Journal of Machine Learning Research 16 (2015) 617-652 Submitted 11/12; Revised 9/14; Published 3/15

Generalized Hierarchical Kernel Learning

Pratik Jawanpuria PRATIK.J@CSE.IITB.AC.IN
Jagarlapudi Saketha Nath SAKETH@CSE.IITB.AC.IN
Ganesh Ramakrishnan GANESH@CSE.IITB.AC.IN
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai 400076, INDIA

Editor: Francis Bach

Abstract

This paper generalizes the framework of Hierarchical Kernel Learning (HKL) and illustrates its utility in the domain of rule learning. HKL involves Multiple Kernel Learning over a set of given base kernels assumed to be embedded on a directed acyclic graph. This paper proposes a two-fold generalization of HKL: the first is employing a generic ℓ_1/ℓ_ρ block-norm regularizer ($\rho \in (1, 2]$) that alleviates a key limitation of the HKL formulation. The second is a generalization to the case of multi-class, multi-label and more generally, multi-task applications. The main technical contribution of this work is the derivation of a highly specialized partial dual of the proposed generalized HKL formulation and an efficient mirror descent based active set algorithm for solving it. Importantly, the generic regularizer enables the proposed formulation to be employed in the Rule Ensemble Learning (REL) where the goal is to construct an ensemble of conjunctive propositional rules. Experiments on benchmark REL data sets illustrate the efficacy of the proposed generalizations.

Keywords: multiple kernel learning, mixed-norm regularization, multi-task learning, rule ensemble learning, active set method

1. Introduction

A Multiple Kernel Learning (MKL) (Lanckriet et al., 2004; Bach et al., 2004) framework for construction of sparse linear combinations of base kernels embedded on a directed acyclic graph (DAG) was recently proposed by Bach (2008). Since the DAG induces hierarchical relations between the base kernels, this framework is more commonly known as Hierarchical Kernel Learning (HKL). It has been established that HKL provides a powerful algorithm for task specific non-linear feature selection. HKL employs a carefully designed ℓ_1/ℓ_2 block-norm regularizer: ℓ_1-norm across some predefined components associated with the DAG and ℓ_2-norm within each such component. However, the sparsity pattern of kernel (feature) selection induced by this regularizer is somewhat restricted: *a kernel is selected only if the kernels associated with all its ancestors in the DAG are selected.* In addition, it can be proved that the weight of the kernel associated with a (selected) node will always be greater than the weight of the kernels associated with its descendants. Such a restricted selection pattern and weight bias may limit the applicability of HKL in real world problems.

This paper proposes a two-fold generalization of HKL. The first is employing a $\ell_1/\ell_\rho, \rho \in (1, 2)$, block-norm regularizer that mitigates the above discussed weight and selection bias

among the kernels, henceforth termed as gHKL. Note that for the special case of $\rho = 2$, gHKL renders the HKL regularizer. Further, gHKL is generalized to the paradigm of Multi-task Learning (MTL), where multiple related tasks need to be learnt jointly. We consider the MTL setup where the given learning tasks share a common sparse feature space (Lounici et al., 2009; Jawanpuria and Nath, 2011; Obozinski et al., 2011). Our goal is to construct a shared sparse feature representation that is suitable for all the given related tasks. We pose the problem of learning this shared feature space as that of learning a shared kernel, common across all the tasks. The proposed generalization is henceforth referred to as gHKL$_{\rm MT}$. In addition to learning a common feature representation, gHKL$_{\rm MT}$ is generic enough to model additional correlations existing among the given tasks.

Though employing a $\ell_1/\ell_\rho, \rho \in (1, 2)$, regularizer is an incremental modification to the HKL formulation, devising an algorithm for solving it is not straightforward. The projected gradient descent employed in the active set algorithm for solving HKL (Bach, 2008) can no longer be employed for solving gHKL as projections onto ℓ_ρ-norm balls are known to be significantly more challenging than those onto ℓ_1-norm balls (Liu and Ye, 2010). Hence naive extensions of the existing HKL algorithm will not scale well. Further, the computational challenge is compounded with the generalization for learning multiple tasks jointly. The key technical contribution of this work is the derivation of a highly specialized partial dual of the gHKL/gHKL$_{\rm MT}$ formulations and an efficient mirror descent (Ben-Tal and Nemirovski, 2001; Beck and Teboulle, 2003) based active set algorithm for solving it. The dual presented here is an elegant convex optimization problem with a Lipschitz continuous objective and constrained over a simplex. Moreover, the gradient of the objective can be obtained by solving a known and well-studied variant of the MKL formulation. This motivates employing the mirror descent algorithm that is known to solve such problems efficiently. Further efficiency is brought in by employing an active set method similar in spirit to that in Bach (2008).

A significant portion of this paper focuses on the application of Rule Ensemble Learning (REL) (Dembczyński et al., 2010, 2008), where HKL has not been previously explored. Given a set of basic propositional features describing the data, the goal in REL is to construct a compact ensemble of conjunctions with the given propositional features that generalizes well for the problem at hand. Such ensembles are expected to achieve a good trade-off between interpretability and generalization ability. REL approaches (Cohen and Singer, 1999; Friedman and Popescu, 2008; Dembczyński et al., 2010) have additionally addressed the problem of learning a compact set of rules that generalize well in order to maintain their readability. One way to construct a compact ensemble is to consider a linear model involving all possible conjunctions of the basic propositional features and then performing a ℓ_1-norm regularized empirical risk minimization (Friedman and Popescu, 2008; Dembczyński et al., 2010). Since this is a computationally infeasible problem, even with moderate number of basic propositions, the existing methods either approximate such a regularized solution using strategies such as shrinkage (Friedman and Popescu, 2008; Dembczyński et al., 2010, 2008) or resort to post-pruning (Cohen and Singer, 1999). This work proposes to solve a variant of this regularized empirical risk minimization problem optimally using the framework of gHKL. The key idea is to define kernels representing every possible conjunction and arranging them on a DAG. The proposed gHKL regularizer is applied on this DAG of kernels, leading to a sparse combination of promising conjunctions. Note that

with such a setup, the size of the gHKL optimization problem is exponential in the number of basic propositional features. However, a key result in the paper shows that the proposed gHKL algorithm is guaranteed to solve this exponentially large problem with a complexity polynomial in the final active set[1] size. Simulations on benchmark binary (and multiclass) classification data sets show that gHKL (and gHKL_{MT}) indeed constructs a compact ensemble that on several occasions outperforms state-of-the-art REL algorithms in terms of generalization ability. These results also illustrate the benefits of the proposed generalizations over HKL: i) the ensembles constructed with gHKL (with low ρ values) involve fewer number of rules than with HKL; though the accuracies are comparable ii) gHKL_{MT} can learn rule ensemble on multiclass problems; whereas HKL is limited to two-class problems.

The rest of the paper[2] is organized as follows. Section 2 introduces the classical Multiple Kernel Learning setup, briefly reviews the HKL framework and summarizes the existing works in Multi-task Learning. In Section 3, we present the proposed gHKL and gHKL_{MT} formulations. The key technical derivation of the specialized dual is also presented in this section. The proposed mirror descent based active set algorithm for solving $\text{gHKL}/\text{gHKL}_{\text{MT}}$ formulations is discussed in Section 4. In Section 5, we propose to solve the REL problem by employing the gHKL formulation and discuss its details. In Section 6, we report empirical evaluations of gHKL and gHKL_{MT} formulations for REL on benchmark binary and multiclass data sets respectively. Section 7 concludes the paper.

2. Related Works

This section provides a brief introduction to the Multiple Kernel Learning (MKL) framework, the HKL setup and formulation (Bach, 2008, 2009) as well as the existing works in Multi-task Learning.

2.1 Multiple Kernel Learning Framework

We begin by discussing the regularized risk minimization framework (Vapnik, 1998), which has been employed in the proposed formulations.

Consider a learning problem like classification or regression and let its training data be denoted by $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, m \mid \mathbf{x}_i \in \mathcal{X}, y_i \in \mathbb{R}\ \forall i\}$, where $(\mathbf{x}_i, y_i)$ represents the i^{th} input-output pair. The aim is to learn an affine prediction function $F(\mathbf{x})$ that generalize well on unseen data. Given a positive definite kernel k that induces a feature map $\phi_k(\cdot)$, the prediction function can be written as: $F(\mathbf{x}) = \langle f, \phi_k(\mathbf{x})\rangle_{\mathcal{H}_k} - b$. Here $\mathcal{H}_k$ is the Reproducing Kernel Hilbert Space (RKHS) (Schölkopf and Smola, 2002) associated with the kernel k, endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_k}$, and $f \in \mathcal{H}_k, b \in \mathbb{R}$ are the model parameters to be learnt. A popular framework to learn these model parameters is the regularized risk minimization (Vapnik, 1998), which considers the following problem:

$$\min_{f \in \mathcal{H}_k, b \in \mathbb{R}} \frac{1}{2}\Omega(f)^2 + C\sum_{i=1}^{m} \ell(y_i, F(\mathbf{x}_i)), \tag{1}$$

1. Roughly, this is the number of selected conjunctions and is potentially far less than the total number of conjunctions.
2. Preliminary results of this work were reported in Jawanpuria et al. (2011).

where $\Omega(\cdot)$ is a norm based regularizer, $\ell : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is a suitable convex loss function and C is a regularization parameter. As an example, the support vector machine (SVM) (Vapnik, 1998) employs $\Omega(f) = \|f\|_{\mathcal{H}_k}$. From the *representer theorem* (Schölkopf and Smola, 2002), we know that the optimal f has the following form $f(\cdot) = \sum_i^m \alpha_i k(\cdot, \mathbf{x}_i)$ where $\alpha = (\alpha_i)_{i=1}^m$ is a vector of coefficients to be learnt.

It can be observed from above that the kernel definition plays a crucial role in defining the quality of the solution obtained by solving (1). Hence learning a kernel suitable to the problem at hand has been an active area of research over the past few years. One way to learn kernels is via the Multiple Kernel Learning (MKL) framework (Lanckriet et al., 2004; Bach et al., 2004). Lanckriet et al. (2004) proposed to learn the kernel k as a conic combination of the given base kernels $k_1, \ldots, k_l$: $k = \sum_{i=1}^l \eta_i k_i,\ \eta_i \geq 0\ \forall\ i$. Here $\eta = (\eta_i)_{i=1}^l$ is a coefficient vector to be (additionally) learnt in the optimization problem (1). In this setting, the feature map with respect to the kernel k is given by $\phi_k = (\sqrt{\eta_i}\phi_{k_i})_{i=1}^l$ (see Rakotomamonjy et al., 2008, for details). It is a weighted concatenation of feature maps induced by the individual base kernels. Hence, sparse kernel weights will result in a low dimensional ϕ_k. Some of the additional constraints on η explored in the existing MKL works are ℓ_1-norm constraint (Bach et al., 2004; Rakotomamonjy et al., 2008), ℓ_p-norm constraint ($p > 1$) (Kloft et al., 2011; Vishwanathan et al., 2010; Aflalo et al., 2011), etc.

2.2 Hierarchical Kernel Learning

Hierarchical Kernel Learning (HKL) (Bach, 2008) is a generalization of MKL and assumes a hierarchy over the given base kernels. The base kernels are embedded on a DAG and a carefully designed ℓ_1/ℓ_2 block-norm regularization over the associated RKHS is proposed to induce a specific sparsity pattern over the selected base kernels. We begin by discussing its kernel setup.

