{w}_j^2, \end{aligned} \tag{1}$$

Here, $C_j = \frac{1}{\mathbf{w}_j} \frac{\partial g}{\partial \mathbf{w}_j}|_{\hat{\mathbf{w}}}$ and $C_0 = g(\hat{\mathbf{w}}) - \sum_{j=1}^{d} (\frac{1}{2\mathbf{w}_j} \frac{\partial g}{\partial \mathbf{w}_j}|_{\hat{\mathbf{w}}}) \hat{\mathbf{w}}_j^2$. The specific values of C_j for the penalty terms under consideration are given in the last column of Table 1. It may be shown that $g^u(\mathbf{w} \mid \hat{\mathbf{w}})$, derived using (1) for $GST_{MKL}(\mathbf{w})$, also has these properties, even though $GST_{MKL}(\mathbf{w})$ does not satisfy the last condition. If $g(\mathbf{w})$ is a linear combination of terms from Table 1, it is easy to see that the corresponding C_j values may be obtained by a linear combination of the individual C_j values for each term.

These bounds are now utilized to derive a generalized algorithm using optimization transfer with surrogate functions [29]. Let $L(\mathbf{w})$ be the function to be minimized and let $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ be an upper bound for $L(\mathbf{w})$. The notation denotes that $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ is a function of $\mathbf{w}$ whose parameters depend on $\hat{\mathbf{w}}^t$, the estimate of $\mathbf{w}$ at time t. Assuming that $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ is closest to $L(\mathbf{w})$ at $\mathbf{w} = \hat{\mathbf{w}}^t$, i.e., $(L(\mathbf{w}) - Q(\mathbf{w} \mid \hat{\mathbf{w}}^t))$ attains its maximum at $\mathbf{w} = \hat{\mathbf{w}}^t$, it can be shown that minimization of $L(\mathbf{w})$ can be done by successively minimizing $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ and using an iterative parameter update law given by $\hat{\mathbf{w}}^{t+1} = \arg\min_{\mathbf{w}} Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$. It can be shown that, with this update law, the value of $L(\hat{\mathbf{w}}^t)$ is guaranteed to decrease monotonically as the number of iterations increases. This may be seen by noting that

- $Q(\hat{\mathbf{w}}^{t+1} \mid \hat{\mathbf{w}}^t) \le Q(\hat{\mathbf{w}}^t \mid \hat{\mathbf{w}}^t)$ (since $\hat{\mathbf{w}}^{t+1}$ minimizes $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$).
- $L(\hat{\mathbf{w}}^{t+1}) - Q(\hat{\mathbf{w}}^{t+1} \mid \hat{\mathbf{w}}^t) \le L(\hat{\mathbf{w}}^t) - Q(\hat{\mathbf{w}}^t \mid \hat{\mathbf{w}}^t)$ (from the above assumption that $(L(\mathbf{w}) - Q(\mathbf{w} \mid \hat{\mathbf{w}}^t))$ attains its maximum at $\mathbf{w} = \hat{\mathbf{w}}^t$).

Therefore, using the above two inequalities,

$$\begin{aligned} L(\hat{\mathbf{w}}^{t+1}) &= L(\hat{\mathbf{w}}^{t+1}) - Q(\hat{\mathbf{w}}^{t+1} \mid \hat{\mathbf{w}}^t) + Q(\hat{\mathbf{w}}^{t+1} \mid \hat{\mathbf{w}}^t) \\ &\le L(\hat{\mathbf{w}}^t) - Q(\hat{\mathbf{w}}^t \mid \hat{\mathbf{w}}^t) + Q(\hat{\mathbf{w}}^t \mid \hat{\mathbf{w}}^t) \\ &= L(\hat{\mathbf{w}}^t). \end{aligned}$$

This algorithm enjoys the same local convergence properties as the Expectation Maximization algorithm [37]. The interested reader is referred to [29] for more details about the algorithm.

Let $l(\mathbf{x}_i, y_i, \mathbf{w})$ be a loss function which determines the training error of the ith data point for a model with parameter vector $\mathbf{w}$ and μ be a regularization parameter which determines the trade-off between regularization and training error minimization. The functions $L(\mathbf{w})$ and $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ are now defined as

$$L(\mathbf{w}) = g(\mathbf{w}) + \mu \sum_i l(\mathbf{x}_i, y_i, \mathbf{w}), \tag{2}$$

$$Q(\mathbf{w} \mid \hat{\mathbf{w}}^t) = g^u(\mathbf{w} \mid \hat{\mathbf{w}}^t) + \mu \sum_i l(\mathbf{x}_i, y_i, \mathbf{w}). \tag{3}$$

From (1)-(3), $L(\mathbf{w}) - Q(\mathbf{w} \mid \hat{\mathbf{w}}^t) = g(\mathbf{w}) - g^u(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ attains its maximum value of zero at $\mathbf{w} = \hat{\mathbf{w}}^t$, thus satisfying the conditions for optimization transfer. $L(\mathbf{w})$ is the objective function to be minimized for embedded group-feature selection with $g(\mathbf{w})$ being any linear combination of GSTs from Table 1. By construction, $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ has a diagonal quadratic penalty term, and hence the update step $\hat{\mathbf{w}}^{t+1} = \arg\min_{\mathbf{w}} Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ can be carried out using an existing Quadratic Regularizer-based Learning Algorithm (*QRLA*) for training a model with a diagonal quadratic regularizer and the selected loss function. For example, the *QRLA* for a squared loss term could be the regularized least squares algorithm. Thus, the use of the proposed quadratic upper bounds for GSTs along with optimization transfer to $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ allows the minimization of $L(\mathbf{w})$ for any loss term, for which a *QRLA* is available. This includes the squared loss, logistic loss, hinge loss, and the epsilon-insensitive loss to name a few. The generalized algorithm for group-feature selection is now given in Algorithm 1. The convergence of Algorithm 1 follows directly from the convergence of the optimization transfer algorithm [29].

