Group Lasso Regularized Multiple Kernel Learning for Heterogeneous Feature Selection

Yi-Ren Yeh, Yung-Yu Chung, Ting-Chu Lin, and Yu-Chiang Frank Wang

***Abstract*— We propose a novel multiple kernel learning (MKL) algorithm with a group lasso regularizer, called group lasso regularized MKL (GL-MKL), for heterogeneous feature selection. We extend the existing MKL algorithm and impose a mixed ℓ_1 and ℓ_2 norm constraint (known as group lasso) as the regularizer. Our GL-MKL determines the optimal base kernels, including the associated weights and kernel parameters, and results in a compact set of features for comparable or improved recognition performance. The use of our GL-MKL avoids the problem of choosing the proper technique to normalize the feature attributes collected from heterogeneous domains (and thus with different properties and distribution ranges). Our approach does not need to exhaustively search for the entire feature space when performing feature selection like prior sequential-based feature selection methods did, and we do not require any prior knowledge on the optimal size of the feature subset either. Comparisons with existing MKL or sequential-based feature selection methods on a variety of datasets confirm the effectiveness of our method in selecting a compact feature subset for comparable or improved classification performance.**

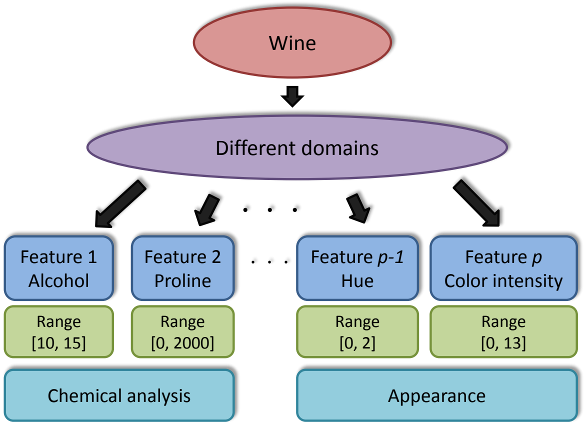

Fig. 1. An example of the UCI dataset (wine) in which the features are collected from heterogeneous domains, and thus each feature attribute has a unique property and distribution range (e.g., Feature 2 corresponds to Proline, ranging between 0 and 2000).

I. Introduction

Different from feature extraction, which requires the determination of feature domains/sources to use for pattern recognition problems, feature selection aims to select a subset of relevant features for reduced computation complexity with improved or comparable recognition performance. More specifically, the task of feature selection is to identify (or to remove) redundant features from the original feature set according to some criterion functions. As a result, a comparable or improved recognition performance will be achieved with a compact feature subset selected.

Several feature selection techniques have been proposed and are widely used due to its simplicity in implementation. For example, sequential forward or backward selection (SFS [1] and SBS [2]) methods provide an intuitive way to select a feature subset by iteratively adding or removing features from the original feature set and improve the recognition performance. However, these sequential selection based approaches might encounter *nesting effects*, i.e. the subset of the N best features must contain the subset of the N-1 best ones, and so on. The sequential forward floating selection (SFFS) [3] algorithm was proposed to alleviate this effect by a dynamic backtracking after each iteration. Recently, an improved forward floating selection (IFFS) [4] was proposed to improve the SFFS algorithm by replacing weak features when backtracking can not locate a better subset than that of previous iterations. Although impressive results were reported in SFFS and IFFS, their high computational cost prohibits practical uses especially for high-dimensional data. Besides the nesting effect, another limitation of this type of methods is that the users are required to specify the preferable size of the feature subset for the termination of the entire selection process. However, the optimal number of feature subset is typically not known in advance.

It is worth noting that, in real-world applications, the features of interest might come from different domains in many pattern recognition problems. For example, to classify different types of wines, one might consider appearance or chemical components as the features of interest, as shown in Figure 1. The features extracted from different domains of interest are typically referred to as *heterogeneous features*. This type of features has a unique property for each feature attribute (e.g., physical meaning, distribution, sensor type, etc.), and thus the direct use of heterogeneous features for designing pattern recognition algorithms often results in the dependency on its dominant feature subset. However, the dominant features do not necessarily imply improved recognition performance, and thus it is very challenging to deal with heterogeneous features for practical problems. While

Yi-Ren Yeh and Ting-Chu Lin are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, 11529 (email: {yryeh, tingchulin}@citi.sinica.edu.tw).

Yung-Yu Chung was with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, and is currently with Dept. Electrical and Computer Engineering, Iowa State University, Ames, IA 50014, USA (email: ychung@iastate.edu).

Yu-Chiang Frank Wang is with the Research Center for Information Technology Innovation and Institute of Information Science, Academia Sinica, Taipei, Taiwan, 11529 (email: ycwang@citi.sinica.edu.tw).

techniques such as linear or zero-mean normalization can be applied on the heterogenous features before designing the recognition algorithms, one will need to carefully select the proper normalization technique to preprocess the data. More importantly, restricting each heterogenous feature to have the same range of distribution does not guarantee improved recognition performance either.