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be the given DAG with $\mathcal{V}$ denoting the set of vertices and $\mathcal{E}$ denoting the set of edges. The DAG structure entails relationships like parent, child, ancestor and descendant (Cormen et al., 2009). Let $D(v)$ and $A(v)$ represent the set of descendants and ancestors of the node v in the $\mathcal{G}$. It is assumed that both $D(v)$ and $A(v)$ include the node v. For any subset of nodes $\mathcal{W} \subset \mathcal{V}$, the *hull* and *sources* of $\mathcal{W}$ are defined as:

$$hull(\mathcal{W}) = \bigcup_{w\in\mathcal{W}} A(w), \quad sources(\mathcal{W}) = \{w \in \mathcal{W} \mid A(w) \cap \mathcal{W} = \{w\}\}.$$

The size and complement of $\mathcal{W}$ are denoted by $|\mathcal{W}|$ and $\mathcal{W}^c$ respectively. Let $k_v : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be the *positive definite* kernel associated with the vertex $v \in \mathcal{V}$. In addition, let $\mathcal{H}_{k_v}$ be its associated RKHS and ϕ_{k_v} be its induced feature map. Given this, HKL employs the following prediction function:

$$F(\mathbf{x}) = \sum_{v\in\mathcal{V}} \langle f_v, \phi_{k_v}(\mathbf{x})\rangle_{\mathcal{H}_{k_v}} - b,$$

which is an affine model parameterized by $f = (f_v)_{v\in\mathcal{V}}$, the tuple with entries as $f_v \in \mathcal{H}_{k_v}$ and $b \in \mathbb{R}$. Some more notations follow: for any subset of nodes $\mathcal{W} \subset \mathcal{V}$, $f_\mathcal{W} = (f_v)_{v\in\mathcal{W}}$ and $\phi_\mathcal{W} = (\phi_v)_{v\in\mathcal{W}}$. In general, the entries in a vector are referred to using an appropriate subscript, i.e., entries in $\mathbf{u} \in \mathbb{R}^d$ are denoted by $u_1, \ldots, u_d$. The kernels are denoted by the lower case 'k' and the corresponding Gram matrices are denoted by the upper case 'K'.

HKL formulates the problem of learning the optimal prediction function F as the following regularized risk minimization problem:

$$\min_{f_v \in \mathcal{H}_{k_v} \forall v \in \mathcal{V}, b \in \mathbb{R}} \ \frac{1}{2} \left(\sum_{v \in \mathcal{V}} d_v \|f_{D(v)}\|_2 \right)^2 + C \sum_{i=1}^{m} \ell\left(y_i, F(\mathbf{x}_i)\right), \tag{2}$$

where $\|f_{D(v)}\|_2 = \left(\sum_{w \in D(v)} \|f_w\|^2 \right)^{\frac{1}{2}} \ \forall v \in \mathcal{V}$, $\ell(\cdot, \cdot)$ is a suitable convex loss function and $(d_v)_{v \in \mathcal{V}}$ are given non-negative parameters.

As is clear from (2), HKL employs a ℓ_1/ℓ_2 block-norm regularizer, which is known to promote group sparsity (Yuan and Lin, 2006). Its implications are discussed in the following. For most of $v \in \mathcal{V}$, $\|f_{D(v)}\|_2 = 0$ at optimality due to the sparsity inducing nature of the ℓ_1-norm. Moreover $(\|f_{D(v)}\|_2 = 0) \Rightarrow (f_w = 0 \ \forall w \in D(v))$. Thus it is expected that most f_v will be zero at optimality. This implies that the prediction function involves very few kernels. Under mild conditions on the kernels (being strictly positive), it can be shown that this hierarchical penalization induces the following sparsity pattern: $(f_w \neq 0) \Rightarrow (f_v \neq 0 \ \forall v \in A(w))$. In other words, if the prediction function employs a kernel k_w then it *certainly* employs *all* the kernels associated with the ancestor nodes of w.

Bach (2008) proposes to solve the following equivalent variational formulation:

$$\min_{\gamma \in \Delta_1} \ \min_{f_v \in \mathcal{H}_{k_v} \forall v \in \mathcal{V}, b \in \mathbb{R}} \ \frac{1}{2} \sum_{w \in \mathcal{V}} \delta_w(\gamma)^{-1} \|f_w\|^2 + C \sum_{i=1}^{m} \ell\left(y_i, F(\mathbf{x}_i)\right), \tag{3}$$

where $\Delta_1 = \left\{ \mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1 \right\}$ and $\delta_w(\gamma)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v}$. From the representer theorem (Schölkopf and Smola, 2002), it follows that the effective kernel employed in the HKL is: $k = \sum_{w \in \mathcal{V}} \delta_w(\gamma) k_w$. Since the optimization problem (3) has a ℓ_1-norm constraint over γ variables, most γ_v at optimality are expected to be zero. Moreover the kernel weight $\delta_w(\gamma)$ is zero whenever $\gamma_v = 0$ for any $v \in A(w)$. Thus, the HKL performs a sparse selection of the base kernels and can be understood as a generalization of the classical MKL framework. However, the sparsity pattern for the kernels has the following restriction: if a kernel is not selected then none of the kernels associated with its descendants are selected, as $(\gamma_v = 0) \Rightarrow (\delta_w(\gamma) = 0 \ \forall w \in D(v))$. For the case of strictly positive kernels, it follows that a kernel is selected only if all the kernels associated with its ancestors are selected. In addition, the following relationship holds among the kernels weights: $\delta_v(\gamma) \geq \delta_w(\gamma) \ \forall w \in D(v)$ (strict inequality holds if $\delta_w(\gamma) > 0$). Hence, the weight of the kernel associated with a (selected) node is always be greater than the weight of the kernels associated with its descendants.

Since the size of γ is same as that of $\mathcal{V}$ and since the optimal γ is known to be sparse, Bach (2008) proposes an active set based algorithm (Lee et al., 2007) for solving (3). At each iteration of the active set algorithm, (3) is solved with respect to only those variables in the active set via the projected gradient descent technique (Rakotomamonjy et al., 2008).

As illustrated in Bach (2008), the key advantage of HKL is in performing non-linear feature selection. For example, consider the case where the input space is $\mathcal{X} = \mathbb{R}^n$ and let I be power set of $\{1, \ldots, n\}$. Consider the following 2^n kernels arranged on the usual subset lattice: $k_i(\mathbf{x}, \mathbf{x}') = \Pi_{j \in i} x_j x'_j \ \forall i \in I$. HKL can be applied in this setup to select

the promising sub-products of the input features over all possible sub-products. Please refer to Bach (2008) for more such pragmatic examples of kernels and corresponding DAGs. The most interesting result in Bach (2008) is that in all these examples where the size of the DAG is exponentially large, the computational complexity of the active set algorithm is polynomial in the training set dimensions and the active set size. Importantly, the complexity is independent of $|\mathcal{V}|$!

Though encouraging, the above discussed weight bias (in favor of the kernels towards the top of the DAG) and restricted kernel selection pattern may limit the applicability of HKL in real world problems. For instance, in case of the sub-product kernel example mentioned above, the following is true: a sub-product is selected only if all the products including it are selected. This clearly may lead to selection of many redundant sub-products (features). In Section 3, we present the proposed generalization that provides a more flexible kernel selection pattern by employing a $\ell_1/\ell_\rho,\ \rho \in (1,2)$, regularizer. A key result of this paper (refer Corollary 6) is that for all the cases discussed in Bach (2008), the proposed mirror descent based active set algorithm for solving the generalization has a computational complexity that is still polynomial in the training set dimensions and the active set size. In other words, the proposed generalization does not adversely affect the computational feasibility of the problem and hence is an interesting result in itself.

2.3 Multi-task Learning

Multi-task Learning (Caruana, 1997; Baxter, 2000) focuses on learning several prediction tasks simultaneously. This is in contrast with the usual approach of learning each task separately and independently. The key underlying idea behind MTL is that an appropriate sharing of information while learning *related* tasks will help in obtaining better prediction models. Various definitions of task-relatedness have been explored over the past few years like proximity of task parameters (Baxter, 2000; Evgeniou and Pontil, 2004; Xue et al., 2007; Jacob et al., 2008; Jawanpuria and Nath, 2012) or sharing common feature space (Ando and Zhang, 2005; Ben-David and Schuller, 2008; Argyriou et al., 2008; Lounici et al., 2009; Obozinski et al., 2011). Many learning settings like multiclass classification, multi-label classification or learning vector-valued function may be viewed as a special case of multi-task learning.

In this work, we consider the common setting in which the task parameters share a simultaneously sparse structure: only a small number of input features are relevant for each of the tasks and the set of such relevant features is common across all the tasks (Turlach et al., 2005; Lounici et al., 2009). Existing works in this setting typically employ a group lasso penalty on the tasks parameters: ℓ_1/ℓ_2 block-norm (Lounici et al., 2009; Obozinski et al., 2011) or the ℓ_1/ℓ_∞ block-norm (Turlach et al., 2005; Negahban and Wainwright, 2009). Thus, they propose a multi-task regularizer of the form: $\Omega(f_1,\ldots,f_T) = \sum_{i=1}^{d}\left(\sum_{t=1}^{T}|f_{ti}|^q\right)^{\frac{1}{q}}$ where the input feature space is assumed to be d dimensional, f_t is the task parameter of the t^{th} task and $f_t = (f_{ti})_{i=1,\ldots,d}$ and $q = \{2,\infty\}$. Note that in addition to (sparse) shared feature selection, the ℓ_1/ℓ_∞ block-norm penalty also promote proximity among the task parameters.

We pose the problem of learning the shared features as that of learning a shared kernel, whose induced feature space is common across all the tasks. The shared kernel is

constructed as a sparse combination of the given base kernels. A hierarchical relationship exists over the given kernels (feature spaces). We employ a graph based ℓ_1/ℓ_ρ block-norm regularization over the task parameters that enable non-linear feature selection for multiple tasks simultaneously. The details of the proposed MTL formulation are discussed in the following section.

3. Generalized Hierarchical Kernel Learning

In this section, we present the proposed generalizations over HKL. As discussed earlier, the first generalization aims at mitigating the weight bias problem as well as the restrictions imposed on the kernel selection pattern of HKL, and is termed as gHKL. The gHKL formulation is then further generalized to the paradigm of MTL, the proposed formulation being termed as gHKL$_{\text{MT}}$. We begin by introducing the gHKL formulation.

3.1 gHKL Primal Formulation

Recall that HKL employs a ℓ_1/ℓ_2 block norm regularizer. As we shall understand in more detail later, a key reason for the kernel weight bias problem and the restricted sparsity pattern in HKL is the ℓ_2-norm regularization. One way to mitigate these restrictions is by employing the following generic regularizer:

$$\Omega_S(f) = \sum_{v \in \mathcal{V}} d_v \|f_{D(v)}\|_\rho, \tag{4}$$

where $f = (f_v)_{v \in \mathcal{V}}$, $\|f_{D(v)}\|_\rho = \left(\sum_{w \in D(v)} \|f_w\|^\rho\right)^{\frac{1}{\rho}}$ and $\rho \in (1, 2]$. The implications of the ℓ_1/ℓ_ρ block-norm regularization are discussed in the following. Since the ℓ_1-norm promotes sparsity, it follows that $\|f_{D(v)}\|_\rho = 0$ (that is $f_w = 0\ \forall w \in D(v)$) for most $v \in \mathcal{V}$. This phenomenon is similar as in HKL. But now, even in cases where $\|f_{D(v)}\|_\rho$ is not forced to zero by the ℓ_1-norm, many components of $f_{D(v)}$ tend to zero[3] (that is $f_w \to \mathbf{0}$ for many $w \in D(v)$) as the value of ρ tends to unity. Also note that $\rho = 2$ renders the HKL regularizer. To summarize, the proposed gHKL formulation is

$$\min_{f_v \in \mathcal{H}_{k_v} \forall v \in \mathcal{V}, b \in \mathbb{R}} \ \frac{1}{2}(\Omega_S(f))^2 + C \sum_{i=1}^{m} \ell\left(y_i, F(\mathbf{x}_i)\right). \tag{5}$$

We next present the gHKL$_{\text{MT}}$ formulation, which further generalizes gHKL to MTL paradigm.