Algorithm 1. *The algorithm estimates parameters when a GST is used as the penalty term by iteratively calling a standard QRLA.*

```
Set $t = 0$. Initialize $\hat{\mathbf{w}}^0 = \mathbf{1}$. Select an appropriate GST
from Table 1.
while (true)
    Compute $C_j(\hat{\mathbf{w}}^t)$ using Table 1
    Obtain $Q(\mathbf{w}|\hat{\mathbf{w}}^t)$ using (1) and (3)
    $\hat{\mathbf{w}}^{t+1} = \arg\min_{\mathbf{w}} Q(\mathbf{w}|\hat{\mathbf{w}}^t)$(obtained using QRLA)
    if $\|\hat{\mathbf{w}}^{t+1} - \hat{\mathbf{w}}^t\|_1 \le tol$
        break
    endif
    $t = t + 1$
end while
```

The use of the bound optimization approach, as opposed to a constraint-based optimization approach, has two significant advantages here. First, it provides a systematic way to introduce feature grouping into existing *QRLA*s, which do not have the capability to include grouping information by themselves, using GSTs from Table 1. Second, it allows the extension of the algorithm to nonlinear models using the kernel trick as described in the next section. At this point, it may be noted that Algorithm 1 looks very much like the M-FOCUSS algorithm [21], but has the advantage that it does not explicitly consider a loss function, whereas the M-FOCUSS algorithm considers only the squared loss function. This is important in the context of kernel-based applications as the most successful and efficient implementations of the kernel-based methods are based on the hinge loss (Support Vector Machines) and the epsilon-insensitive loss (Support Vector Regressors), and the main aim of developing Algorithm 1 is to extend it to composite kernel learning in kernel-based models while efficiently utilizing existing single kernel algorithms. The M-FOCUSS algorithm can be derived as a specific case of Algorithm 1 while using the squared loss and using a matrix-inversion-based least squares solution as the resulting QRLA.

3 Sparse Multiple Kernel Learning

A straightforward, but computationally expensive way of extending the proposed method to handle nonlinear models is to make use of the GST as a penalty term for the scaling factors in feature scaling kernels [35], [38]. A more elegant way of extending Algorithm 1 to nonlinear settings is through the use of the "kernel trick." Kernel-based methods find models of the form $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \mathbf{k}(\mathbf{x}, \mathbf{x}_i) + b$ for a prespecified kernel $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$. The performance of such models depends critically on the definition of the kernel. The idea behind the framework of MKL proposed by Lanckriet et al. [24] is to learn a composite kernel of the form $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{K} \beta_k \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$ from the data, where $\mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$ are predefined kernels and β_k are parameters. If each individual kernel, $\mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$, is constructed using a different group of features, then choosing a kernel is equivalent to choosing the corresponding group. Therefore, kernel selection may be seen as a generalization of the group-feature selection framework if it is possible to obtain a sparse solution for β without sacrificing prediction accuracy.

In [25], the primal problem for MKL (binary classification) is formulated as shown in (4), where each $\mathbf{x}_i$ is translated via K mappings $\Phi_k(\mathbf{x}) \mapsto \mathbb{R}^{d_k}$, $k = 1, 2, \ldots, \mathrm{K}$, from the input space to the combined feature space $\Phi(\mathbf{x}) = (\Phi_1(\mathbf{x}), \ldots, \Phi_K(\mathbf{x}))$, d_k denotes the dimensionality of the kth feature space and ξ_is denote the slack variables for the constraints. The term $\frac{1}{2}\left(\sum_{k=1}^{K} \|\mathbf{w}^k\|_2\right)^2$ in (4) (which may be recognized as $GST_{MKL}(\mathbf{w})$ from Table 1) was chosen to make (4) a convex programming problem, thus allowing optimization of its dual and the application of the "kernel trick." This is the approach taken in [25] and [26]. Here, a different approach is adopted and the optimization algorithm is derived in the primal space by a direct application of Algorithm 1. This allows the use of the concave penalty terms such as $GST_{Log}(\mathbf{w})$, unlike the methods mentioned previously. This should result in the achievement of the selection of a sparser number of groups. The relationship between $\mathbf{w}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ will be made clear in the coming section and it will be evident that the selection of a small number of groups in terms of $\mathbf{w}$ is equivalent to obtaining a sparse solution for $\boldsymbol{\beta}$, which is one of the main goals of this work because of the advantages mentioned previously.

$$\begin{aligned} &\min \frac{1}{2}\left(\sum_{k=1}^{K} \|\mathbf{w}^k\|_2\right)^2 + C\sum_{i=1}^{N} \xi_i, \\ &\mathbf{w}^k \in \mathbb{R}^{d_k}, \boldsymbol{\xi} \in \mathbb{R}^N, b \in \mathbb{R}, \\ &\text{such that } \xi_i \geq 0 \text{ and} \\ &\quad y_i\left(\sum_{k=1}^{K} \langle \mathbf{w}^k, \Phi_k(\mathbf{x}_i)\rangle + b\right) \geq 1 - \xi_i, \forall i = 1, \ldots, N. \end{aligned} \tag{4}$$

In the following section, it is shown that Algorithm 1 is directly applicable to the problem of composite kernel learning because of the decoupling introduced by the diagonal quadratic upper bounds. Although the following derivation considers the problem of the hinge loss for binary classification, it should be clear that SMKL is easily applicable to other loss functions as well. The primal problem for SMKL with any linear combination of GSTs may be written as shown in (5). Note that the primal problem can be reformulated as an unconstrained optimization problem with respect to $\mathbf{w}$ and b by substituting for ξ_i in the objective function and this would justify referring to it as $\mathrm{L}(\mathbf{w})$.

$$\mathrm{L}(\mathbf{w}) \begin{cases} \min g(\mathbf{w}) + C\sum_{i=1}^{N} \xi_i, \\ \text{such that } \xi_i \geq 0 \text{ and,} \\ y_i\left(\sum_{k=1}^{K} \langle \mathbf{w}^k, \Phi_k(\mathbf{x}_i)\rangle + b\right) \geq 1 - \xi_i, \ \forall i = 1, \ldots, N, \\ \mathbf{w}^k \in \mathbb{R}^{d_k}, \boldsymbol{\xi} \in \mathbb{R}^N, b \in \mathbb{R}. \end{cases} \tag{5}$$