During the past few years, kernel methods such as support vector machines (SVM) [5] have been shown to be very effective for data representation, dimension reduction, and classification. A more flexible learning model using multiple kernels instead of one, which is known as multiple kernel learning (MKL), has recently been proposed [6]. MKL has shown to improve the performance of many learning tasks such as [7], [8]. Prior work such as [9], [10], [11] has proposed to use MKL for feature selection related problems. For example, Gehler *et al.* [9] applied MKL to learn the optimal weights when combining different types of features for object recognition; however, they did not focus on selecting the optimal feature subset for each type of features, and thus their work is considered as feature fusion instead of selection. Dileep *et al.* [10] proposed to identify the feature subset according to the weights of the base kernels learned for each dimension in the feature space, but they only observed limited performance improvements. Although an improved feature selection method recently proposed by Xu *et al.* [11] obtained promising results using MKL, their method needs to solve a complex combinatorial optimization problem, which is not preferable when the numbers of data samples or the corresponding feature dimensions are large. Their method also requires the user to specify the preferable size of the feature subset. Nonetheless, these prior MKL based approaches treat all features equally important, and none of them addresses the problem of heterogeneous feature selection.

In this paper, we propose a novel MKL-based method for heterogeneous feature selection, which does not require the selection of proper feature normalization techniques nor the prior knowledge on the optimal size of the feature subset. We extend the standard MKL formulation and impose a mixed ℓ_1 and ℓ_2 norm constraint as the group lasso regularizer, which will determine the optimal weights for each base kernel and thus achieve the goal of feature selection. In our framework, each heterogeneous feature is associated with multiple base kernels and is thus considered as a group. The imposed group lasso regularizer tends to maintain sparsity between different groups (i.e. features), while the associated weights of the selected kernels for each group need not be sparse. This allows our MKL algorithm to select more than one base kernel for each type of heterogeneous features, while a compact set of groups (features) will be enforced due to the added sparsity at the group (feature) level. Since we associate each heterogeneous feature with multiple base kernels with different kernel parameter (e.g. width of the Gaussian kernel), our MKL has the capability to deal with heterogeneous data without the need to perform any specific normalization procedure. Using our approach, a compact feature subset and their optimal kernels (including the associated weights and kernel parameters) can be learned automatically and simultaneously.

We note that the combination of ℓ_1 and ℓ_2 regularization terms is used in the elastic net [12]. Different from our method utilizing a mixed ℓ_1 and ℓ_2 norm regularizer, the elastic net model balances the trade-off between ℓ_1 and ℓ_2 regularization for feature selection in the original input space. Although the elastic net is able to associate highly correlated feature attributes during the selection process, it treats all feature attributes equally important and might not be preferable for heterogeneous feature selection. Besides, it cannot be easily extended to feature selection in nonlinear feature spaces. It is worth noting that our proposed method not only addresses the above issues, our GL-MKL framework can be simplified using linear kernels, and thus we can perform feature selection in either linear or nonlinear spaces.

The remaining of this paper is organized as follows. Section II briefly reviews multiple kernel learning and its use for feature selection. Our group lasso regularized MKL framework for feature selection is introduced in Section III. Experimental results in terms of both classification accuracy and the number of selected features on several datasets are presented in Section IV. Finally, Section V concludes this paper.

II. Multiple Kernel Learning

A. Review of MKL

The support vector machine (SVM) [13] has been known to be an effective binary classifier due to its generalization ability. It learns an optimal separating hyperplane to distinguish data between two different classes without any assumption on data distribution. For the standard SVM, the norm vector $\mathbf{w}$ of the hyperplane needs to address the following problem,

$$\begin{aligned} \min_{(\mathbf{w},b,\boldsymbol{\xi})\in\mathbb{R}^{p+1+n}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) + \xi_i \geq 1\,, \\ & \xi_i \geq 0, \quad \text{for } i = 1,2,\ldots,n\,, \end{aligned} \tag{1}$$

where C is the trade-off between the SVM's generalization and the training error ξ_i, which is the Hinge loss of $\mathbf{x}_i$. The corresponding dual problem of (1) is expressed as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i\alpha_i = 0, \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}C, \end{aligned} \tag{2}$$

where α_i are the Lagrange coefficients. The kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ is applied in (2) to calculate the inner product between different $\mathbf{x}_i$ in the transformed space. However, a single kernel function might not be sufficient to model the separating hyperplane, especially when the data distribution is complex. As a result, multiple kernels are recently applied for this purpose, and this is referred to as multiple kernel learning (MKL) [6]. More precisely, one can replace the single kernel in (2) by a linear combination of base kernels, while each kernel describes a different property of the data of interest (i.e., different feature spaces or distributions). Thus, this will provide improved generalization ability for the learning model.