3.2 gHKL$_{\text{MT}}$ Primal Formulation

We begin by introducing some notations for the multi-task learning setup. Let T be the number of tasks and let the training data for the t^{th} task be denoted by $\mathcal{D}_t = \{(\mathbf{x}_{ti}, y_{ti}),\ i = 1, \ldots, m \mid \mathbf{x}_{ti} \in \mathcal{X}, y_{ti} \in \mathbb{R}\ \forall i\}$, where $(\mathbf{x}_{ti}, y_{ti})$ represents the i^{th} input-output pair of the

3. Note that as ℓ_ρ-norm ($\rho > 1$) is differentiable, it rarely induce sparsity (Szafranski et al., 2010). However, as $\rho \to 1$, they promote only a few leading terms due to the high curvatures of such norms (Szafranski et al., 2007). In order to obtain a sparse solution in such cases, thresholding is commonly employed by existing ℓ_p-MKL ($\rho > 1$) algorithms (Vishwanathan et al., 2010; Orabona et al., 2012; Jain et al., 2012; Jawanpuria et al., 2014). We employed thresholding in our experiments.

t^{th} task. For the sake of notational simplicity, it is assumed that the number of training examples is same for all the tasks. The prediction function for the t^{th} task is given by: $F_t(\mathbf{x}) = \sum_{v\in\mathcal{V}} \langle f_{tv}, \phi_{k_v}(\mathbf{x})\rangle_{\mathcal{H}_{k_v}} - b_t$, where $f_t = (f_{tv})_{v\in\mathcal{V}}$ and b_t are the task parameters to be learnt. We propose the following regularized risk minimization problem for estimating these task parameters and term it as gHKL$_{\text{MT}}$:

$$\min_{f_t, b_t \forall t} \frac{1}{2} \Bigg(\underbrace{\sum_{v\in\mathcal{V}} d_v \left(\sum_{w\in D(v)} (Q_w(f_1,\ldots,f_T))^{\rho} \right)^{\frac{1}{\rho}}}_{\Omega_T(f_1,\ldots,f_T)} \Bigg)^2 + C \sum_{t=1}^{T} \sum_{i=1}^{m} \ell(y_{ti}, F_t(\mathbf{x}_{ti})), \tag{6}$$

where $\rho \in (1, 2]$ and $Q_w(f_1, \ldots, f_T)$ is a norm-based multi-task regularizer on the task parameters f_{tw} $\forall t$. In the following, we discuss the effect of the above regularization. Firstly, there is a ℓ_1-norm regularization over the group of nodes (feature spaces) and a ℓ_ρ-norm regularization within each group. This ℓ_1/ℓ_ρ block-norm regularization is same as that of gHKL and will have the same effect on the sparsity pattern of the selected feature spaces (kernels). Hence, only a few nodes (feature spaces) will be selected by the gHKL$_{\text{MT}}$ regularizer $\Omega_T(f_1, \ldots, f_T)$. Secondly, nature of the task relatedness within each (selected) feature space is governed by the $Q_w(f_1, \ldots, f_T)$ regularizer.

For instance, consider the following definition of $Q_w(f_1, \ldots, f_T)$ (Lounici et al., 2009; Jawanpuria and Nath, 2011):

$$Q_w(f_1, \ldots, f_T) = \left(\sum_{t=1}^{T} \|f_{tw}\|^2 \right)^{\frac{1}{2}}. \tag{7}$$

The above regularizer couples the task parameters within each feature space via ℓ_2-norm. It encourages the task parameters within a feature space to be either zero or non-zero across all the tasks. Therefore, $\Omega_T(f_1, \ldots, f_T)$ based on (7) has the following effect: i) all the tasks will simultaneously select or reject a given feature space, and ii) overall only a few feature spaces will be selected in the gHKL style sparsity pattern.

Several multi-task regularizations (Evgeniou and Pontil, 2004; Evgeniou et al., 2005; Jacob et al., 2008) have been proposed to encourage proximity among the task parameters within a given feature space. This correlation among the tasks may be enforced while learning a shared sparse feature space by employing the following $Q_w(f_1, \ldots, f_T)$:

$$Q_w(f_1, \ldots, f_T) = \left(\mu \left\| \frac{1}{T+\mu} \sum_{t=1}^{T} f_{tw} \right\|^2 + \sum_{t=1}^{T} \left\| f_{tw} - \frac{1}{T+\mu} \sum_{t=1}^{T} f_{tw} \right\|^2 \right)^{\frac{1}{2}}, \tag{8}$$

where $\mu > 0$ is a given parameter. The above $Q_w(f_1, \ldots, f_T)$ consists of two terms: the first regularizes the mean while the second regularizes the variance of the task parameters in the feature space induced by kernel k_w. The parameter μ controls the degree of proximity among the task parameters, with lower μ encouraging higher proximity. Note that when $\mu = \infty$, (8) simplifies to (7). The gHKL$_{\text{MT}}$ regularizer $\Omega_T(f_1, \ldots, f_T)$ based on (8) has the

following effect: i) all the tasks will simultaneously select or reject a given feature space, ii) overall only a few feature spaces will be selected in the gHKL style sparsity pattern, and iii) within each selected feature space, the task parameters $f_{tw}\ \forall t$ are in proximity.

Thus, gHKL_{MT} framework provides a mechanism to learn a shared feature space across the tasks. In addition, it can also preserve proximity among the tasks parameters in the learnt feature space. As we shall discuss in the next section, more generic correlations among task parameters may be also modeled within the gHKL_{MT} framework.

It is clear that the gHKL optimization problem (5) may be viewed as a special case of the gHKL_{MT} optimization problem (7), with the number of tasks set to unity. Hence the rest of the discussion regarding dual derivation and optimization focuses primarily on gHKL_{MT} formulation.

3.3 gHKL_{MT} Dual Formulation

As mentioned earlier, due to the presence of the ℓ_ρ-norm term in gHKL_{MT} formulation, naive extensions of the projected gradient based active set method in Bach (2008) will be rendered computationally infeasible on real world data sets. Hence, we first re-write gHKL_{MT} formulation in an elegant form, which can then be solved efficiently. To this end, we note the following variational characterization of $\Omega_T(f_1, \ldots, f_T)$.

Lemma 1 *Given $\Omega_T(f_1, \ldots, f_T)$ and $Q_w(f_1, \ldots, f_T)$ as defined in (6) and (8) respectively, we have:*

$$\Omega_T(f_1, \ldots, f_T)^2 = \min_{\gamma \in \Delta} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v\ \forall v \in \mathcal{V}} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda)^{-1} Q_w(f_1, \ldots, f_T)^2, \tag{9}$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$, $\delta_w(\gamma, \lambda)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v \lambda_{vw}}$, $\Delta_1 = \left\{\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1\right\}$ and $\Delta_r^v = \left\{\mathbf{z} \in \mathbb{R}^{|D(v)|} \mid \mathbf{z} \geq 0, \sum_{w \in D(v)} \mathbf{z}_w^r \leq 1\right\}$.

Note that $\rho \in (1, 2) \Rightarrow \hat{\rho} \in (1, \infty)$. The proof of the above lemma is provided in Appendix A.2.

In order to keep the notations simple, in the remainder of this section, it is assumed that the learning tasks at hand are binary classification, i.e., $y_{ti} \in \{-1, 1\}\ \forall t, i$, and the loss function is the hinge loss. However, one can easily extend these ideas to other loss functions and learning problems. Refer Appendix A.8 for gHKL_{MT} dual formulation with general convex loss functions.

Lemma 2 *Consider problem (6) with the regularizer term replaced with its variational characterization (9) and the loss function as the hinge loss $\ell(y, F_t(\mathbf{x})) = \max(0, 1 - yF_t(\mathbf{x}))$. Then the following is a partial dual of it with respect to the variables $f_t, b_t\ \forall t = 1, \ldots, T$:*

$$\min_{\gamma \in \Delta_1} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v\ \forall v \in \mathcal{V}} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} G(\gamma, \lambda, \alpha), \tag{10}$$

where

$$G(\gamma, \lambda, \alpha) = \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \left(\sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) H_w \right) \mathbf{Y} \alpha,$$

$\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$, $S(\mathbf{y}_t, C) = \{\beta \in \mathbb{R}^m \mid 0 \leq \beta \leq C,\ \sum_{i=1}^m y_{ti}\beta_i = 0\}$, $\mathbf{y}_t = [y_{t1}, \ldots, y_{tm}]^\top$, $\mathbf{Y}$ *is the diagonal matrix corresponding to the vector* $[\mathbf{y}_1^\top, \ldots, \mathbf{y}_T^\top]^\top$, $\mathbf{1}$ *is a* $mT \times 1$ *vector with entries as unity,* $\delta_w(\gamma, \lambda)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v \lambda_{vw}}$, $\Delta_1 = \left\{\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1\right\}$, $\Delta_r^v = \left\{\mathbf{z} \in \mathbb{R}^{|D(v)|} \mid \mathbf{z} \geq 0, \sum_{w \in D(v)} \mathbf{z}_w^r \leq 1\right\}$, $\hat{\rho} = \frac{\rho}{2-\rho}$, *and* $H_w \in \mathbb{R}^{mT \times mT}$ *is the multi-task kernel matrix corresponding to the multi-task kernel* $h_w\ \forall w \in \mathcal{V}$. *The kernel function* h_w *is defined as follows:*

$$h_w(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j}) = k_w(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j}) B(t_1, t_2), \tag{11}$$

where B *is a* $T \times T$ *matrix.* $B = I$ *(identity matrix) when the multi-task regularizer (7) is employed in (6). Alternatively,* $B = I + \mathbf{1}\mathbf{1}^\top/\mu$ *(here* $\mathbf{1}$ *is a* $T \times 1$ *vector with entries as unity) in the case when the regularizer (8) is employed. The prediction function for the task* t_1 *is given by*

$$F_{t_1}(\mathbf{x}_{t_1 j}) = \sum_{t_2=1}^{T} \sum_{i=1}^{m} \bar{\alpha}_{t_2 i} y_{t_2 i} \left(\sum_{w \in \mathcal{V}} \delta_w(\bar{\gamma}, \bar{\lambda}) k_w(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j}) B(t_1, t_2) \right),$$

where $(\bar{\gamma}, \bar{\lambda}, \bar{\alpha})$ *is an optimal solution of (10).*

Proof The proof follows from the representer theorem (Schölkopf and Smola, 2002). Also refer to Appendix A.3. ■

This lemma shows that gHKL$_{\text{MT}}$ essentially constructs the same prediction function as an SVM with the effective multi-task kernel as: $h = \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) h_w$. Similarly, in the case of the gHKL, the effective kernel is $k = \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) k_w$ (since the terms T and B are unity). Here, as well as in the rest of the paper, we employ the symbols 'h' and 'H' for the multi-task kernel and the corresponding Gram matrix respectively.

The multi-task kernel (11) consists of two terms: the first term corresponds to the similarity between two instances $\mathbf{x}_{t_1 i}$ and $\mathbf{x}_{t_2 j}$ in the feature space induced by the kernel k_w. The second term corresponds to the correlation between the tasks t_1 and t_2. In the case of the regularizer (7), the matrix B simplifies to: $B(t_1, t_2) = 1$ if $t_1 = t_2$ and $B(t_1, t_2) = 0$ if $t_1 \neq t_2$, thereby making the kernel matrices $H_w (w \in \mathcal{V})$ block diagonal. Hence, the gHKL$_{\text{MT}}$ regularizer based on (7) promotes simultaneous sparsity in kernel selection among the tasks, without enforcing any additional correlations among the tasks.

In general, any $T \times T$ positive semi-definite matrix may be employed as B to model generic correlations among tasks. The multi-task kernel given by (11) will still remain a valid kernel (Sheldon, 2008; Álvarez et al., 2012). The matrix B is sometimes referred to as the output kernel in the setting of learning vector-valued functions. It is usually constructed from the prior domain knowledge.

We now discuss the nature of the optimal solution of (10). Most of the kernel weights $\delta_w(\gamma, \lambda)$ are zero at optimality of (10): $\delta_w(\gamma, \lambda) = 0$ whenever $\gamma_v = 0$ or $\lambda_{vw} = 0$ for any $v \in A(w)$. The vector γ is sparse due to ℓ_1-norm constraint in (10). In addition, $\rho \to 1 \Rightarrow \hat{\rho} \to 1$. Hence the vectors $\lambda_v\ \forall v \in \mathcal{V}$ get close to becoming sparse as $\rho \to 1$ due to the $\ell_{\hat{\rho}}$-norm constraint in (10). The superimposition of these two phenomena leads to a

flexible[4] sparsity pattern in kernel selection. This is explained in detail towards the end of this section.

Note that $\rho = 2 \Rightarrow \lambda_{vw} = 1 \ \forall v \in A(w), w \in \mathcal{W}$ at optimality in (10). Hence for $\rho = 2$, the minimization problem in (10) can be efficiently solved using a projected gradient method (Rakotomamonjy et al., 2008; Bach, 2009). However, as established in Liu and Ye (2010), projection onto the kind of feasibility set in the minimization problem in (10) is computationally challenging for $\rho \in (1, 2)$. Hence, we wish to re-write this problem in a relatively simpler form that can be solved efficiently. To this end, we present the following important theorem.