Recall that the GSTs may be written as $g(\mathbf{w}) = G(\mathbf{r})$, where $\mathbf{r}_k = \sum_{j:j\in \mathbf{s}_k} \mathbf{w}_j^2$. For these terms,

$$\begin{aligned} C_j(\hat{\mathbf{w}}) &= \frac{1}{\mathbf{w}_j}\frac{\partial g}{\partial \mathbf{w}_j}\bigg|_{\hat{\mathbf{w}}} = \frac{1}{\mathbf{w}_j}\frac{\partial G}{\partial \mathbf{r}_k}\frac{\partial \mathbf{r}_k}{\partial \mathbf{w}_j}\bigg|_{\hat{\mathbf{w}}} = \frac{1}{\mathbf{w}_j}\frac{\partial G}{\partial \mathbf{r}_k} 2\mathbf{w}_j\bigg|_{\hat{\mathbf{w}}} \\ &= 2\frac{\partial G}{\partial \mathbf{r}_k}\bigg|_{\hat{\mathbf{r}}}, \end{aligned}$$

for $j \in \mathbf{s}_k$, i.e., the coefficient $C_j(\hat{\mathbf{w}})$ is the same for all features within a group as their value is dependent only on $\mathbf{r}_k$. Let the common value be B_k. Therefore, the upper bound for such a term may be written as $\frac{1}{2}\sum_{k=1}^{K} B_k \sum_{j\in s_k} \mathbf{w}_j^2 + B_0 = \frac{1}{2}\sum_{k=1}^{K} B_k \|\mathbf{w}^k\|_2^2 + B_0$. Considering an SVM-like formulation with this term for regularization and each $\mathbf{x}_i$ translated via K mappings $\Phi_k(\mathbf{x}) \mapsto \mathbb{R}^{d_k}$, $k = 1, 2, \ldots, \mathrm{K}$, from the input space to the combined feature space $(\Phi_1(\mathbf{x}), \ldots, \Phi_K(\mathbf{x}))$, the problem formulation for $Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$ is as shown in (6). It will be shown that this surrogate problem may be solved using the standard SVM implementations. It may be noted that, although the dual problem for (6) may be solved by standard SVMs to perform the update $\hat{\mathbf{w}}^{t+1} = \arg\min_{\mathbf{w}} Q(\mathbf{w} \mid \hat{\mathbf{w}}^t)$, as far as the original SMKL problem (5) is concerned, the optimization is done in the primal space.

$$Q(\mathbf{w}|\hat{\mathbf{w}}^t) \quad \begin{array}{l} \min \dfrac{1}{2}\displaystyle\sum_{k=1}^{K} B_k \|\mathbf{w}^k\|_2^2 + C \sum_{i=1}^{N} \xi_i, \\ \mathbf{w}^k \in \mathbb{R}^{d_k}, \xi \in \mathbb{R}^N, b \in \mathbb{R}, \\ \text{such that } \xi_i \geq 0 \text{ and,} \\ y_i\left(\displaystyle\sum_{k=1}^{K} \langle \mathbf{w}^k, \Phi_k(\mathbf{x}_i)\rangle + b\right) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, N. \end{array} \tag{6}$$

Let Ω be a diagonal matrix with $\Omega_{jj} = B_k$ for $j \in s_k$. The Lagrangian function for (6) is

$$\begin{aligned} J = &\frac{1}{2}\mathbf{w}^T \Omega \mathbf{w} + C \sum_{i=1}^{n} \xi_i \\ &- \sum_{i=1}^{n} \alpha_i \left\{ y_i \left(\sum_{k=1}^{K} (\mathbf{w}^k)^T \Phi_k(\mathbf{x}_i) + b \right) - 1 + \xi_i \right\} - \sum_{i=1}^{n} \gamma_i \xi_i. \end{aligned} \tag{7}$$

Setting the derivative of J w.r.t. the primal variables to zero,

$$\begin{aligned} &\frac{\partial J}{\partial \mathbf{w}} = \mathbf{0} \rightarrow \Omega \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i) \Rightarrow \mathbf{w} = \Omega^{-1} \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i), \\ &\frac{\partial J}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0, \\ &\frac{\partial J}{\partial \xi_i} = 0 \rightarrow \alpha_i + \gamma_i = \mathrm{C}, \quad i = 1, \ldots, n. \end{aligned} \tag{8}$$

Substituting $\mathbf{w}$ back into (7) and noting that $\Omega^{-1}\Phi(\mathbf{x}_i) = (\frac{1}{B_1}\Phi_1(\mathbf{x}), \ldots, \frac{1}{B_K}\Phi_K(\mathbf{x}))$, the dual problem is as shown in (9).

$$\begin{aligned} &\min \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j \sum_{k=1}^{K} \frac{1}{B_k} \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i, \\ &\text{s.t. } 0 \leq \alpha_i \leq \mathrm{C},\ i = 1, \ldots, n \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0. \end{aligned} \tag{9}$$

It can be seen that this is the same as solving the standard SVM with a kernel given by $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{K} \frac{1}{B_k} \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$. Some important observations are

$$\begin{aligned} \mathbf{w} &= \Omega^{-1} \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i) \\ \Rightarrow \mathbf{w}^k &= \Omega^{-1} \sum_{i=1}^{n} \alpha_i y_i \Phi_k(\mathbf{x}_i) \end{aligned} \tag{10}$$

$$\begin{aligned} \Rightarrow \|\mathbf{w}^k\|_2^2 &= \frac{1}{B_k^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j), \\ \beta_k &= \frac{1}{B_k}. \end{aligned} \tag{11}$$

From (10), $\|\mathbf{w}^k\|_2^2$ can be recovered from the solution of the standard SVM (9). Since $C_j(\mathbf{w})$ values may be updated using $\|\mathbf{w}^k\|_2^2$, the information required for applying Algorithm 1 to SMKL is now completely available. Since no specific information about the loss term was used in the derivation, it is easy to see that SMKL can be used to extend other single-kernel-based methods such as SVR [5] and LS-SVM [39] to learn sparse composite kernels. The only requirement for the algorithm to work is that the solution of the single kernel algorithm is of the form $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$. The representer theorem [5] guarantees the existence of solutions of this form under very mild assumptions which include all pointwise convex loss functions with quadratic regularizers.. The generalized SMKL algorithm is given in Algorithm 2.