Similar to SVM, one can approach MKL by formulating and solving its primal form. This process can be considered as describing the data in multiple feature spaces using different norm vectors $\mathbf{w}_m$. According to [14], the primal form of MKL is thus formulated as the following optimization problem:

$$\begin{aligned} \min_{(\boldsymbol{\beta},\mathbf{w},b,\boldsymbol{\xi})\in\mathbb{R}^{p+p+1+n}} \quad & \frac{1}{2}\sum_{\ell=1}^{p}\frac{1}{\beta_\ell}\|\mathbf{w}_\ell\|_2^2 + C\sum_{i=1}^{n}\xi_i \qquad (3)\\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) + \xi_i \geq 1\,,\\ & \xi_i \geq 0, \quad \text{for } i = 1,2,\ldots,n\,,\\ & \|\boldsymbol{\beta}\|_1 = 1, \quad \boldsymbol{\beta} > \mathbf{0}, \end{aligned}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_p]^\top$. Similar to SVM, C in (3) is the trade-off between the generalization of MKL and its training errors ξ_i. From the above formulation, we see that the primal form of MKL restricts the weight of the norm vector w_ℓ with $\beta_\ell \geq 0$, and it also imposes the constraint of $\|\boldsymbol{\beta}\|_1 = 1$, which tends to produce a sparse solution for $\boldsymbol{\beta}$. These weight coefficients β_ℓ determine the significance of norm vectors $\mathbf{w}_\ell$ for MKL, which will be associated with the weights for each kernel in the MKL dual form, as discussed next.

Using the kernel functions, the MKL can also be transformed and be solved in higher-order transformed sapce. With the existing constraint on β_ℓ, the minimization problem (3) can thus be transformed into the following min-max problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}}\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \sum_{\ell=1}^{p}\beta_\ell k_\ell(\mathbf{x}_i, \mathbf{x}_j)\\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i\alpha_i = 0 \qquad (4)\\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}C, \quad \|\boldsymbol{\beta}\|_1 = 1, \quad \boldsymbol{\beta} > \mathbf{0}. \end{aligned}$$

Comparing (4) to (2), multiple base kernel functions are applied in (4) while only a single kernel is used in (2). In (4), since the constraint $\|\boldsymbol{\beta}\|_1 = 1$ tends to result in a sparse solution of β_ℓ, this learning process can be viewed as the removal of redundant kernels among the base kernels. Simply speaking, the MKL formulation in (4) aims to determine an optimal and compact linear combination of base kernels for improved recognition performance, and this is achieved by learning the best weights β_ℓ for the base kernels and the predictors α_i for the associated data (for classification). Once the training process is complete, the coefficients α_i and β_ℓ are determined. For a test input $\mathbf{x}$, the decision function of MKL can be computed as:

$$F_{MKL}(\mathbf{x}) = \text{sign}\Big(\sum_{i=1}^{n}\sum_{\ell=1}^{p}\beta_\ell(k_\ell(\mathbf{x}_i, \mathbf{x})\cdot\alpha_i + b)\Big). \qquad (5)$$

B. MKL for Feature Selection

MKL has recently been applied for feature combination or selection [9], [10], [11]. Existing methods typically approach this type of problem as solving a task of learning the optimal weights for each feature representation or feature attribute. More specifically, for feature selection in a p-dimensional space, MKL aims to learn the weights β_m for each of the p feature dimensions according to its relevance to the task of classification (see (4)). More specifically, MKL uses each feature to generate its corresponding kernel and determine the weight coefficient of each kernels as shown in Fig. 2. Although the weighted sum of these kernels calculated from individual features is expected to improve the classification performance, results reported in previous works such as [10] only indicate negligible improvements on several data sets. However, the above methods treat all features equally important during the selection process, and thus the resulting feature subset will be dominated by those with larger distribution ranges. In other words, they did not address the problem of feature selection from heterogeneous data. In the next section, we will detail our proposed method for heterogeneous feature selection.

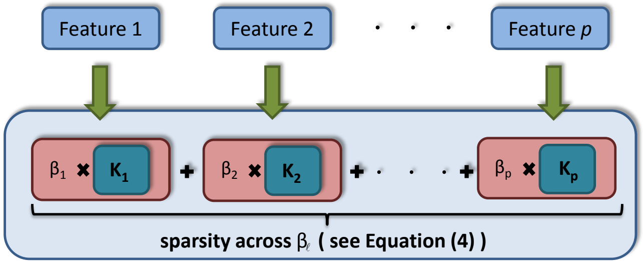
Feature 1
Feature 2
· · ·
Feature p
β1 ✖ K1 ✚ β2 ✖ K2 ✚ · · · ✚ βp ✖ Kp
sparsity across βℓ (see Equation (4))

Fig. 2. Illustration of MKL for feature selection. Each feature constructs a base kernel, and the MKL determines weight coefficients β_ℓ for each base kernel with a sparsity constraint.