Theorem 3 *The following is a dual of (6) considered with the hinge loss function, and the objectives of (6) (with the hinge loss), (10) and (12) are equal at optimality:*

$$\min_{\eta \in \Delta_1} \ g(\eta), \tag{12}$$

where $g(\eta)$ is the optimal objective value of the following convex problem:

$$\max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \ \mathbf{1}^\top \alpha - \frac{1}{2} \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}, \tag{13}$$

where $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}$, $\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$, $S(\mathbf{y}_t, C) = \{\beta \in \mathbb{R}^m \mid 0 \leq \beta \leq C, \ \sum_{i=1}^m y_{ti} \beta_i = 0\}$, $\mathbf{y}_t = [y_{t1}, \ldots, y_{tm}]^\top$, $\mathbf{Y}$ is the diagonal matrix corresponding to the vector $[\mathbf{y}_1^\top, \ldots, \mathbf{y}_T^\top]^\top$, $\mathbf{1}$ is a $mT \times 1$ vector with entries as unity, $\bar{\rho} = \frac{\hat{\rho}}{\hat{\rho}-1}$, $\hat{\rho} = \frac{\rho}{2-\rho}$, $\Delta_1 = \left\{ \mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1 \right\}$, and $H_w \in \mathbb{R}^{mT \times mT}$ is the multi-task kernel matrix corresponding to the multi-task kernel (11).

The key idea in the proof of the above theorem is to eliminate the λ variables and the details are presented in Appendix A.5. The expression for the prediction function F, in terms of the variables η and α, is provided in Appendix A.9.

This theorem provides some key insights: firstly, we have that (12) is essentially a ℓ_1-norm regularized problem and hence it is expected that most η will be zero at optimality. Since $(\eta_v = 0) \Rightarrow (\zeta_w(\eta) = 0 \ \forall w \in D(v))$, it follows that most nodes in $\mathcal{V}$ will not contribute in the optimization problems (12) and (13). Secondly, in a single task learning setting ($T = 1$), the problem in (13) is equivalent to the $\ell_{\hat{\rho}}$-norm MKL dual problem (Kloft et al., 2011) with the base kernels as $(\zeta_v(\eta))^{\frac{1}{\bar{\rho}}} k_v \ \forall v \in \mathcal{V} \ni \zeta_v(\eta) \neq 0$. The optimization problem (13) essentially learns an effective kernel of the form $h = \sum_{v \in \mathcal{V}} \theta_v \left(\zeta_v(\eta)\right)^{\frac{1}{\bar{\rho}}} h_v$, where the θ are intermediate optimization variables constrained to be non-negative and lie within a $\ell_{\hat{\rho}}$-norm ball. The expression for θ in terms of the variables η and α is provided in Appendix A.9.

The variable θ influence the nature of the effective kernel h in two important ways: i) it follows from the expression of θ that

$$\theta_v \left(\zeta_v(\eta)\right)^{\frac{1}{\bar{\rho}}} \propto \zeta_v(\eta) \left(\alpha^\top \mathbf{Y} H_v \mathbf{Y} \alpha \right)^{\frac{1}{(\hat{\rho}-1)}}.$$

4. The HKL dual formulation (Bach, 2009) is a special case of (10) with $\rho = 2$, $T = 1$ and $B = 1$. When $\rho = 2$, $\hat{\rho} = \infty$. This implies $\lambda_{vw} = 1 \ \forall \ v \in A(w), \ w \in \mathcal{V}$ at optimality, resulting in the weight bias towards kernels embedded in the ancestor nodes and restricted sparsity pattern in kernel selection

Algorithm 1 Active Set Algorithm - Outline

Input: Training data $\mathcal{D}$, the kernels (k_v) embedded on the DAG $(\mathcal{V})$, the $T \times T$ matrix B that models task correlations and tolerance ϵ.
Initialize the active set $\mathcal{W}$ with $sources(\mathcal{V})$.
Compute η, α by solving (14)
while Optimal solution for (12) is NOT obtained **do**
  Add *some* nodes to $\mathcal{W}$
  Recompute η, α by solving (14)
end while
Output: $\mathcal{W}, \eta, \alpha$

The above relation implies that the weight of the kernel h_v in the DAG $\mathcal{V}$ is not only dependent on the position[5] of the node v, but also on the suitability of the kernel h_v to the problem at hand. This helps in mitigating the kernel weight bias in favour of the nodes towards the top of the DAG from gHKL_{MT}, but which is present in HKL, and ii) as $\rho \to 1$ (and hence as $\hat{\rho} \to 1$), the optimal θ get close to becoming sparse (Szafranski et al., 2007; Orabona et al., 2012). This superimposed with the sparsity of η promotes a more flexible sparsity pattern in kernel selection that HKL, especially when $\rho \to 1$.

Next, we propose to solve the problem (12) by exploiting the sparsity pattern of the η variables and the corresponding $\zeta(\eta)$ terms at optimality. We discuss it in detail in the following section.

4. Optimization Algorithm

Note that problem (12) remains the same whether solved with the original set of variables (η) or when solved with only those $\eta_v \neq 0$ at optimality (refer Appendix A.4 for details). However the computational effort required in the latter case can be significantly lower since it involves low number of variables and kernels. This motivates us to explore an active set algorithm, which is similar in spirit to that in Bach (2008).

An outline of the proposed active set algorithm is presented in Algorithm 1. The algorithm starts with an initial guess for the set $\mathcal{W}$ such that $\eta_w \neq 0$ $(\forall w \in \mathcal{W})$ at the optimality of (12). This set $\mathcal{W}$ is called the active set. Since the weight associated with the kernel h_w will be zero whenever $\eta_v = 0$ for any $v \in A(w)$, the active set $\mathcal{W}$ must contain $sources(\mathcal{V})$, else the problem has a trivial solution. Hence, the active set is initialized with $sources(\mathcal{V})$. At each iteration of the algorithm, (12) is solved with variables restricted to those in $\mathcal{W}$:

$$\min_{\eta \in \Delta_1} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \mathbf{1}^\top \alpha - \frac{1}{2} \left(\sum_{w \in \mathcal{W}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}. \tag{14}$$

In order to formalize the active set algorithm, we need: i) an efficient algorithm for solving problem (14), ii) a condition for verifying whether a candidate solution is optimal

5. Similar to the δ_v function in HKL (3), it follows from the definition of ζ_v that $\zeta_v(\eta) \geq \zeta_w(\eta)$ $\forall w \in D(v)$ (strict inequality holds if $\zeta_w(\eta) > 0$).

Algorithm 2 Mirror Descent Algorithm for solving (14)

Input: Gram matrices H_w $(w \in \mathcal{W})$ and the regularization parameter C
Initialize $\eta_{\mathcal{W}}$ $(w \in \mathcal{W})$ such that $\eta_{\mathcal{W}} \in \Delta_1$ (warm-start may be used)
Iteration number: $i = 0$
while convergence criterion is not met[6] **do**
  $i = i + 1$
  Compute $\zeta_w(\eta_{\mathcal{W}})$ $\forall w \in \mathcal{W}$ (Theorem 3)
  Compute $\alpha_{\mathcal{W}}$ (13) using $\ell_{\hat{\rho}}$-norm MKL algorithm with kernels as $\left((\zeta_w(\eta_{\mathcal{W}}))^{\frac{1}{\hat{\rho}}} H_w\right)_{w \in \mathcal{W}}$
  Compute $\nabla g(\eta_{\mathcal{W}})$ as in (24)
  Compute step size $s = \sqrt{\log(|\mathcal{W}|)/i \cdot \|\nabla g(\eta_{\mathcal{W}}))\|_\infty^2}$
  Compute $\eta_w = \exp\left(1 + \log(\eta_w) - s \cdot \nabla g(\eta_{\mathcal{W}})_w\right)$ $\forall w \in \mathcal{W}$
  Normalize $\eta_w = \frac{\eta_w}{\sum_{v \in \mathcal{W}} \eta_v}$ $\forall w \in \mathcal{W}$
end while
Output: $\eta_{\mathcal{W}}, \alpha_{\mathcal{W}}$

with respect to the optimization problem (12), and iii) a procedure for building/improving the active set after each iteration.

We begin with the first. We propose to solve the optimization problem (14) using the mirror descent algorithm (Ben-Tal and Nemirovski, 2001; Beck and Teboulle, 2003). Mirror descent algorithm is known to efficiently solve convex programs with Lipschitz continuous and differentiable objectives constrained over a convex compact set. It achieves a near-optimal convergence rate whenever the feasibility set is a simplex (which is true in our optimization problem (14)). Mirror descent is close in spirit to the projected gradient descent algorithm and hence assumes that an oracle for computing the gradient of the objective is available.

Following the common practice of smoothing (Bach, 2009), in the rest of the paper, we employ $\zeta_w((1-\varepsilon)\eta + \frac{\varepsilon}{|\mathcal{V}|})$ instead[7] of $\zeta_w(\eta)$ in (13) with $\varepsilon > 0$. The following theorem establishes the applicability of mirror descent for solving (14):

Theorem 4 *The function $g(\eta)$ given by (13) is convex. Also, the expression for the i^{th} entry in the gradient $(\nabla g(\eta))_i$ is given in (24). If all the eigenvalues of the Gram matrices H_w are finite and non-zero, then g is Lipschitz continuous.*

The proof of the above theorem is technical and is provided in Appendix A.6.

Algorithm 2 summarizes the proposed mirror descent based algorithm for solving (14). One of its steps involve computing $\nabla g(\eta_{\mathcal{W}})$ (expression provided in (24)), which in turn requires solving (13). As noted before, (13) is similar to the $\ell_{\hat{\rho}}$-norm MKL problem (Kloft et al., 2011) but with a different feasibility set for the optimization variables α. Hence, (13) can be solved by employing a modified cutting planes algorithm (Kloft et al., 2011) or a modified sequential minimal optimization (SMO) algorithm (Platt, 1999; Vishwanathan

6. Relative objective gap between two successive iteration being less than a given tolerance ϵ is taken to be the convergence criterion. Objective here is the value of $g(\eta_{\mathcal{W}})$, calculated after $\ell_{\hat{\rho}}$-norm MKL step.
7. Note that this is equivalent to smoothing the regularizer Ω_T while preserving its sparsity inducing properties (Bach, 2009).

Algorithm 3 Active Set Algorithm

Input: Training data $\mathcal{D}$, the kernels (k_v) embedded on the DAG $(\mathcal{V})$, the $T \times T$ matrix B that models task correlations and tolerance ϵ.
Initialize the active set $\mathcal{W}$ with $sources(\mathcal{V})$
Compute η, α by solving (14) using Algorithm 2
while sufficient condition for optimality (15) is not met **do**
  Add those nodes to $\mathcal{W}$ that violate (15)
  Recompute η, α by solving (14) using Algorithm 2
end while
Output: $\mathcal{W}, \eta, \alpha$

et al., 2010). Empirically, we observed the SMO based algorithm to be much faster than the cutting planes algorithm for gHKL$_{\text{MT}}$ (and gHKL) with SVM loss functions. In the special case of $\rho = 2, T = 1$ and $B = 1$, (13) is simply a regular SVM problem.

Now we turn our attention to the second requirement of the active set algorithm: a condition to verify the optimality of a candidate solution. We present the following theorem that provides a sufficient condition for verifying optimality of a candidate solution.

Theorem 5 *Suppose the active set $\mathcal{W}$ is such that $\mathcal{W} = hull(\mathcal{W})$. Let $(\eta_{\mathcal{W}}, \alpha_{\mathcal{W}})$ be a $\epsilon_{\mathcal{W}}$-approximate optimal solution of (14), obtained from Algorithm (2). Then, it is an optimal solution for (12) with a duality gap less than ϵ if the following condition holds:*

$$\max_{u \in sources(\mathcal{W}^c)} \alpha_{\mathcal{W}}^\top \mathbf{Y} \mathcal{K}_u \mathbf{Y} \alpha_{\mathcal{W}} \leq \left(\sum_{w \in \mathcal{W}} \zeta_w(\eta_{\mathcal{W}}) \left(\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} + 2(\epsilon - \epsilon_{\mathcal{W}}), \qquad (15)$$

where $\mathcal{K}_u = \sum_{w \in D(u)} \frac{H_w}{\left(\sum_{v \in A(w) \cap D(u)} d_v\right)^2}$.

The proof is provided in Appendix A.7. It closely follows that for the case of HKL (Bach, 2008).