Algorithm 2. *The generalized SMKL algorithm optimizes a convex combination of candidate kernels by iteratively calling an efficient single-kernel-based learning algorithm (SKLA).*

```
Set $\beta_k = \frac{1}{K}$ and $t = 0$.
while (true)
    Obtain $\alpha^t$ as the solution of the SKLA with kernel
        $\mathbf{k} = \sum_k \beta_k \mathbf{k}_k$
        $\left(\alpha^t \text{ is such that } \mathbf{w} = \sum_{i=1}^{N} \alpha_i^t \Phi(\mathbf{x}_i)\right)$
    Compute $\mathbf{r}_k = \|w^k\|_2^2 = \beta_k^2 \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathbf{k}_k(\mathbf{x}_i, \mathbf{x}_j)$ for
        $k = 1, 2 \ldots, K$
    Update the kernel weight estimates using,
        $\frac{1}{\beta_k} = B_k = 2 \left.\frac{\partial \mathrm{G}}{\partial \mathbf{r}_k}\right|_{\mathbf{r}}$ (from Table 1)
    if $\|\beta^{t+1} - \beta^t\|_1 \leq tol$
        break
    endif
    $t = t + 1$
end while
```

A comparison with some of the existing multiple kernel learning approaches is now presented. The biggest advantage of the proposed method, of course, is the fact that it can easily incorporate combinations of certain concave and convex regularization terms to produce sparser solutions for the kernel weighting vector β. In this sense, the proposed algorithm is a more generic one which can be used to solve any of the problem formulations reported in the literature to date as special cases. Even when using the same regularization term, $GST_{MKL}(\mathbf{w})$, the proposed algorithm has some advantages. Comparing it to the Sequential Minimal Optimization (SMO)-like approach in [25], the major advantage is that it is applicable to a wider class of loss functions without the need to implement specialized algorithms as it can exploit existing efficient single kernel algorithms such as the SMO algorithms developed for various loss functions. It will be demonstrated later that the computational cost of the algorithms for classification using the hinge loss are comparable. In terms of implementation, the proposed algorithm is more similar to the methods proposed in [26], which is based on solving a Semi-Infinite Linear Programming (SILP) problem and [28], which is based on iteratively solving a single-kernel-based method and performing a line search. Yet, the optimization strategies and approaches are totally different. Algorithm 2 is simpler to implement as it only requires an efficient single-kernel algorithm, while the method in [26] requires alternately solving a linear programming problem and the single kernel algorithm. Moreover, the use of SILPs could result in slow convergence of kernel weights due to iterative oscillations of parameter estimates [28]. Algorithm 2 is also better than the method in [28] as it uses the solution from the single kernel algorithm to not only implicitly determine the gradient of the actual objective function being minimized but also to automatically determine a suitable step size for the parameters, while simultaneously guaranteeing convergence of the estimates. The method in [28], on the other hand, requires an expensive line search, where each evaluation of the objective function requires solving a single kernel algorithm (although this is speeded by seeding the single kernel algorithm with the previous solution) and the convergence of the algorithm is dependent on the accuracy of this line search.

4 Experimental Results

In this section, results obtained by applying the SMKL algorithm to classification and regression problems are presented. Specifically, the performance of using a GST of the form $g(\mathbf{w}) = GST_{Log}(\mathbf{w}) + GST_{GL}(\mathbf{w})$ is compared against the standard penalty, $GST_{MKL}(\mathbf{w})$, currently used in MKL. This is because, though it is possible to consider any combination of GSTs from Table 1 as the regularization term, it was found that using the parameter free combination $g(\mathbf{w}) = GST_{Log}(\mathbf{w}) + GST_{GL}(\mathbf{w})$ consistently gave good results. As mentioned before, $GST_{Log}(\mathbf{w})$ alone is not expected to provide sufficient regularization when $\|\mathbf{w}^k\|_2$ becomes large and hence additional regularization in the form of $GST_{GL}(\mathbf{w})$ helps to prevent overfitting. It may be noted that the effect of $GST_{Log}(\mathbf{w})$ tends to be dominant when $\|\mathbf{w}^k\|_2$ is small, thus retaining the sparsity promoting properties discussed earlier, as verified by the results presented in this section. It may also be noted that, although a number of different algorithms have already been proposed for MKL, they all use equivalent problem formulations and, except for numerical differences in reaching the optimum, they are not supposed to have a significant difference in the sparsity of the resulting solution for β, and hence we do not implement each and every algorithm individually for comparison purposes and only use results from implementing the method in [26].

The use of $GST_{MKL}(\mathbf{w})$ in MKL guarantees that the obtained kernel weightings, β, satisfy $\sum_{k=1}^{K} \beta_k = 1$. This can be easily verified by observing that $\beta_k = 1/B_k = \|\mathbf{w}^k\|_2 / \sum_{j=1}^{K} \|\mathbf{w}^j\|_2$. The model obtained after MKL is of the form $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \sum_{k=1}^{K} \beta_k \mathbf{k}_k(\mathbf{x}, \mathbf{x}_i) + b)$. Generally, the norm of β is constrained as the parameters α and β are coupled, i.e., it is possible to decrease α by a constant factor while simultaneously increasing β without changing the model prediction and therefore the norm of β might grow in an unbounded fashion [27]. For the SMKL, $\frac{1}{\beta_k} = B_k = 2\frac{\partial G}{\partial \mathbf{r}_k}|_{\mathbf{r}^*}$. Since $G(\mathbf{r})$ is concave in the first quadrant, the slope $\frac{\partial G}{\partial \mathbf{r}_k}$ decreases monotonically with increasing $\mathbf{r}_k = \|\mathbf{w}^k\|_2^2$ and therefore the magnitude of β_k is related to $\|\mathbf{w}^k\|_2$. Since $\|\mathbf{w}^k\|_2$ is finite when the solution exists, this prevents β from growing in an unbounded fashion and hence there is no need to impose additional constraints on β for the SMKL.