III. Group Lasso Regularized MKL

A. Algorithm of GL-MKL

In this paper, we focus on the problem of heterogeneous feature selection, i.e., each or some feature attributes from the original p-dimensional space are collected from heterogeneous sources/domains and thus have different properties and distribution ranges. As discussed earlier, prior feature selection methods using MKL impose the ℓ_1 regularizer on the coefficient vector $\boldsymbol{\beta}$, and this constraint tends to produce a sparse solution [6], [14] (i.e., only few weights β_ℓ are with non-zero values). While this provides an effective way to select discriminating features (depending on the associated β_ℓ values), this type of approach still considers different

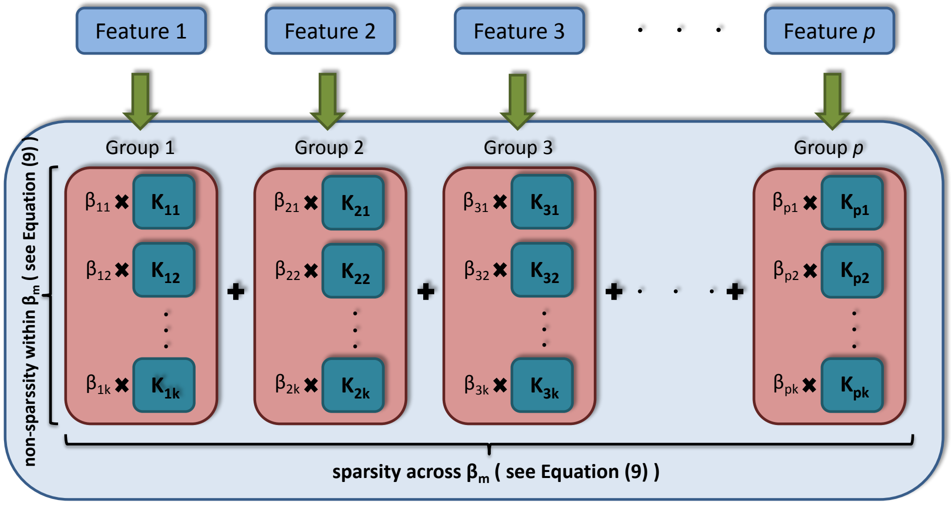

Fig. 3. Illustration of our GL-MKL for heterogeneous feature selection. Different from Fig. 1, multiple kernels constructed for each heterogeneous feature form a group, and we enforce the group-lasso constraint on the weights of each base kernel. Our group-lasso constraint tends to select a compact set of groups for the purpose of feature selection. Since different types of feature attributes should be associated with their preferable kernels, our GL-MKL allows the existence of non-sparsity for the kernel weights within each group.

features equally important in the selection process, and thus it is not clear how to extend this MKL-based feature selection approach to deal with heterogeneous data.

To address this problem, we propose a novel MKL with a group lasso regularizer, called *group lasso regularized MKL* (GL-MKL), which constrains the coefficient $\boldsymbol{\beta}$ with a mixed ℓ_1 and ℓ_2 norm. Suppose that we have each of the p features paired with k different kernel choices (e.g. different σ choices if using Gaussian kernels). There is a total of $p \times k$ base kernels in our group lasso regularized MKL. That is, we have $\boldsymbol{\beta} = [\boldsymbol{\beta}_1; \boldsymbol{\beta}_2; \cdots; \boldsymbol{\beta}_p] = [\beta_{11}, \beta_{12}, \ldots, \beta_{pk}]^\top \in \mathbb{R}^{(p \times k) \times 1}$, which are associated with base kernels as shown in Fig. 3. Our mixed ℓ_1 and ℓ_2 constraint imposed on $\boldsymbol{\beta}$ will maintain sparsity between different groups (i.e. different features), while the associated β_{mj} values in each group need not be sparse (see Fig. 3). More precisely, we enforce the sparsity constraint at the feature level (for feature selection), and we allow our MKL to select more than one kernels for each feature to improve overall performance (to handle heterogeneous features).

With these $p \times k$ kernels and the corresponding coefficient $\boldsymbol{\beta} \in \mathbb{R}^{(p \times k) \times 1}$, the primal form of our GL-MKL is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta}, \mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \sum_{m=1}^{p} \sum_{j=1}^{k} \frac{1}{\beta_{mj}} \|\mathbf{w}_{mj}\|_2^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) + \xi_i \geq 1\,, \qquad (6) \\ & \xi_i \geq 0, \quad \text{for } i = 1, 2, \ldots, n\,, \\ & \sum_{m=1}^{p} \|\boldsymbol{\beta}_m\|_2 \leq 1, \quad \boldsymbol{\beta}_m \in \mathbb{R}^k > \mathbf{0}, \ \forall\, m, \end{aligned}$$

where $\boldsymbol{\beta}_m = [\beta_{m1}, \beta_{m2}, \ldots, \beta_{mk}]$. We also apply the same setting in [7] and relax the equality constraint $\sum_{m=1}^{p} \|\boldsymbol{\beta}_m\|_2 = 1$ to $\sum_{m=1}^{p} \|\boldsymbol{\beta}_m\|_2 \leq 1$ due to the convexity of the optimization problem. It is also worth noting that the constraint of the mixed norm in (6) provides us the flexibility to control the sparsity within $\boldsymbol{\beta}$. More specifically, assigning ($k = 1$, $p = 0$) or ($k = 0$, $p = 1$) will convert our algorithm back to ℓ_1 or ℓ_2 regularized MKL problem, which can be considered as two special cases of our proposed MKL. In practice, one can choose different numbers of kernels for each feature using our MKL, while we fix this number k in this paper.