The summary of the proposed mirror descent based active set algorithm is presented in Algorithm 3. At each iteration, Algorithm (3) verifies optimality of the current iterate by verifying the condition in (15). In case the current iterate does not satisfy this condition, the nodes in $sources(\mathcal{W}^c)$ that violate the condition (15) are included in the active set.[8] This takes care of the third requirement of the active set algorithm. The algorithm terminates if the condition (15) is satisfied by the iterate.

In the following, an estimate of the computational complexity of the active set algorithm is presented. Let W be the final active set size. The optimization problem (14) needs to be solved at most W times, assuming the worst case scenario of adding one node per active set iteration. Each run of the mirror descent algorithm requires at most $O(\log(W))$ iterations (Ben-Tal and Nemirovski, 2001; Beck and Teboulle, 2003). A conservative time complexity estimate for computing the gradient $\nabla g(\eta_{\mathcal{W}})$ by solving a variant of the $\ell_{\hat{\rho}}$-norm MKL problem (13) is $O(m^3 T^3 W^2)$. This amounts to $O(m^3 T^3 W^3 \log(W))$. As for the computational cost of the sufficient condition, let z denote the maximum out-degree

8. It is easy to see that with this update scheme, $\mathcal{W}$ is always equal to $hull(\mathcal{W})$, as required in Theorem 5.

of a node in $\mathcal{G}$, i.e., z is an upper-bound on the the maximum number of children of any node in $\mathcal{G}$. Then the size of $sources(\mathcal{W}^c)$ is upper-bounded by Wz. Hence, a total of $O(\omega m^2 T^2 W z)$ operations are required for evaluating the matrices $\mathcal{K}$ in (15), where ω is the complexity of computing a single entry in any $\mathcal{K}$. In all the pragmatic examples of kernels and the corresponding DAGs provided by Bach (2008), ω is polynomial in the training set dimensions. Moreover, caching of $\mathcal{K}$ usually renders ω to be a constant (Bach, 2009). Further, the total cost of the quadratic computation in (15) is $O(m^2 T^2 W^2 z)$. Thus the overall computational complexity is $O(m^3 T^3 W^3 \log(W) + \omega m^2 T^2 W z + m^2 T^2 W^2 z)$. More importantly, because the sufficient condition for optimality (Theorem 5) is independent of ρ, we have the following result:

Corollary 6 *In a given input setting, HKL algorithm converges in time polynomial in the size of the active set and the training set dimensions if and only if the proposed mirror descent based active set algorithm (i.e., gHKL$_{\mathrm{MT}}$ algorithm) has a polynomial time convergence in terms of the active set and training set sizes.*

The proof is provided in Appendix A.10.

In the next section, we present an application of the proposed formulation that illustrate the benefits of the proposed generalizations over HKL.

5. Rule Ensemble Learning

In this section, we propose a solution to the problem of learning an ensemble of decision rules, formally known as Rule Ensemble Learning (REL) (Cohen and Singer, 1999), employing the gHKL and gHKL$_{\mathrm{MT}}$ formulations. For the sake of simplicity, we only discuss the single task REL setting in this section, i.e., REL as an application of gHKL. Similar ideas can be applied to perform REL in multi-task learning setting, by employing gHKL$_{\mathrm{MT}}$. In fact, we present empirical results of REL in both single and multiple task learning settings in Section 6. We begin with a brief introduction to REL.

If-then decision rules (Rivest, 1987) are one of the most expressive and human readable representations for learned hypotheses. It is a simple logical pattern of the form: IF *condition* THEN *decision*. The *condition* consists of a conjunction of a small number of simple boolean statements (propositions) concerning the values of the individual input variables while the *decision* specifies a value of the function being learned. An instance of a decision rule from Quinlan's play-tennis example (Quinlan, 1986) is:

IF *HUMIDITY==normal* AND *WIND==weak* THEN *PlayTennis==yes*.

The dominant paradigm for induction of rule sets, in the form of decision list (DL) models for classification (Rivest, 1987; Michalski, 1983; Clark and Niblett, 1989), has been a greedy *sequential covering* procedure.

REL is a general approach that treats decision rules as base classifiers in an ensemble. This is in contrast to the more restrictive decision list models that are disjunctive sets of rules and use only one in the set for each prediction. As pointed out in Cohen and Singer (1999), boosted rule ensembles are in fact simpler, better-understood formally than other state-of-the-art rule learners and also produce comparable predictive accuracy.

REL approaches like SLIPPER (Cohen and Singer, 1999), LRI (Weiss and Indurkhya, 2000), RuleFit (Friedman and Popescu, 2008), ENDER/MLRules (Dembczyński et al., 2008, 2010) have additionally addressed the problem of learning a compact set of rules that generalize well in order to maintain their readability. Further, a number of rule learners like RuleFit, LRI encourage shorter rules (i.e., fewer conjunctions in the condition part of the rule) or rules with a restricted number of conjunctions, again for purposes of interpretability. We build upon this and define our REL problem as that of learning a small set of simple rules and their weights that leads to a good generalization over new and unseen data. The next section introduces the notations and the setup in context of REL.

5.1 Notations and Setup

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be the training data described using p basic (boolean) propositions, i.e., $\mathbf{x}_i \in \{0,1\}^p$. In case the input features are not boolean, such propositions can be derived using logical operators such as $==, \neq, \leq$ or $\geq$ over the input features (refer Friedman and Popescu, 2008; Dembczyński et al., 2008, for details). Let $\mathcal{V}$ be an index-set for all possible conjunctions with the p basic propositions and let $\phi_v : \mathbb{R}^n \mapsto \{0,1\}$ denote the v^{th} conjunction in $\mathcal{V}$. Let $f_v \in \mathbb{R}$ denote the weight for the conjunction ϕ_v. Then, the rule ensemble to be learnt is the weighted combination of these conjunctive rules: $F(\mathbf{x}) = \sum_{v \in \mathcal{V}} f_v \phi_v(\mathbf{x}) - b$, where perhaps many weights (f_v) are equal to zero.

One way to learn the weights is by performing a ℓ_1-norm regularized risk minimization in order to select few promising conjunctive rules (Friedman and Popescu, 2008; Dembczyński et al., 2008, 2010). However, to the best of our knowledge, rule ensemble learners that identify the need for sparse f, either approximate such a regularized solution using strategies such as shrinkage (Rulefit, ENDER/MLRules) or resort to post-pruning (SLIPPER). This is because the size of the minimization problem is exponential in the number of basic propositions and hence the problem becomes computationally intractable with even moderately sized data sets. Secondly, conjunctive rules involving large number of propositions might be selected. However, such conjunctions adversely effect the interpretability. We present an approach based on the gHKL framework that addresses these issues.

We begin by noting that $\langle \mathcal{V}, \subseteq \rangle$ is a subset-lattice; hereafter this will be referred to as the *conjunction lattice*. In a conjunction lattice, $\forall\ v_1, v_2 \in \mathcal{V},\ v_1 \subseteq v_2$ *if and only if* the set of propositions in conjunction v_1 is a subset of those in conjunction v_2. As an example, (*HUMIDITY==normal*) is considered to be a subset of (*HUMIDITY==normal* AND *WIND==weak*). The top node of this lattice is a node with no conjunctions and is also *sources*($\mathcal{V}$). Its children, the second level nodes, are all the basic propositions, p in number. The third level nodes, children of these basic propositions, are the conjunctions of length two and so on. The bottom node at $(p+1)^{th}$ level is the conjunction of all basic propositions. The number of different conjunctions of length r is $\binom{p}{r}$ and the total number of nodes in this *conjunction lattice* is 2^p. Figure (1) shows a complete conjunction lattice with $p = 4$.

We now discuss how the proposed gHKL regularizer (5) provides an efficient and optimal solution to a regularized empirical risk minimization formulation for REL.

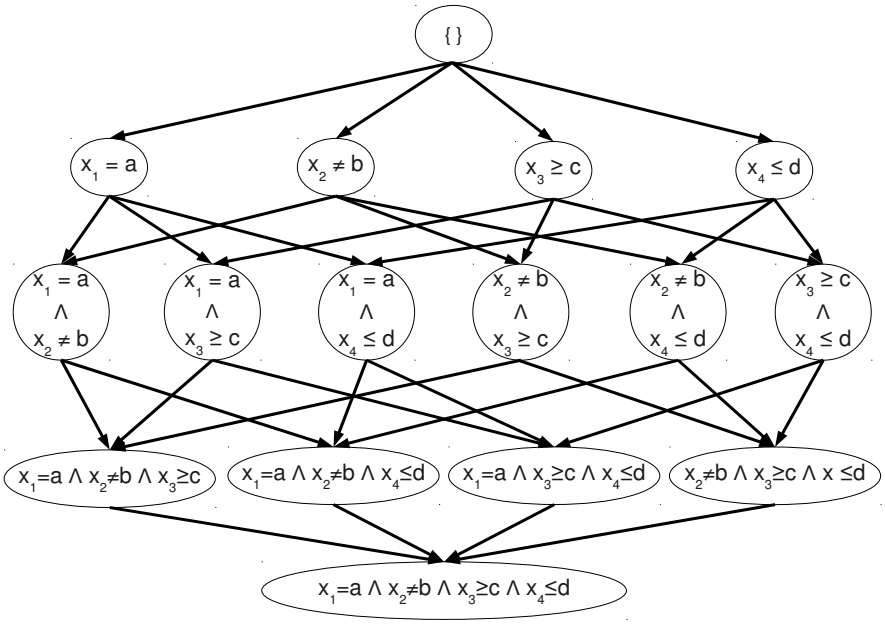

Figure 1: Example of a conjunction lattice with 4 basic propositions: $(x_1 = a)$, $(x_2 \neq b)$, $(x_3 \geq c)$ and $(x_4 \leq d)$. The input space consist of four features: x_1, x_2, x_3 and x_4. The number of nodes in conjunction lattice is exponential in the number of basic propositions. In this particular example, the number of nodes is 16 $(= 2^4)$.

5.2 Rule Ensemble Learning with gHKL

The key idea is to employ gHKL formulation (5) with the DAG as the conjunction lattice and the kernels as $k_v(\mathbf{x}_i, \mathbf{x}_j) = \phi_v(\mathbf{x}_i)\phi_v(\mathbf{x}_j)$ for learning an ensemble of rules. Note that with such a setup, the ℓ_1/ℓ_ρ block-norm regularizer in gHKL ($\Omega_S(f) = \sum_{v\in\mathcal{V}} d_v \|f_{D(v)}\|_\rho$) implies: 1) for most $v \in \mathcal{V}$, $f_v = 0$, and 2) for most $v \in \mathcal{V}$, $f_w = 0 \; \forall \; w \in D(v)$. In the context of the REL problem, the former statement is equivalent to saying: selection of a compact set of conjunctions is promoted, while the second reads as: selection of conjunctive rules with small number of propositions is encouraged. Thus, gHKL formulation constructs a compact ensemble of simple conjunctive rules. In addition, we set $d_v = a^{|S_v|}$ $(a > 1)$, where S_v is the set of basic propositions involved in the conjunction ϕ_v. Such a choice further encourages selection of short conjunctions and leads to the following elegant computational result:

Theorem 7 *The complexity of the proposed gHKL algorithm in solving the REL problem, with the DAG, the base kernels and the parameters d_v as defined above, is polynomial in the size of the active set and the training set dimensions. In particular, if the final active set size is W, then its complexity is given by $O(m^3W^3\log(W) + m^2W^2p)$.*

The proof is provided in Appendix A.11.

We end this section by noting the advantage of the generic regularizer in gHKL formulation over the that in HKL formulation in the context of REL application. Recall that

the sparsity pattern allowed by HKL has the following consequence: a conjunction is selected only after selecting all the conjunctions which are subsets of it. This, particularly in the context of REL, is psycho-visually redundant, because a rule with k propositional statements, if included in the result, will necessarily entail the inclusion of $(2^k - 1)$ more general rules in the result. This violates the important requirement for a small set (Friedman and Popescu, 2008; Dembczyński et al., 2008, 2010) of human-readable rules. The gHKL regularizer, with $\rho \in (1, 2)$, alleviates this restriction by promoting additional sparsity in selecting the conjunctions. We empirically evaluate the proposed gHKL based solution for REL application in the next section.