First, a comparison of the computational complexity for $g(\mathbf{w}) = GST_{L0}^{Log}(\mathbf{w}) + GST_{L1}(\mathbf{w})$ using SMKL and $g(\mathbf{w}) = GST_{L1sq}(\mathbf{w})$ using the data reported for the SMO-like approach in [25] is presented. All experiments were run on a Pentium 4, 3 GHz machine with 1.25 GB RAM. For SMKL, the single kernel SVMs and SVRs were solved to an accuracy of 0.0001 using SVM^{light} [40], a popular and freely available implementation of the SVM. The termination criterion *tol* in Algorithm 2 was set to 1e-3. The traces of all of the candidate kernels were normalized to one. The same data sets (Ionosphere and Breast Cancer from the UCI Repository [41] and nested subsets of the adult data set from [42]) and procedure (using candidate basis kernels as Gaussian kernels on random subsets of features with varying widths) used in [25] are used for benchmarking the computational efficiency. Since this algorithm was run on a different machine, the absolute times are not comparable but the scaling with respect to the number of kernels and the number of data points may still be compared. Based on these experiments, the average computational complexity of SMKL was found to be proportional to $K^{1.0}$ and $n^{1.6}$. Comparing this with a complexity of $K^{1.1}$ and $n^{1.4}$ reported in [25], it can be seen that the computational efficiency of SMKL, in spite of its simplicity, is comparable to the SMO-like algorithm of [25]. Although the scaling of the proposed algorithm with respect to the number of data points is slightly higher, the scaling is still less than quadratic and acceptable for many practical situations.

The next two sections present the results of applying SMKL to classification and regression problems, respectively, using publicly available data sets. The results are compared in terms of prediction accuracy to the standard MKL algorithm [26], a baseline single-kernel-based method (SVM/SVR) which uses a kernel computed using all the features (called SKL for the sake of convenience) and also to the best reported results in literature for each data set. The single kernel algorithm was tried with linear, cubic polynomial, and radial basis kernels, and the kernel parameters were tuned using cross validation. In terms of the sparsity of the kernels selected, comparison is made only to the standard MKL algorithm since details about the selected features/wavelengths are either not available for the other algorithms or are meaningless (for the baseline single kernel methods). In both the cases, kernel weights which were less than 1e-6 times the maximum kernel weight were set to zero.

4.1 Classification

The first example makes use of a mass-spectroscopy data set for ovarian cancer detection [43], which is publicly

TABLE 2
Results for Ovarian Cancer Detection

Method	Test Accuracy (%)	Avg. Num. of Bands
SMKL	100.00±0.00	**2.0±0.0**
MKL [26]	98.67±0.27	8.5±1.5
SKL	100.00±0.00	10±0.0
LDA [44]	100.00	-NA-

Error is given in percentage prediction. Numbers in brackets denote upper bound of 95 percent confidence intervals.

TABLE 3
Results for Hyperspectral Image Classification

Method	Test Accuracy (%)	Avg. Num. of Kernels
SMKL	97.80	**2.82**
MKL [26]	97.50	7.12
SKL [45]	94.21	-NA-
Composite Kernel [45]	96.53	-NA-

available at http://home.ccr.cancer.gov/ncifdaproteomics. Each mass-spectrum curve represents the expression profiles of 15,154 peptides defined by their mass/charge (m/z) ratios in blood serum samples. The data set consists of 162 cancer samples and 91 control cases. In order to reduce the dimension of the data set, only the first 10,000 peptides were considered as the expression levels for the remaining peptides were close to zero. Further, five adjacent readings were averaged, resulting in 2,000 features per sample. These 2,000 features were divided into 40 groups with 50 features in each group and a linear kernel was defined for each group. Thirty random splits of the data into training (80 percent) and testing (20 percent) were used to assess the prediction accuracy of the data. The parameter C was tuned using the final leave-one-out error estimate of the SVM which can be obtained with negligible additional computational expense after training. The results are given in Table 2 and the prediction accuracy of SMKL is also compared to [44] where the authors use an optimization-procedure-based logical analysis of data (LDA). The two spectral bands selected by SMKL correspond to m/z ratios in the range (195.7, 266.4) and (348.1, 440.5), and these bands have been found to be sensitive to ovarian cancer detection through extensive data analysis in [44], whereas the spectral bands selected by MKL are much less selective and are harder to interpret.

The second example considered is a multiclass classification problem for hypersepctral image classification. The data set is freely available from http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/documentation.html and makes use of an image taken over northwest Indiana's Indian Pine test site in June 1992 by AVIRIS [45]. The image consists of 145×145 pixels, and reflectance values were recorded over 220 bands for each pixel. Out of this, 20 noisy bands covering the region of water absorption were removed to finally get 200 spectral bands. Based on available ground data, the image was divided into 16 classes and the goal is to classify pixels into one of these 16 classes based on the observed spectra. This example is used to demonstrate the capability of SMKL to fuse heterogeneous data to improve classification accuracy. As mentioned in [45] it is possible to improve classification accuracy by augmenting the spectral feature vector with "spatial features" that make use of information from neighboring pixels. In [45], the mean and standard deviation of the reflectance values for each band in a window around each pixel was used to define an additional "spatial kernel." The size of the window and the coefficients of the composite kernels were tuned manually. Based on the capacity of the SMKL to automatically select and tune the coefficients of a composite kernel, a large number of spatial features corresponding to the mean and standard deviation of the reflectance values for window sizes of 3×3, 5×5, 7×7, 9×9, and 11×11 pixels are considered here. Therefore, there are a total of 11 feature groups that include one spectral component and two spatial components for each of the five window sizes resulting in $11 \times 200 = 2{,}200$ features for each pixel. In order to allow the use of nonlinear classifiers four different kernels (one linear, one cubic polynomial, and two radial basis kernels with different widths) were considered for each feature group resulting in a total of 44 candidate kernels. A one versus one approach [5], which trains a separate classifier for each combination of two classes and then uses a voting scheme to pick the final class, was used to handle the multiclass problem. Therefore, a total of ${}^{16}C_2 = 120$ classifiers had to be trained. Twenty percent of the available data was used for training, and the remaining 80 percent was used for testing. The parameter C was tuned in all the cases using the leave-one-out estimate. The results are given in Table 3 along with a comparison of the best results reported in literature. The number of kernels for this example was computed as the average number of kernels used by the 120 classifiers.