We see that, if $\boldsymbol{\beta}$ is fixed in (6), our GL-MKL formulation becomes a Lagrangian function of variables $\mathbf{w}$, b, and $\boldsymbol{\xi}$:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}) \quad = \quad & \frac{1}{2} \sum_{m=1}^{p} \sum_{j=1}^{k} \frac{1}{\beta_{mj}} \|\mathbf{w}_{mj}\|_2^2 + C \sum_{i=1}^{n} \xi_i \qquad (7) \\ & + \sum_{i=1}^{n} \alpha \left(1 - \xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\right) - \sum_{i=1}^{n} \nu_i \xi_i, \end{aligned}$$

where α_i and ν_i are the Lagrangian multipliers. Setting the derivatives of this function with respect to the corresponding variables to zero, we have the following conditions:

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi})}{\partial \mathbf{w}} = 0 \quad & \Rightarrow \quad \mathbf{w}_{mj} = \beta_{mj} \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \quad \forall\ m, j \\ \frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi})}{\partial b} = 0 \quad & \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \qquad (8) \\ \frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = 0 \quad & \Rightarrow \quad C - \alpha_i - \nu_i = 0, \quad \forall\ i. \end{aligned}$$

Substitute the KKT conditions to (7), we then transform (6) into the following min-max optimization problem

$$\begin{aligned}
\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad & S(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \\
& \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \sum_{m,j=1}^{p,k} \beta_{mj} k_{mj}(\mathbf{x}_i, \mathbf{x}_j) \\
\text{s.t.} \quad & \sum_{i=1}^{n} y_i \alpha_i = 0 \qquad (9) \\
& \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}C, \quad \sum_{m=1}^{p} \|\boldsymbol{\beta}_m\|_2 \leq 1, \quad \boldsymbol{\beta} > \mathbf{0}.
\end{aligned}$$

The above min-max problem be solved by gradient based methods (e.g., [6], [14]). Alternatively, we can formulate (9) as a semi-infinite programming (SIP) problem [15] and search for the best $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ iteratively. To be more specific, we fix $\boldsymbol{\beta}$ and solve the maximization problem of (9) with respect to $\boldsymbol{\alpha}$; we note that this procedure can be addressed using any regular SVM solver such as `libSVM` [16], which solves (9) with fixed $\boldsymbol{\beta}$. Once the variables $\boldsymbol{\alpha}$ are determined in an iteration, we fix $\boldsymbol{\alpha}$ and solve the minimization problem of (9) with respect to $\boldsymbol{\beta}$. Suppose that $\boldsymbol{\alpha}^*$ is the optimal solution in (9), we have the objective value $S(\boldsymbol{\alpha}^*, \boldsymbol{\beta}) = \theta$ and $\theta \geq S(\boldsymbol{\alpha}, \boldsymbol{\beta})$ for all $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Thus, by fixing $\boldsymbol{\alpha}$, we convert (9) into the following SIP problem which minimizing θ to its lower bound:

$$\begin{aligned}
\min_{\theta, \boldsymbol{\beta}} \quad & \theta \qquad (10) \\
\text{s.t.} \quad & \theta \geq S(\boldsymbol{\alpha}, \boldsymbol{\beta}), \\
& \sum_{m=1}^{p} \|\boldsymbol{\beta}_m\|_2 \leq 1, \ \boldsymbol{\beta}_m \in \mathbb{R}^k > \mathbf{0} \ \forall m, \\
& \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{1}C, \quad \sum_{i=1}^{m} y_i \alpha_i = 0, \ \forall \boldsymbol{\alpha} \in \mathbb{R}^n.
\end{aligned}$$

In our implementation, we use the function `fmincon` in MATLAB to solve (10). By iteratively solving the above two types of optimization problems (with respect to $\boldsymbol{\beta}$ or $\boldsymbol{\alpha}$), the optimal solution of (9) is thus determined. The pseudo code of our GL-MKL is described in Algorithm 1, and the decision function of GLMKL is calculated as:

$$F_{GLMKL}(\mathbf{x}) = \text{sign}\Big(\sum_{i=1}^{n} \sum_{m,j=1}^{p,k} \beta_{mj}(k_{mj}(\mathbf{x}_i, \mathbf{x})\alpha_i + b)\Big). \qquad (11)$$

B. GL-MKL for Heterogeneous Feature Selection

Recalled that MKL has been applied for feature selection. It simply considers the use of each feature attribute to construct the base kernel, and the learning process is to determine the associated weight for each kernel. If the weight is zero, the corresponding feature is redundant and thus is discarded. However, this method cannot be easily extended to heterogeneous feature selection, since each feature exhibits a unique property and thus has a different distribution in its attribute values.