6. Experimental Results

In this section, we report the results of simulation in REL on several benchmark binary and multiclass classification data sets from the UCI repository (Blake and Lichman, 2013). The goal is to compare various rule ensemble learners on the basis of: (a) generalization, which is measured by the predictive performance on unseen test data, and (b) ability to provide compact set of simple rules to facilitate their readability and interpretability (Friedman and Popescu, 2008; Dembczyński et al., 2010; Cohen and Singer, 1999). The latter is judged using i) average number of rules learnt, and ii) average number of propositions per rule. The following REL approaches were compared.

- **RuleFit:** Rule ensemble learning algorithm proposed by Friedman and Popescu (2008). All the parameters were set to the default values mentioned by the authors. In particular, the model was set in the mixed linear-rule mode, average tree size was set 4 and maximum number of trees were kept as 500. The same configuration was also used by Dembczyński et al. (2008, 2010) in their simulations. This REL system cannot handle multi-class data sets and hence is limited to the simulations on binary classification data sets. Its code is available at `www-stat.stanford.edu/~jhf/R-RuleFit.html`.

- **SLI:** The SLIPPER algorithm proposed by Cohen and Singer (1999). Following Dembczyński et al. (2008, 2010), all parameters were set to their defaults. We retained the internal cross-validation for selecting the optimal number of rules.

- **ENDER:** State-of-the-art rule ensemble learning algorithm (Dembczyński et al., 2010). For classification setting, `ENDER` is same as `MLRules` (Dembczyński et al., 2008). The parameters were set to the default values suggested by the authors. The second order heuristic was used for minimization. Its code is available at `www.cs.put.poznan.pl/wkotlowski`.

- **HKL-ℓ_1-MKL:** A two-stage rule ensemble learning approach. In the first stage, HKL is employed to prune the exponentially large search space of all possible conjunctive rules and select a set of candidate rules (kernels). The rule ensemble is learnt by employing ℓ_1-MKL over the candidate set of rules. In both the stages, a three-fold cross validation procedure was employed to tune the C parameter with values in $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.

- **gHKL$_\rho$:** The proposed gHKL based REL formulation for binary classification problem. We considered three different values of ρ: 2, 1.5 and 1.1. Note that for binary

classification, $\rho = 2$ renders the HKL formulation (Bach, 2008). In each case, a three-fold cross validation procedure was employed to tune the C parameter with values in $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. As mentioned earlier, the parameters $d_v = 2^{|v|}$.

- **gHKL$_{\mathbf{MT}-\rho}$:** The proposed gHKL$_{\mathrm{MT}}$ based REL formulation for multiclass classification problem. For each class, a one-vs-rest binary classification task is created. Since we did not have any prior knowledge about the correlation among the classes in the data sets, we employed the multi-task regularizer (7) in the gHKL$_{\mathrm{MT}}$ primal formulation (6).

We considered three different values of ρ: 2, 1.5 and 1.1. Its parameters and cross validation details are same as that of gHKL$_\rho$. The implementations of both gHKL$_\rho$ and gHKL$_{\mathtt{MT}-\rho}$ are available at `http://www.cse.iitb.ac.in/~pratik.j/ghkl`.

Note that the above methods differ in the way they control the number of rules (M) in the ensemble. In the case of gHKL$_\rho$ (gHKL$_{\mathtt{MT}-\rho}$), M implicitly depends on the parameters: ρ, C and d_v. SLI has a parameter for maximum number of rules M_{max} and M is decided via a internal cross-validation such that $M \leq M_{max}$. For the sake of fairness in comparison with gHKL$_\rho$, we set $M_{max} = \max(M_{1.5}, M_{1.1})$, where M_ρ is the average number of rules obtained with gHKL$_\rho$ (gHKL$_{\mathtt{MT}-\rho}$). ENDER has an explicit parameter for the number of rules, which is also set to $\max(M_{1.5}, M_{1.1})$. In case of RuleFit, the number of rules in the ensemble is determined internally and is not changed by us.

6.1 Binary Classification in REL

This section summarizes our results on binary REL classification. Table 1 provides the details of the binary classification data sets. For every data set, we created 10 random train-test splits with 10% train data (except for MONK-3 data set, whose train-test split of $122 - 432$ instances respectively was already given in the UCI repository). Since many data sets were highly unbalanced, we report the average F1-score along with the standard deviation (Table 5 in Appendix A.12 reports the average AUC). The results are presented in Table 2. The best result, in terms of the average F1-score, for each data set is highlighted.

Data set		Num	Bias	p	$\lvert\mathcal{V}\rvert$	Data set		Num	Bias	p	$\lvert\mathcal{V}\rvert$
TIC-TAC-TOE	(TIC)	958	1.89	54	$\approx 10^{16}$	HEARTSTAT	(HTS)	270	0.8	76	$\approx 10^{22}$
B-CANCER-W	(BCW)	699	0.53	72	$\approx 10^{21}$	MONK-3	(MK3)	554	1.08	30	$\approx 10^{9}$
DIABETES	(DIA)	768	0.54	64	$\approx 10^{19}$	VOTE	(VTE)	232	0.87	32	$\approx 10^{9}$
HABERMAN	(HAB)	306	0.36	40	$\approx 10^{12}$	B-CANCER	(BCC)	277	0.41	76	$\approx 10^{22}$
HEARTC	(HTC)	296	0.85	78	$\approx 10^{23}$	MAM. MASS	(MAM)	829	0.94	46	$\approx 10^{13}$
BLOOD TRANS	(BLD)	748	3.20	32	$\approx 10^{9}$	LIVER	(LIV)	345	1.38	48	$\approx 10^{14}$

Table 1: Data sets used for binary REL classification. 'Num' is the number of instances in the data set while 'Bias' denotes the ratio of # of +ve and −ve instances. The number of number of basic propositions is 'p' and $|\mathcal{V}|$ represents the total number of possible conjunctions. For each numerical input feature, 8 basic propositions were derived. The letters in brackets are the acronym used for the corresponding data set in Table 2.

	RuleFit	**SLI**	**ENDER**	**HKL-ℓ_1-MKL**	**gHKL$_\rho$**		
					$\rho = 2$	$\rho = 1.5$	$\rho = 1.1$
TIC	0.517 ± 0.092 (57.7, 2.74)	0.665 ± 0.053 (10.3, 1.96)	0.668 ± 0.032 (187, 3.17)	0.749 ± 0.040 (74.8, 1.89)	0.889 ± 0.093 (161.7, 1.72)	0.897 ± 0.093 (186.6, 1.76)	$\mathbf{0.905 \pm 0.096}^*$ (157.6, 1.72)
BCW	0.879 ± 0.025 (17.5, 2.03)	$\mathbf{0.928 \pm 0.018}$ (4.4, 1.15)	0.900 ± 0.041 (21, 1.56)	0.925 ± 0.032 (27,1.03)	0.923 ± 0.032 (30.9, 1)	0.924 ± 0.032 (20, 1.03)	0.925 ± 0.032 (20.4, 1.02)
DIA	0.428 ± 0.052 (32.9, 2.66)	0.659 ± 0.027 (4.9, 1.42)	0.656 ± 0.027 (74.0, 2.65)	0.658 ± 0.028 (47.6, 1.40)	0.661 ± 0.018 (83.2, 1.31)	$\mathbf{0.663 \pm 0.017}$ (73.2, 1.17)	0.661 ± 0.023 (62.6, 1.27)
HAB	0.175 ± 0.079 (7.5, 1)	0.483 ± 0.057 (2.1, 1)	0.474 ± 0.057 (52, 3.59)	0.506 ± 0.048 (45.6, 1.48)	$\mathbf{0.523 \pm 0.062}$ (112.1, 1.366)	0.521 ± 0.060 (51.2, 1.235)	0.521 ± 0.060 (17.1, 1.142)
HTC	0.581 ± 0.047 (8.8, 1)	0.727 ± 0.05 (3.2, 1.23)	0.724 ± 0.032 (32, 2.05)	$\mathbf{0.750 \pm 0.038}$ (32.9, 1.09)	0.743 ± 0.038 (46.7, 1.06)	0.735 ± 0.058 (23.9, 1)	0.736 ± 0.055 (32, 1.09)
BLD	0.163 ± 0.088 (40.7, 2.26)	0.476 ± 0.057 (2.0, 1)	0.433 ± 0 (63, 1.97)	0.572 ± 0.029 (175.9, 2.13)	0.586 ± 0.029 (229.7, 1.98)	0.587 ± 0.028 (62.8, 1.79)	$\mathbf{0.588 \pm 0.027}$ (19, 1.29)
HTS	0.582 ± 0.040 (9.3, 1)	0.721 ± 0.065 (3.5, 1.07)	0.713 ± 0.055 (25, 2.02)	$\mathbf{0.752 \pm 0.036}$ (24.6, 1.06)	0.747 ± 0.031 (34.7, 1.02)	0.746 ± 0.028 (25, 1.02)	0.747 ± 0.028 (24.4, 1.03)
MK3	0.947 (52, 2.88)	0.802 (1, 3)	**0.972** (93, 1.96)	**0.972** (17, 1.88)	**0.972** (200, 2.07)	**0.972** (93, 1.84)	**0.972** (7, 1.43)
VTE	0.913 ± 0.047 (2.7, 1)	0.935 ± 0.055 (1.3, 1.15)	$\mathbf{0.951 \pm 0.035}$ (9, 1.07)	0.927 ± 0.045 (23.5, 1.17)	0.93 ± 0.042 (39, 1.11)	0.929 ± 0.043 (8.2, 1)	0.934 ± 0.038 (6.4, 1)
BCC	0.254 ± 0.089 (8.1, 1)	0.476 ± 0.086 (1.2, 1)	0.452 ± 0.079 (31, 2.93)	$\mathbf{0.588 \pm 0.057}$ (33.6, 1.17)	0.565 ± 0.059 (39.6, 1.15)	0.563 ± 0.061 (30.2, 1.07)	0.569 ± 0.063 (29.4, 1.17)
MAM	0.668 ± 0.032 (26.4, 2.68)	0.808 ± 0.022 (5.3, 1.43)	$\mathbf{0.816 \pm 0.018}$ (48, 2.53)	0.805 ± 0.028 (38.7, 1.32)	0.796 ± 0.026 (92.2, 1.27)	0.796 ± 0.026 (47.6, 1.24)	0.797 ± 0.024 (40.5, 1.25)
LIV	0.357 ± 0.016 (10, 1)	0.445 ± 0.083 (1.5, 1)	0.563 ± 0.058 (59, 2.35)	0.585 ± 0.071 (43.4, 1.56)	0.594 ± 0.046 (242.5, 1.42)	$\mathbf{0.595 \pm 0.048}$ (58.2, 1.32)	0.588 ± 0.049 (45.7, 1.36)

Table 2: Results on binary REL classification. We report the F1-score along with standard deviation and, in brackets below, the number of the learnt rules as well as the average length of the learnt rules. The proposed REL algorithm, gHKL$_\rho$ ($\rho = 1.5, 1.1$), obtains better generalization performance than state-of-the-art ENDER in most data sets, with the additional advantage of learning a smaller set of more compact rules. The '*' symbol denotes statistically significant improvement. The results are averaged over ten random train-test splits.

Additionally if the best result achieves a statistically significant improvement over its nearest competitor, it is marked with a '*'. Statistical significance test is performed using the paired t-test at 99% confidence. We also report the average number of rules learnt (r) and the average length of the rules (c), specified below each F1-score as: (r, c). As discussed earlier, it is desirable that REL algorithms achieve high F1-score with a compact set of simple rules, i.e., low r and c.

We can observe from Table 2 that $\texttt{gHKL}_\rho$ obtains better generalization performance than state-of-the-art ENDER in most of the data sets with the additional advantage of having rules with smaller number of conjunctions. In fact, when averaged over the data sets, $\texttt{gHKL}_{1.1}$ and $\texttt{gHKL}_{1.5}$ output the shortest rules among all the methods. $\texttt{gHKL}_{1.1}$ obtains statistically significant performance in TIC-TAC-TOE data set. Though the generalization obtained by $\texttt{gHKL}_2$ (HKL), $\texttt{gHKL}_{1.5}$ and $\texttt{gHKL}_{1.1}$ are similar, the number of rules selected by $\texttt{gHKL}_2$ is always higher than $\texttt{gHKL}_{1.1}$ (by as much as 25 times in a few cases), hampering its interpretability.