It can be seen that both SMKL and MKL perform better than the best results reported in [45] with SMKL getting the overall best results with a significantly fewer average number of kernels used per classifier.

4.2 Regression

All three examples considered for regression involve the use of near-infrared (NIR) spectra for monitoring purposes. The NIR spectrum of a sample provides a global fingerprint of the physical and chemical state of the sample, which makes this information rich in content but not selective to any particular physical or chemical property, and hence the use of informative wavelengths is expected to increase prediction accuracy.

The first data set for protein content prediction is described in [4] and is publicly available from http://www.spectroscopynow.com. This data set was obtained by illuminating 150 ground wheat samples and then measuring light reflectance from the samples over a broad range of

TABLE 4
Results for Protein Content Prediction in Wheat

Method	Test NMSE	Avg. Num. of Bands
SMKL	0.024±0.007	**3.1±0.55**
MKL [26]	0.024±0.013	8.3±2.76
SKL	0.025±0.008	10±0.0
PLSR [4]	0.029±0.007	10.0±0.00

TABLE 5
Results for Nitrogen Content Prediction in Plants

Method	Test NMSE	Avg. Num. of Bands
SMKL	0.120±0.034	**3.1±0.31**
MKL [26]	0.120±0.032	9.2±0.78
SKL	0.150±0.024	10±0.0
PLSR [3]	0.151	10.0±0.00
B-splines+MI+RBFN [3]	0.121	-NA-

wavelengths over 1,000-2,500 nm at 2 nm intervals, resulting in 750 readings per sample. The measurements were transformed by taking the logarithm of the reciprocal of the reflectance. To reduce the amount of raw data, five adjacent wavelengths were averaged, resulting in 150 features. In order to develop a calibration model from this data, the protein content of the samples was measured by the well established, but time consuming, *Kjeldahl*-N method [4].

The second data set considered is selected from "The software shootout" [3] and is publicly available at http://kerouac.pharm.uky.edu/. It consists of scans and chemistry gathered from fescue grass (*Festuca elatior*). The grass was bred on soil medium with several nitrogen fertilization levels. The aim of the experiments was to try to find the optimum fertilization level to maximize production and minimize the consequences on the environment. In this context, the problem to be addressed was the determination of the nitrogen content of the plants using NIR spectrometry. Although the scans were performed on both wet and dry grass samples, only the wet samples are considered here, i.e., the scans were performed directly after harvesting. The data set contains 141 spectra discretized to 1,050 different wavelengths, from 400 to 2,498 nm with readings taken at an interval of 2 nm.

The third data set considered is selected from the diesel database, which was built by the Southwest Research Institute under a US Army contract and can be obtained from http://software.eigenvector.com/Data/SWRI/. It consists of scans of approximately 250 diesel fuel samples. The research was conducted to develop instrumentation for fuel quality assessment on battle fields. The aim was to predict several quantities from the NIR analysis: density, total aromatics, kinematic viscosity, net heat of combustion, freezing temperature, cetane number (CN), etc. The database contains only summer fuels, and outliers were removed. The problem of prediction of CN of the fuel, ranging from 40 to 60, is considered here. The corresponding data set contains 245 NIR transmission spectra. All spectra range from 750 to 1,550 nm, discretized into 401 wavelength values.

The same preprocessing technique is used for all three examples. All of the spectra are standardized to have zero mean and unit standard deviation. All of the spectra are augmented using features obtained by taking the second difference of the spectral feature vectors, which eliminates baseline drift from the spectra. The spectra are then divided into 10 bands and four different kernels (one linear, one cubic polynomial, and two Gaussian radial basis kernels) are defined for each band, giving a total of 40 candidate kernels for each problem. Thirty random splits of the data into training (80 percent) and testing (20 percent) sets were used to assess prediction accuracy. The parameter C and the tubewidth parameter for SKML and MKL were set using 10-fold cross validation using the training set only. The results are given in Tables 4, 5, and 6, along with 95 percent confidence intervals when available. The NMSE or the normalized mean squared error is defined as the ratio of the mean squared error to the variance of the output, which is estimated using the entire data set. PLSR denotes Partial Least Squares Regression which is the most widely used method for spectral chemometric applications [4] but makes use of the entire spectrum. "B-splines+MI+RBFN" denotes the method used in [3] which makes use of mutual-information-based forward selection on the coefficients of a B-spline representation of the spectrum to perform bandwidth selection and then uses a Radial Basis Function Network.