Algorithm 1: Our Group Lasso Regularized MKL

```
Input: Data matrix $\mathbf{A}$, label $\mathbf{y}$, kernel function $k_{mj}$
Output: $\boldsymbol{\alpha}$ and $\boldsymbol{\beta} \in \mathbb{R}^{p \times k}$
begin
    $t \leftarrow 1$; $S^0 \leftarrow 1$; $\theta^0 \leftarrow 0$; $\beta^0_{mj} \leftarrow \frac{1}{p\sqrt{k}}$ $\forall\, m, j$;
    while $| 1 - \frac{\theta^{t-1}}{S^{t-1}} | \leq \epsilon$ do
        $\boldsymbol{\alpha}^t \leftarrow$ solve (9) with fixed $\boldsymbol{\beta}^{t-1}$;
        $S^t \leftarrow S(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^{t-1})$;
        $\boldsymbol{\beta}^t,\ \theta^t \leftarrow$ solve (10) with fixed $\boldsymbol{\alpha}^t$;
        $t \leftarrow t + 1$
```

TABLE I

UCI DATASETS

Dataset	Wpbc	Wdbc	Ionosphere	Wine
Number of instances	198	569	351	178
Number of features	31	30	34	13
Number of classes	2	2	2	3
Heterogeneous features	Yes	Yes	No	Yes

When using our proposed GL-MKL for heterogenous feature selection, we associate multiple types of kernels (with different kernel parameters) with each feature and consider them as a group, as shown in Fig. 3. The weight coefficients of each kernel, denoted by β_{mj}, are constrained by our group lasso regularization (see (9)), and thus our feature selection framework maintains sparsity across $\boldsymbol{\beta_m}$ but allows non-sparsity within $\boldsymbol{\beta_m}$. We note that the above sparsity among different groups (features) is preferable, since it meets the goal of feature selection (i.e., a compact set of features is desirable). On the other hand, our group lasso constraint allows the non-sparsity within each group of base kernels; this is to accommodate the presence of heterogeneous data, which will require different (and possibly multiple) kernels with distinct kernel parameters to describe the data in different feature spaces. Therefore, the use of our GL-MKL provides additional flexibility in fitting heterogeneous data, and this cannot be easily achieved by standard MKL or sequential-based feature selection methods. Another advantage over sequential-based approaches is that we do not require the prior knowledge on the preferable/optimal size of the feature subset to be selected. In the next section, we will evaluate our GL-MKL feature selection on a variety of datasets and show the effectiveness of our proposed method.

IV. Experiments

A. Experiment Setup

In this section, we evaluate the performance of recognition and feature selection on four UCI datasets[1] (see Table I

[1] The UCI datasets are available at `http://archive.ics.uci.edu/ml/`

for detailed descriptions) using our proposed GL-MKL and state-of-the-art MKL or sequential-based feature selection methods. Among the datasets we consider, all contain heterogeneous features except for the Ionosphere dataset. Besides, the Wine dataset contains multiple classes to be recognized, and we use the one-against-one strategy for classification in our experiments. It is worth noting that, the use of both types of data (heterogeneous or not) is to show the feasibility and robustness of our method for both types of problems. We do not normalize each feature attribute, since we observed that scaling each feature into the same range (by either linear or zero-mean) actually degrade the performance in several cases. This is why a method (like ours) which can directly handle heterogeneous feature data is preferable.

In our experiments, we compare our proposed GL-MKL with SVM (using all features), standard MKL (with ℓ_1 norm constraint on $\boldsymbol{\beta}$) [10], non-sparse MKL (with ℓ_2 norm constraint) [7], SFFS [3], and IFFS [4]. Gaussian kernels are used for nonlinear mapping in SVM and all MKL-based methods. For SVM, there is only one Gaussian kernel, and its parameter σ is chosen by cross-validation. For standard and non-sparse MKL, each feature dimension constructs a base kernel, and σ is the same for all base kernels (i.e. $k = 1$ in (9)). The σ value in these types of MKL-based methods is also selected by cross-validation for fair performance comparisons. To deal with heterogeneous features, we allow our proposed GL-MKL to choose among four different Gaussian kernels (with different σ) for each feature dimension in the p-dimensional data space, so that our GL-MKL has a total of $4 \times p$ base kernels. We then group these kernels at feature level to enforce the group lasso constraint. Recall that, since our approach learns the optimal kernels for feature selection, we do not require any validation data to select σ. For all our tests, we randomly select 80% of the data for training, and the remaining as the test set data. Each experiment is repeated with 5 random trials, and we present the average recognition rate and the average size of the selected feature subset for each case, as shown in Table II and III.

B. Comparisons with MKL-based Feature Selection Methods

We first compare our results with those using SVM, standard MKL (with ℓ_1 norm constraint on $\boldsymbol{\beta}$) [10], and non-sparse MKL (with ℓ_2 norm constraint) [7]; all methods do not assume that the optimal number of features are known in advance. For different datasets and feature selection methods, the averaged recognition performance and the size of the selected feature subset are presented in Table II. We note that the Wpbc, Wdbc, and Wine datasets contain heterogeneous data, and Ionosphere dataset is homogeneous data. While the nonlinear SVM does not have the capability of selecting discriminating features, it is used as the baseline classifier for comparisons.

From Table II, it can be observed that the recognition rates reported by non-sparse MKL, MKL, and our heterogenous feature selection method are statistically comparable to each other. However, it is worth noting that *our GL-MKL resulted in the most compact feature subset for each dataset*, as shown in the last column of Table II. Therefore, these results verify the use of our method for feature selection with comparable recognition performance achieved. As mentioned earlier, one of the advantages using our method is *the ability to deal with raw feature data from heterogeneous domains*, and thus the determination of proper feature normalization technique is not needed (which is required in most feature selection work when dealing with this kind of data).