6.2 Multiclass Classification in REL

This section summarizes our results on multiclass REL classification. The details of the multiclass data sets are provided in Table 3. Within the data sets, classes with too few instances (< 3) were not considered for simulations since we perform a three-fold cross validation for hyper-parameter selection. The results, averaged over 10 random train-test splits with 10% train data are presented in Table 4. Following Dembczyński et al. (2008, 2010), we report the accuracy to compare generalization performance among the algorithms. The number of rules as well as the average length of the rules is also reported to judge the interpretability of the output.

We can observe that $\texttt{gHKL}_{\texttt{MT}-\rho}$ obtains the best generalization performance in seven data sets, out of which four are statistically significant. Moreover, $\texttt{gHKL}_{\texttt{MT}-1.5}$ and $\texttt{gHKL}_{\texttt{MT}-1.1}$ usually select the shortest rules among all the methods. The number of rules as well as the average rule length of $\texttt{gHKL}_{\texttt{MT}-2}$ is generally very large compared to $\texttt{gHKL}_{\texttt{MT}-1.5}$ and $\texttt{gHKL}_{\texttt{MT}-1.1}$. This again demonstrates the suitability of the proposed ℓ_1/ℓ_ρ regularizer in obtaining a compact set of simple rules.

Data set	Num	c	p	$\lvert\mathcal{V}\rvert$	Data set	Num	c	p	$\lvert\mathcal{V}\rvert$
BALANCE	625	3	32	$\approx 10^9$	IRIS	150	3	50	$\approx 10^{15}$
CAR	1728	4	42	$\approx 10^{12}$	LYMPH	146	3	86	$\approx 10^{25}$
C.M.C.	1473	3	54	$\approx 10^{16}$	T.A.E.	151	3	114	$\approx 10^{34}$
ECOLI	332	6	42	$\approx 10^{12}$	YEAST	1484	10	54	$\approx 10^{16}$
GLASS	214	6	72	$\approx 10^{21}$	ZOO	101	7	42	$\approx 10^{12}$

Table 3: Data sets used for multiclass REL classification. 'Num' is the number of instances in the data set while 'c' denotes the number of classes. The number of number of basic propositions is 'p' and $|\mathcal{V}|$ represents the total number of possible conjunctions. For each numerical input feature, 8 basic propositions were derived.

	SLI	**ENDER**	**gHKL$_{\mathbf{MT}-\rho}$**		
			$\rho = 2$	$\rho = 1.5$	$\rho = 1.1$
BALANCE	0.758 ± 0.025 (10.4, 1.7)	0.795 ± 0.034 (112, 2.4)	$\mathbf{0.817 \pm 0.028}$ (2468.9, 2.84)	0.808 ± 0.032 (112, 1.64)	0.807 ± 0.034 (85, 1.61)
CAR	0.823 ± 0.029 (18.3, 2.93)	0.835 ± 0.024 (270, 3.05)	0.864 ± 0.020 (9571.2, 3.14)	0.86 ± 0.028 (220.3, 1.64)	$\mathbf{0.875 \pm 0.029}^*$ (269.3, 1.85)
C.M.C.	0.446 ± 0.016 (21.1, 1.9)	$\mathbf{0.485 \pm 0.015}^*$ (513, 4.36)	0.472 ± 0.014 (10299.3, 2.85)	0.463 ± 0.017 (512.9, 1.95)	0.465 ± 0.016 (396.4, 1.88)
ECOLI	0.726 ± 0.042 (7.8, 1.34)	0.636 ± 0.028 (35, 2.15)	0.779 ± 0.057 (4790.2, 2.99)	$\mathbf{0.784 \pm 0.045}^*$ (34.3, 1.05)	0.778 ± 0.054 (32.4, 1.16)
GLASS	0.43 ± 0.061 (7.4, 1.41)	0.465 ± 0.052 (70, 3.21)	0.501 ± 0.049 (5663.7, 2.40)	$\mathbf{0.525 \pm 0.043}^*$ (69.1, 1.15)	0.524 ± 0.046 (54.6, 1.04)
IRIS	0.766 ± 0.189 (2.2, 1.02)	0.835 ± 0.093 (10, 1.34)	0.913 ± 0.083 (567, 2.44)	$\mathbf{0.927 \pm 0.024}^*$ (9.8, 1)	0.893 ± 0.091 (8.6, 1)
LYMPH	0.61 ± 0.066 (2.7, 1)	0.706 ± 0.058 (34, 2.2)	0.709 ± 0.061 (4683.8, 2.30)	$\mathbf{0.724 \pm 0.078}$ (33.7, 1.01)	0.722 ± 0.078 (33, 1.01)
T.A.E.	0.334 ± 0.035 (1.1, 1)	0.41 ± 0.065 (39, 1.86)	$\mathbf{0.418 \pm 0.049}$ (5707.4, 2.25)	0.399 ± 0.049 (38.3, 1.00)	0.402 ± 0.046 (38.1, 1.05)
YEAST	0.478 ± 0.035 (23.4, 1.63)	$\mathbf{0.497 \pm 0.015}$ (218, 5.78)	0.487 ± 0.021 (8153.6, 2.85)	0.485 ± 0.022 (217.8, 1.80)	0.486 ± 0.021 (179.6, 1.73)
ZOO	0.556 ± 0.062 (7.1, 1.24)	$\mathbf{0.938 \pm 0.033}$ (33, 1.29)	0.877 ± 0.06 (3322.2, 2.70)	0.928 ± 0.037 (32.3, 1.00)	0.927 ± 0.039 (31.9, 1.01)

Table 4: Results on multiclass REL classification. We report the accuracy along with standard deviation and, in the brackets below, the number of learnt rules as well as the average length of the learnt rules. The proposed REL algorithm, gHKL$_{\mathtt{MT}-\rho}$, obtains the best generalization performance in most data sets. In addition, for $\rho = 1.5$ and 1.1, our algorithm learns a smaller set of more compact rules than state-of-the-art ENDER. The '*' symbol denotes statistically significant improvement. The results are averaged over ten random train-test splits.

7. Summary

This paper generalizes the HKL framework in two ways. First, a generic ℓ_1/ℓ_ρ block-norm regularizer, $\rho \in (1, 2)$, is employed that provides a more flexible kernel selection pattern than HKL by mitigating the weight bias towards the kernels that are nearer to the sources of the DAG. Secondly, the framework is further generalized to the setup of learning a shared feature representation among multiple related tasks. We pose the problem of learning shared features across the tasks as that of learning a shared kernel. An efficient mirror descent based active set algorithm is proposed to solve the generalized formulations (gHKL/gHKL$_{\mathrm{MT}}$). An interesting computational result is that gHKL/gHKL$_{\mathrm{MT}}$ can be solved in time polynomial in the active set and training set sizes whenever the HKL formulation can be solved in polynomial time. The other important contribution in this paper is the application of the proposed gHKL/gHKL$_{\mathrm{MT}}$ formulations in the setting of Rule Ensemble Learning (REL),

where HKL has not been previously explored. We pose the problem of learning an ensemble of propositional rules as a kernel learning problem. Empirical results on binary as well as multiclass classification for REL demonstrate the effectiveness of the proposed generalizations.

Acknowledgments

We thank the anonymous reviewers for the valuable comments. We acknowledge Chiranjib Bhattacharyya for initiating discussions on optimal learning of rule ensembles. Pratik Jawanpuria acknowledges support from IBM Ph.D. fellowship.

Appendix A.

In the appendix section, we provide the proofs of theorems/lemmas referred to in the main paper.

A.1 Lemma 26 of Micchelli and Pontil (2005)

Let $a_i \geq 0, i = 1, \ldots, d$, $1 \leq r < \infty$ and $\Delta_{d,r} = \left\{ \mathbf{z} \in \mathbb{R}^d \mid \mathbf{z} \geq 0, \sum_{i=1}^d \mathbf{z}_i^r \leq 1 \right\}$. Then, the following result holds:

$$\min_{\mathbf{z} \in \Delta_{d,r}} \sum_{i=1}^d \frac{a_i}{\mathbf{z}_i} = \left(\sum_{i=1}^d a_i^{\frac{r}{r+1}} \right)^{1+\frac{1}{r}}.$$

The minimum is attained at

$$\mathbf{z}_i = \frac{a_i^{\frac{1}{r+1}}}{\left(\sum_{j=1}^d a_j^{\frac{r}{r+1}} \right)^{\frac{1}{r}}} \quad \forall i = 1, \ldots, d.$$

The proof follows from Holder's inequality.

A.2 Proof of Lemma 1

Proof Applying the above lemma (Appendix A.1) on the outermost ℓ_1-norm of the regularizer $\Omega_T(f_1, \ldots, f_T)^2$ in (6), we get

$$\Omega_T(f_1, \ldots, f_T)^2 = \min_{\gamma \in \Delta_1} \sum_{v \in \mathcal{V}} \frac{d_v^2}{\gamma_v} \left(\sum_{w \in D(v)} (Q_w(f_1, \ldots, f_T))^\rho \right)^{\frac{2}{\rho}},$$

where $\Delta_1 = \left\{ \mathbf{z} \in \mathbb{R}^{|\mathcal{V}|} \mid \mathbf{z} \geq 0, \sum_{v \in \mathcal{V}} \mathbf{z}_v \leq 1 \right\}$. Reapplying the above lemma on the individual terms of the above summation gives

$$\left(\sum_{w \in D(v)} (Q_w(f_1, \ldots, f_T)^2)^{\frac{\rho}{2}} \right)^{\frac{2}{\rho}} = \min_{\lambda_v \in \Delta_\rho^v} \sum_{w \in D(v)} \frac{Q_w(f_1, \ldots, f_T)^2}{\lambda_{vw}},$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$ and $\Delta_r^v = \left\{ \mathbf{z} \in \mathbb{R}^{|D(v)|} \mid \mathbf{z} \geq 0, \sum_{w \in D(v)} \mathbf{z}_w^r \leq 1 \right\}$. Using the above two results and regrouping the terms will complete the proof. ■

A.3 Re-parameterization of the Multi-task Regularizer in (8)

The gHKL$_{\text{MT}}$ dual formulation (10) follows from the representer theorem (Schölkopf and Smola, 2002) after employing the following re-parameterization in (8).

Define $f^{0w} = \frac{1}{T+\mu} \sum_{t=1}^{T} f_{tw}$ and $f^{tw} = f_{tw} - f^{0w}$. Then, $Q_w(f_1, \ldots, f_T)$ in (8) may be rewritten as:

$$Q_w(f_1, \ldots, f_T) = \left(\mu \|f^{0w}\|^2 + \sum_{t=1}^{T} \|f^{tw}\|^2 \right)^{\frac{1}{2}}.$$

Further, construct the following feature map (Evgeniou and Pontil, 2004)

$$\Phi_w(\mathbf{x}, t) = \Big(\frac{\phi_w(\mathbf{x})}{\sqrt{\mu}}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{\text{for tasks before t}}, \phi_w(\mathbf{x}), \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{\text{for tasks after t}} \Big) \tag{16}$$

and define $f_w = (\sqrt{\mu} f^{0w}, f^{1w}, \ldots, f^{Tw})$.

With the above definitions, we rewrite the gHKL$_{\text{MT}}$ primal regularizer as well as the prediction function: $Q_w(f_1, \ldots, f_T)^2 = \|f_w\|^2$ and $F_t(\mathbf{x}) = \sum_{w \in \mathcal{V}} \langle f_w, \Phi_w(\mathbf{x}, t) \rangle - b_t \ \forall\ t$. It follows from Lemma 1 that the gHKL$_{\text{MT}}$ primal problem based on (8) is equivalent to the following optimization problem:

$$\min_{\gamma \in \Delta_1} \ \min_{\lambda_v \in \Delta_\rho^v \ \forall v \in \mathcal{V}} \ \min_{f, b} \ \frac{1}{2} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda)^{-1} \|f_w\|^2 + C \sum_{t=1}^{T} \sum_{i=1}^{m} \ell(y_{ti}, F_t(\mathbf{x}_{ti})), \tag{17}$$

where $f = (f_w)_{w \in \mathcal{V}}$ and $b = [b_1, \ldots, b_T]$.