It is clear from the results that both SMKL and MKL consistently outperform PLSR on all tasks. The performance of the SKL is comparable to SMKL and MKL for the protein and cetane number prediction tasks, but is not as good for nitrogen content prediction, indicating the benefit of bandwidth selection for this example. Moreover, the results from the single kernel model cannot be interpreted any further whereas the sensitivity of different bandwidths to the task at hand can be inferred from the solutions of SMKL and MKL. The top three spectral bands selected by SMKL for protein

TABLE 6
Results for Cetane Number Prediction of Diesel

Method	Test NMSE	Avg. Num. of Bands
SMKL	0.265±0.030	**3.0±0.00**
MKL [26]	0.270±0.034	7.0±0.00
SKL	0.274±0.030	10±0.0
PLSR [3]	0.367	10.0±0.00
B-splines+MI+RBFN [3]	0.375	-NA-

content prediction are 1,600-1,750 nm, 2,200-2,350 nm, and 2,350-2,500 nm. Of these, the bands between 1,600-1,750 nm and 2,200-2,350 nm are known to be sensitive to protein content [4]. For the task of nitrogen content prediction both SMKL and MKL perform as well as the "B-splines+ MI+RBFN" method, which involves a lot more manual tuning for the selection of the order of the B-spline, selection of the number of features during forward selection, the number of RBFN nodes, etc., whereas, for the multiple kernel approaches, the use of different candidate kernels for each band automatically handles a lot of parameter tuning tasks. In principle, a larger number of candidate kernels could be used to further fine tune parameters such as the width of the radial basis kernel. Finally, both SMKL and MKL easily outperform both PLSR and "B-splines+MI+RBFN" for the task of cetane number prediction.

Based on the examples presented in this section for both classification and regression, as far as accuracy is concerned the performance of SMKL and MKL are comparable to each other and definitely produce results which are either comparable to or superior to the best reported results in literature for the various data sets while having fewer parameters to tune. It is also clear that the SMKL algorithm produces solutions which are significantly sparser in terms of the number of kernels used as compared to MKL. The advantage of this in terms of interpretability was demonstrated using two examples for ovarian cancer detection and protein content prediction where prior knowledge about the sensitive parts of the spectrum was used to verify that the SMKL does indeed select these bands. Also the SMKL algorithm is much simpler to implement as compared to the currently existing methods [25], [26], [28]. Hence, we would recommend the use of SMKL over MKL for almost all applications. If the computational burden is not an issue, we would recommend the use of SMKL over single kernel algorithms as well as there is the chance that prediction accuracy may be increased by the use of a combination of kernels. If this is not the case, it is expected that the SMKL would automatically suggest the use of a single kernel.

5 Conclusion

Composite kernel learning is a problem with many significant applications including grouping of features in nonlinear models, a concept that has significant applications in the area of multisource data fusion. This is significantly different from the problem of feature selection on which most of the current machine learning research has focused and is especially tricky when only a few data points are available. It is shown in this work that using the proposed GST as a regularization term works well in this scenario. A significant advantage of the proposed method, as compared to methods based on forward/backward selection, is that, when the hyperparameter (C) of the algorithm is properly tuned, the proposed method automatically selects the optimal number of sources from a large candidate list, of which many may be irrelevant or redundant. The sparser solutions provided by the SMKL as compared to MKL has many benefits in terms of interpretability in sparse signal approximation, cost reduction for sensor fusion problems, and reduced implementation time for spectral chemometric applications. The effectiveness of the proposed SMKL algorithm has been demonstrated on a number of real world data sets. The parameter grouping strategy proposed in this work can also be extended to other areas such as feature/ kernel selection for multiclass problems. Finally, it is possible to parallelize the implementation of SMKL to exploit the increasing availability of cheap parallel processors, and thus handle a large number of kernels and much larger data sets.

References

[1] D.L. Hall, *Mathematical Techniques in Multisensor Data Fusion.* Artech House, Inc., 1992.

[2] M. Vannucci, N. Sha, and P.J. Brown, "NIR and Mass Spectra Classification: Bayesian Methods for Wavelet-Based Feature Selection," *Chemometrics and Intelligent Laboratory Systems,* vol. 77, nos. 1/2, pp. 139-148, May 2005.

[3] F. Rossi et al., "Fast Selection of Spectral Variables with B-Spline Compression," *Chemometrics and Intelligent Laboratory Systems, Selected Papers Presented at the Chemometrics Congress,* vol. 86, no. 2, pp. 208-218, Apr. 2007.

[4] H. Martens and M. Martens, *Multivariate Analysis of Quality: An Introduction.* John Wiley & Sons, Ltd., 2001.

[5] B. Scholkopf and A.J. Smola, *Learning with Kernels.* MIT Press, 2002.

[6] I. Guyon et al., *Feature Extraction, Foundations and Applications.* Physica-Verlag, Springer, 2006.

[7] I. Guyon et al., *Feature Extraction, Foundations and Applications,* Physica-Verlag, Springer, 2006.

[8] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Machine Learning Research,* vol. 3, pp. 1157-1182, Mar. 2003.

[9] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Royal Statistical Soc. Series B,* vol. 58, no. 1, pp. 267-288, 1996.

[10] P.S. Bradley and O.L. Mangasarian, "Feature Selection via Concave Minimization and Support Vector Machines," *Proc. 15th Int'l Conf. Machine Learning,* pp. 82-90, 1998.

[11] M.A.T. Figueiredo, "Adaptive Sparseness for Supervised Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 9, pp. 1150-1159, Sept. 2003.

[12] J. Weston et al., "Use of the Zero-Norm with Linear Models and Kernel Methods," *J. Machine Learning Research,* vol. 3, pp. 1439-1461, Mar. 2003.

[13] M.E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *J. Machine Learning Research,* vol. 1, no. 3, pp. 211-244, 2001.

[14] T.N. Lal et al., "Support Vector Channel Selection in BCI," *IEEE Trans. Biomedical Eng.,* vol. 51, no. 6, pp. 1003-1010, June 2004.

[15] Y. Kim, J. Kim, and Y. Kim, "Blockwise Sparse Regression," *Statistica Sinica,* vol. 16, pp. 375-390, 2006.

[16] T. Similä and J. Tikka, "Input Selection and Shrinkage in Multiresponse Linear Regression," *Computational Statistics and Data Analysis,* vol. 52, pp. 406-422, 2007.

[17] L. Wang, G. Chen, and H. Li, "Group SCAD Regression Analysis for Microarray Time Course Gene Expression Data," *Bioinformatics,* vol. 23, no. 12, pp. 1486-1494, 2007.

[18] M. Yuan and Y.B. Lin, "Model Selection and Estimation in Regression with Grouped Variables," *J. Royal Statistical Soc.,* vol. 68, pp. 49-67, 2006.