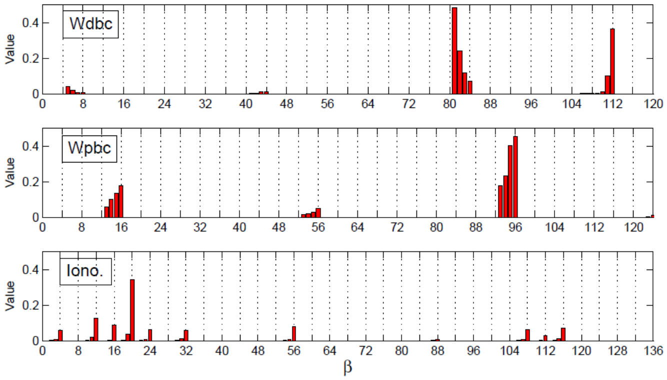

Fig. 4. The weights $\boldsymbol{\beta}$ determined for each base kernel and the corresponding feature. The x-axis is the index of β_{mj} (out of $k \times p$ = $4p$ for each dataset), and the y-axis is the associated weight. Each grid in the figure indicates a feature of interest, and the four components (red bars) in each grid represent the selected kernel weights.

Fig. 4 illustrates the weight β_{mj} for each base kernel and the associated feature using our GL-MKL for feature selection on the above datasets. For each feature (i.e., each grid in Fig. 4), we assign a total of $k = 4$ base kernels, and the feature selection results are depicted by their weights (red bars in Fig. 4). It can be seen that our approach selected a compact feature subset (e.g. only 3 out of 31 features were selected in Wpbc, as shown in Fig. 4), while the associated kernels need *not* be sparse. These results support that our approaches is able to provide a sparse yet discriminating feature subset, and achieves comparable performance as standard methods do.

C. Comparisons with Sequential Feature Selection Methods

We also compare our results with IFFS [4], and SFFS [3], which are state-of-the-art sequential-based feature selection methods and are popular due to its simplicity in implementation. The major concern of this type of approaches is that it needs to exhaustively search for the entire feature space for feature selection, or it will need the user to specify the preferable number of features to be selected. These concerns would prohibit their use for applications with high dimensional data. Since these two methods need the prior knowledge of the feature subset size, we use the number of features selected by our GL-MKL (determined in Table II), and compare the recognition performance using the same size of the feature subset.

From Table III, we see that our method outperforms SFFS and IFFS in terms of recognition on both heterogeneous or homogeneous feature data. Comparing with sequential-based

TABLE II

PERFORMANCE COMPARISONS. FOR EACH DATASET AND FEATURE SELECTION APPROACH, THE AVERAGE RECOGNITION ACCURACY (%) AND ITS STANDARD DEVIATION ARE PRESENTED, FOLLOWED BY THE AVERAGE SIZE OF THE SELECTED FEATURE SUBSET NOTED IN (). WHILE COMPARABLE RECOGNITION RATES AMONG DIFFERENT APPROACHES ARE OBSERVED IN THIS TABLE, OUR METHOD SELECTS THE SMALLEST FEATURE SUBSET FOR EACH DATASET (HIGHLIGHTED IN BOLD), AND THUS PRODUCES PREFERABLE FEATURE SELECTION RESULTS.

Dataset	SVM	Non-sparse MKL [7]	MKL [10]	Our method
Wdbc	93.81 ± 2.65 (30)	95.40 ± 1.81 (23)	95.22 ± 2.04 (7.2)	94.87 ± 4.26 (**4.2**)
Wpbc	76.12 ± 1.16 (31)	75.10 ± 2.46 (12)	75.71 ± 0.85 (3.2)	75.71 ± 1.33 (**2.4**)
Ionosphere	95.14 ± 2.17 (34)	87.43 ± 3.70 (33)	89.71 ± 1.86 (11.2)	93.71 ± 3.29 (**10.8**)
Wine	81.76 ± 2.46 (13)	92.94 ± 6.10 (11.8)	90.00 ± 4.92 (4.5)	95.29 ± 1.61 (**5.3**)

TABLE III

PERFORMANCE COMPARISONS WITH SEQUENTIAL-BASED FEATURE SELECTION METHODS. NOTE THAT BOTH SFFS AND IFFS USE ABOUT THE SAME NUMBER OF FEATURES SELECTED BY OUR METHOD.

Dataset	SFFS [3]	IFFS [4]	Our method
Wdbc	91.68 ± 2.28 (5)	94.51 ± 1.17 (5)	**94.87** ± 4.26 (4.2)
Wpbc	**76.92** ± 7.60 (3)	68.20 ± 8.82 (3)	75.71 ± 1.33 (2.4)
Ionosphere	89.42 ± 2.94 (11)	91.14 ± 1.89 (11)	**93.71** ± 3.29 (10.8)
Wine	90.85 ± 3.33 (5)	88.00 ± 5.54 (5)	**95.29** ± 1.61 (5.3)

feature selection methods, our method exhibits excellent ability in automatically determining the least number of features when producing satisfactory recognition performance.