[19] P. Zhao, G. Rocha, and B. Yu, "Grouped and Hierarchical Model Selection through Composite Absolute Penalties," technical report, Univ. of California, 2006.

[20] J. Stoeckel and G. Fung, "SVM Feature Selection for Classification of SPECT Images of Alzheimer's Disease Using Spatial Information," *Proc. Fifth IEEE Int'l Conf. Data Mining,* pp. 410-417, 2005.

[21] S.F. Cotter et al., "Sparse Solutions to Linear Inverse Problems with Multiple Measurement Vectors," *IEEE Trans. Signal Processing,* vol. 53, no. 7, pp. 2477-2488, July 2005.

[22] S. Ma, X. Song, and J. Huang, "Supervised Group Lasso with Applications to Microarray Data Analysis," *Bioinformatics,* vol. 8, no. 60, 2007.

[23] L. Meier, S. van de Geer, and P. Buhlmann, "The Group Lasso for Logistic Regression," technical report, Eidgenössische Technische Hochschule, 2006.

[24] G.R.G. Lanckriet et al., "Learning the Kernel Matrix with Semidefinite Programming," *J. Machine Learning Research,* vol. 5, pp. 27-72, 2004.

[25] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan, "Multiple Kernel Learning, Conic Duality, and the SMO Algorithm," *Proc. 21st Int'l Conf. Machine Learning,* pp. 41-48, 2004.

[26] S. Sonnenburg et al., "Large Scale Multiple Kernel Learning," *J. Machine Learning Research,* vol. 7, pp. 1531-1565, 2006.

[27] M. Girolami and S. Rogers, "Hierarchic Bayesian Models for Kernel Learning," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 241-248, 2005.

[28] A. Rakotomamonjy et al., "More Efficiency in Multiple Kernel Learning," *Proc. 24th Int'l Conf. Machine Learning,* pp. 775-782, 2007.

[29] K. Lange, D. Hunter, and I. Yang, "Optimization Transfer Using Surrogate Objective Functions," *J. Computational and Graphical Statistics,* vol. 9, pp. 1-59, 2000.

[30] B. Krishnapuram et al., "Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 957-968, June 2005.

[31] M. Girolami, "A Variational Method for Learning Sparse and Overcomplete Representations," *Neural Computation,* vol. 13, no. 11, pp. 2517-2532, 2001.

[32] Z. Zhang, J.T. Kwok, and D.-Y. Yeung, "Surrogate Maximization/Minimization Algorithms for AdaBoost and the Logistic Regression Model," *Proc. 21st Int'l Conf. Machine Learning,* pp. 927-934, 2004.

[33] E.J. Candès, M. Wakin, and S. Boyd, "Enhancing Sparsity by Reweighted l1 Minimization," *J. Fourier Analysis and Applications,* vol. 14, pp. 877-905, 2007.

[34] D.P. Wipf and B.D. Rao, "An Empirical Bayesian Strategy for Solving the Simultaneous Sparse Approximation Problem," *IEEE Trans. Signal Processing,* vol. 55, no. 7, pp. 3704-3716, July 2007.

[35] J. Neumann, C. Schnorr, and G. Steidl, "Combined SVM-Based Feature Selection and Classification," *Machine Learning,* vol. 61, nos. 1-3, pp. 129-150, 2005.

[36] N. Subrahmanya and Y.C. Shin, "Automated Sensor Selection and Fusion for Monitoring and Diagnostics of Plunge Grinding," *J. Manufacturing Science and Eng.,* Trans. ASME, vol. 130, no. 3, 031014, 2008.

[37] R. Neal and G. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models,* M.I. Jordan, ed., Kluwer, 1998.

[38] B. Krishnapuram et al., "A Bayesian Approach to Joint Feature Selection and Classifier Design," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1105-1111, Sept. 2004.

[39] J.A.K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters,* vol. 9, no. 3, pp. 293-300, June 1999.

[40] T. Joachims, "Making Large-Scale SVM Learning Practical," *Advances in Kernel Methods—Support Vector Learning:* B. Schölkopf, C. Burges, and A. Smola, eds., MIT-Press, 1999.

[41] D.J. Newman et al., *UCI Repository of Machine Learning Databases,* Dept. of Information and Computer Science, Univ. of California, Irvine, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[42] J. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," *Advances in Kernel Methods—Support Vector Learning,* MIT Press, 1998.

[43] E.F. Petricoin et al., "Use of Proteomic Patterns in Serum to Identify Ovarian Cancer," *Lancet,* vol. 359, pp. 572-577, 2002.

[44] G. Alexe et al., "Ovarian Cancer Detection by Logical Analysis of Proteomic Data," *Proteomics,* vol. 4, no. 3, pp. 766-783, 2004.

[45] G. Camps-Valls et al., "Composite Kernels for Hyperspectral Image Classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 3, no. 1, pp. 93-97, Jan. 2006.

Niranjan Subrahmanya received the PhD degree from Purdue University, West Lafayette, Indiana, in May 2009. He is currently working with the Complex Systems Science group at Exxon Mobil Research and Engineering. His research interests include intelligent systems, dynamics and control, data-based systems modeling, monitoring and diagnostics, and machine learning.

Yung C. Shin received the BS degree from Seoul National University in 1976, the MS degree from the Korea Advanced Institute of Science and Technology in 1978, and the PhD degree from the University of Wisconsin-Madison in 1984, and currently is a professor of mechanical engineering at Purdue University, West Lafayette, Indiana. He worked as a senior project engineer at the GM Technical Center from 1984 to 1988 and as a faculty member at the Pennsylvania State University from 1988 to 1990, prior to joining Purdue University in 1990. His research areas include intelligent and adaptive control, process monitoring and diagnostics, laser processing of materials, high speed machining, process modeling, and simulation. He has published more than 200 papers in archived journals and refereed conference proceedings, and has authored chapters in several engineering handbooks and coedited two books. He also is a coauthor of *Intelligent Systems: Modeling, Optimization and Control* (CRC Press, 2008).