To make the comparisons more complete, we also search for the entire feature space on heterogeneous datasets with binary and multiple classes using SFFS and IFFS, and we plot their corresponding averaged recognition rates in Fig. 5. It can be seen that, if the user does not specify the preferable number of features to be selected, one will need to exhaustively search for the optimal size of the feature subset using sequential-based methods. When using our GL-MKL, the optimal number of features can be selected automatically, while we achieve improved or comparable recognition performance (marked by black $*$ in Fig. 5) as the sequential-based methods do.

V. CONCLUSIONS

A novel group lasso regularized MKL (GL-MKL) was proposed in this paper for heterogeneous feature selection. The group lasso regularizer of our MKL, i.e. a mixed ℓ_1 and ℓ_2 norm constraint on the kernel weights, results in a compact feature subset, while the associated weights of the selected kernels for each feature are not necessarily sparse. The use of our GL-MKL for feature selection avoids the problem of normalizing the data when features are collected from different domains, which is why our method was able to handle heterogeneous feature data and to outperformed standard MKL or sequential-based methods in the experiments. Using our GL-MKL, the optimal kernels for each feature including the associated weights and kernel parameters can be determined simultaneously, while we do not exhaustively search for the entire feature space. Our feature selection method does not assume the size of the feature subset a priori as sequential selection methods do. Compared with existing feature selection approaches, our GL-MKL exhibited excellent capability in handling heterogeneous data, and selected the most compact feature subset for comparable or improved recognition performance.

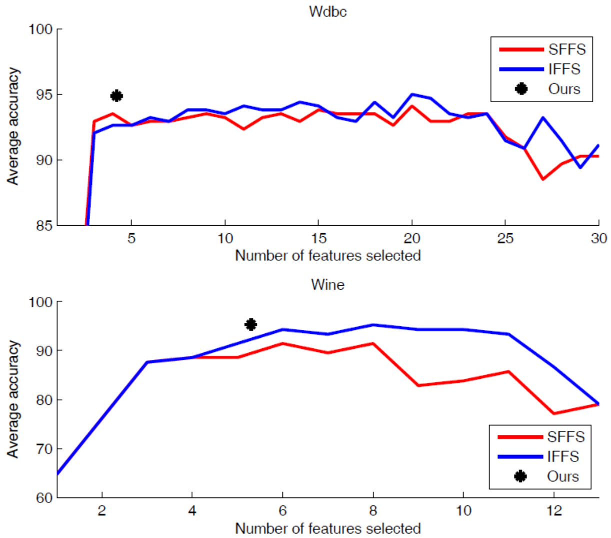

Fig. 5. Recognition performance of SFFS and IFFS on Wdbc and Wine datasets using different numbers of features. The one reported by our GL-MKL is denoted by $*$ in the figure, which achieves the best or comparable performance with the smallest numbers of features.

Future research directions will be directed at extensions of our proposed framework to both feature selection and fusion problems. In many real-world pattern recognition and computer vision applications, one typically needs to integrate multiple types of features in order to further increase the performance. Although fusion at the classifier level is possible, combining different features (e.g., from visual, audio, text, etc. domains) is known to provide/preserve more information then using a single one does. While a simple concatenation of feature vectors is possible for feature-level fusion, one will still need to properly normalize those feature vectors before training/testing. To address this issue, we will extend our proposed framework and evaluate its effectiveness in feature level fusion for improved performance.

VI. ACKNOWLEDGEMENTS

This work is supported in part by the National Science Council of Taiwan under NSC99-2221-E-001-020 and NSC100-2631-H-001-013.

References

[1] A. Whitney, “A direct method of nonparametric measurement selection,” *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 1100–1103, 1971.

[2] T. Marill and D. Green, “On the effectiveness of receptors in recognition systems,” *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.

[3] P. Pudil, J. Novovičová, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letter*, vol. 15, pp. 1119–1125, November 1994.

[4] S. Nakariyakul and D. Casasent, “An improvement on floating search algorithms for feature subset selection,” *Pattern Recognition*, vol. 42, no. 9, pp. 1932–1940, 2009.

[5] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2002.

[6] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the SMO algorithm,” *In Proceeding of International Conference on Machine Learning*, 2004.

[7] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, “Nonsparse multiple kernel learning,” in *In Proceeding of NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

[8] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[9] P. Gehler and S. Nowozin, “On Feature Combination for Multiclass Object Classification,” *In Proceeding of IEEE International Conference on Computer Vision*, pp. 221–228, 2009.

[10] A. D. Dileep and C. Chandra Sekhar, “Representation and feature selection using multiple kernel learning,” *In Proceeding of IEEE International Joint Conference on Neural Networks*, pp. 717–722, 2009.

[11] Z. Xu, R. Jin, J. Ye, M. R. Lyu, and I. King, “Non-monotonic feature selection,” *In Proceeding of International Conference on Machine Learning*, 2009.

[12] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B*, vol. 67, no. 2, pp. 301–320, 2005.

[13] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, “SimpleMKL,” *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[15] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, “Large scale multiple kernel learning,” *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[16] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.