A.4 Motivation for the Active Set Algorithm

Lemma 8 *The problem (12) remains the same whether solved with the original set of variables (η) or when solved with only those $\eta_v \neq 0$ at optimality.*

Proof The above follows from the following reasoning: a) variables η owe their presence in (12) only via $\zeta(\eta)$ functions, b) $(\eta_v = 0) \Rightarrow (\zeta_w(\eta) = 0 \ \forall w \in D(v))$, c) Let (η', α') be an optimal solution of the problem (12). If $\zeta_v(\eta') = 0$ and $\eta'_v \neq 0$, then (η^*, α') is also an optimal solution of the problem (12) where $\eta^*_w = \eta'_w \ \forall w \in \mathcal{V} \setminus v$ and $\eta^*_v = 0$, and d) min-max interchange in (12) yields an equivalent formulation. ■

Lemma 9 *The following min-max interchange is equivalent:*

$$\min_{\eta \in \Delta_1} \ \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \bar{G}(\eta, \alpha) = \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \ \min_{\eta \in \Delta_1} \bar{G}(\eta, \alpha),$$

where

$$\bar{G}(\eta,\alpha)=\mathbf{1}^\top\alpha-\frac{1}{2}\left(\sum_{w\in\mathcal{V}}\zeta_w(\eta)\left(\alpha^\top\mathbf{Y}H_w\mathbf{Y}\alpha\right)^{\bar{\rho}}\right)^{\frac{1}{\bar{\rho}}}.$$

Proof Note that $G(\eta,\alpha)$ is a convex function in η and a concave function in α. The min-max interchange follows from Sion-Kakutani minimax theorem (Sion, 1958). ■

A.5 Proof of Theorem 3

Before stating the proof of Theorem 3, we first prove the results in Lemma 10, Proposition 11 and Lemma 12, which will be employed therein (also see Bach, 2009, Lemma 10 and Proposition 11).

Lemma 10 *Let $a_i>0\ \forall i=1,\ldots,d$, $1<r<\infty$ and $\Delta_1=\left\{\mathbf{z}\in\mathbb{R}^d\mid\mathbf{z}\geq 0,\sum_{i=1}^d\mathbf{z}_i\leq 1\right\}$. Then, the following holds true:*

$$\min_{\mathbf{z}\in\Delta_1}\ \sum_{i=1}^d a_i\mathbf{z}_i^r=\left(\sum_{i=1}^d a_i^{\frac{1}{1-r}}\right)^{1-r}$$

and the minimum is attained at

$$\mathbf{z}_i=a_i^{\frac{1}{1-r}}\left(\sum_{j=1}^d a_i^{\frac{1}{1-r}}\right)^{-1}\quad\forall\ i=1,\ldots,d.$$

Proof Take vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ as those with entries $a_i^{\frac{1}{r}}\mathbf{z}_i$ and $a_i^{-\frac{1}{r}}\ \forall\ i=1,\ldots,d$ respectively. The result follows from the Holder's inequality: $\mathbf{u}_1^\top\mathbf{u}_2\leq\|\mathbf{u}_1\|_r\|\mathbf{u}_2\|_{\frac{r}{r-1}}$. Note that if any $a_i=0$, then the optimal value of the above optimization problem is zero. ■

Proposition 11 *The following convex optimization problems are dual to each other and there is no duality gap:*

$$\max_{\gamma\in\Delta_1}\ \sum_{w\in\mathcal{V}}\delta_w(\gamma,\lambda)M_w,\tag{18}$$

$$\min_{\kappa\in L}\ \max_{u\in\mathcal{V}}\ \sum_{w\in D(u)}\frac{\kappa_{uw}^2\lambda_{uw}M_w}{d_u^2},\tag{19}$$

where $L=\{\kappa\in\mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}\mid\kappa\geq 0,\ \sum_{v\in A(w)}\kappa_{vw}=1,\ \kappa_{vw}=0\ \forall v\in A(w)^c,\ \forall w\in\mathcal{V}\}$, $\Delta_1=\left\{\mathbf{z}\in\mathbb{R}^{|\mathcal{V}|}\mid\mathbf{z}\geq 0,\sum_{v\in\mathcal{V}}\mathbf{z}_v\leq 1\right\}$ and $M_w\geq 0\ \forall w\in\mathcal{V}$.

Proof The optimization problem (19) may be equivalently rewritten as:

$$\min_{\kappa\in L}\ \min_{A}A,\quad\text{subject to}\quad A\geq\sum_{w\in D(u)}\frac{\kappa_{uw}^2\lambda_{uw}M_w}{d_u^2}\ \forall u\in\mathcal{V},$$

$$= \min_{\kappa \in L} \max_{\gamma \in \Delta_1} \sum_{u \in \mathcal{V}} \sum_{w \in D(u)} \frac{\gamma_u \kappa_{uw}^2 \lambda_{uw} M_w}{d_u^2} \qquad \text{(Lagrangian dual with respect to } A\text{)}$$

$$= \max_{\gamma \in \Delta_1} \min_{\kappa \in L} \sum_{w \in \mathcal{V}} \left(\sum_{u \in A(w)} \kappa_{uw}^2 \frac{\gamma_u \lambda_{uw}}{d_u^2} \right) M_w \qquad \text{(min-max interchange and rearranging terms)}$$

$$= \max_{\gamma \in \Delta_1} \sum_{w \in \mathcal{V}} \left(\sum_{u \in A(w)} \left(\frac{\gamma_u \lambda_{uw}}{d_u^2} \right)^{-1} \right)^{-1} M_w \qquad \text{(Lemma 10 with respect to variables } \kappa\text{)}$$

$$= \max_{\gamma \in \Delta_1} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) M_w \qquad \blacksquare$$

Lemma 12 *The following min-max interchange is equivalent:*

$$\min_{\gamma \in \Delta_1} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} G(\gamma, \lambda, \alpha) = \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} \min_{\gamma \in \Delta_1} \min_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} G(\gamma, \lambda, \alpha),$$

where $G(\gamma, \lambda, \alpha)$ is as defined in (10).

Proof We proceed by applying a change of variables. Note that $\gamma_v = 0$ implies that the variables λ_{vw} ($\forall w \in D(v)$) do not influence the objective of optimization problem (10). This follows from the definition of the $\delta(\gamma, \lambda)$ function. Hence, we define $\beta_{vw} = \gamma_v \lambda_{vw}$, $\forall w \in D(v)$ as it is a one-to-one transformation for $\gamma_v \neq 0$ (see also Szafranski et al., 2010). The gHKL dual (10) (the L.H.S. of the proposed lemma) can be equivalently rewritten as:

$$\min_{\substack{\beta_{vw} \geq 0 \ \forall w \in D(v), v \in \mathcal{V} \\ \sum_v \|\beta_{vD(v)}\|_{\hat{\rho}} \leq 1}} \max_{\alpha_t \in S(\mathbf{y}_t, C) \forall t} G(\beta, \alpha), \quad \text{where} \quad \beta_{vD(v)} = (\beta_{vw})_{w \in D(v)},$$

$$G(\beta, \alpha) = \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \left(\sum_{w \in \mathcal{V}} \delta_w(\beta) H_w \right) \mathbf{Y} \alpha, \ \text{ and } \ \delta_w(\beta)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\beta_{vw}}.$$

Note that $\delta_w(\beta)$ is a concave function of β (in the given feasibility set) and hence $G(\beta, \alpha)$ is convex-concave function with convex and compact feasibility sets. Therefore, we obtain $\min_\beta \max_\alpha G(\beta, \alpha) = \max_\alpha \min_\beta G(\beta, \alpha)$ (with constraints over β and α as stated above) by applying the Sion-Kakutani minimax theorem (Sion, 1958). Finally, we revert to the original variables (γ, λ) by substituting $\gamma_v = (\sum_{w \in D(v)} (\beta_{vw})^{\hat{\rho}})^{\frac{1}{\hat{\rho}}}$ $\forall v \in \mathcal{V}$ and $\lambda_{vw} = \frac{\beta_{vw}}{\gamma_v}$ $\forall w \in D(v)$, $\forall v \in \mathcal{V}$ s.t. $\gamma_v \neq 0$. This gives us the equivalent R.H.S. $\blacksquare$

Now we begin the proof of Theorem 3.

Proof From Lemma 12, the gHKL dual (10) can be equivalently written as:

$$\max_{\alpha \in S(\mathbf{y}, C)} \mathbf{1}^\top \alpha - \frac{1}{2} \underbrace{\max_{\gamma \in \Delta_1} \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \left(\sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) \alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)}_{\mathcal{O}}, \tag{20}$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$. In the following, we equivalently rewrite the second part of the above formulation.

$$
\begin{aligned}
\mathcal{O} &= \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ \max_{\gamma \in \Delta_1} \sum_{w \in \mathcal{V}} \delta_w(\gamma, \lambda) \underbrace{\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha}_{M_w} \\
&= \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ \min_{\kappa \in L} \ \max_{u \in \mathcal{V}} \sum_{w \in D(u)} \frac{\kappa_{uw}^2 \lambda_{uw} M_w}{d_u^2} && \text{(Proposition 11)} \\
&= \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ \min_{\kappa \in L} \ \min_{A} \ A && \text{(Eliminating } u\text{)} \\
&\qquad \text{s.t. } A \geq \sum_{w \in D(v)} \frac{\kappa_{vw}^2 \lambda_{vw} M_w}{d_v^2} \ \forall v \in \mathcal{V} \\
&= \min_{\kappa \in L} \ \min_{A} \ \max_{\lambda_v \in \Delta_{\hat{\rho}}^v \ \forall v \in \mathcal{V}} \ A && \text{(Sion-Kakutani theorem)} \\
&\qquad \text{s.t. } A \geq \sum_{w \in D(v)} \frac{\kappa_{vw}^2 \lambda_{vw} M_w}{d_v^2} \ \forall v \in \mathcal{V} \\
&= \min_{\kappa \in L} \ \min_{A} \ A && \text{(Holder's inequality, } \bar{\rho} = \tfrac{\hat{\rho}}{\hat{\rho}-1}\text{)} \\
&\qquad \text{s.t. } A \geq d_v^{-2} \left(\sum_{w \in D(v)} \left(\kappa_{vw}^2 M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \ \forall v \in \mathcal{V} \\
&= \min_{\kappa \in L} \ \max_{u \in \mathcal{V}} \ d_u^{-2} \left(\sum_{w \in D(u)} \left(\kappa_{uw}^2 M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} && \text{(Eliminating A)} \qquad (21)
\end{aligned}
$$

Now consider the problem $\mathcal{O}^{\bar{\rho}} = \min_{\kappa \in L} \max_{u \in \mathcal{V}} d_u^{-2\bar{\rho}} \sum_{w \in D(u)} \left(\kappa_{uw}^2 M_w \right)^{\bar{\rho}}$. Its Lagrangian is

$$
\mathcal{L}(\kappa, A, \eta) = A + \sum_{v \in \mathcal{V}} \eta_v \left(d_v^{-2\bar{\rho}} \sum_{w \in D(v)} \left(\kappa_{vw}^2 M_w \right)^{\bar{\rho}} - A \right).
$$

Minimization of $\mathcal{L}$ with respect to A leads to the constraint $\eta \in \Delta_1$. Hence, we have:

$$
\mathcal{O}^{\bar{\rho}} = \max_{\eta \in \Delta_1} \ \min_{\kappa \in L} \ \sum_{v \in \mathcal{V}} \sum_{w \in D(v)} \eta_v \left(d_v^{-2} \kappa_{vw}^2 M_w \right)^{\bar{\rho}}.
$$

Using the special structure of L, the above can be rewritten as:

$$
\mathcal{O}^{\bar{\rho}} = \max_{\eta \in \Delta_1} \ \sum_{w \in \mathcal{V}} (M_w)^{\bar{\rho}} \left(\min_{\kappa_w \in \Delta_{|A(w)|}} \sum_{v \in A(w)} \left(\eta_v d_v^{-2\bar{\rho}} \right) \kappa_{vw}^{2\bar{\rho}} \right),
$$

where $\Delta_{|A(w)|} = \left\{ \eta \in \mathbb{R}^{|A(w)|} \mid \eta \geq 0, \sum_{w \in A(w)} \eta_w \leq 1 \right\}$. By applying Lemma 10 with respect to variables κ, we obtain the following equivalence:

$$
\min_{\kappa_w \in \Delta_{|A(w)|}} \sum_{v \in A(w)} \left(\eta_v d_v^{-2\bar{\rho}} \right) \kappa_{vw}^{2\bar{\rho}} = \zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^{\rho} \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}. \qquad (22)
$$

From the above two results, we obtain the following equivalent dual of (21):

$$\mathcal{O} = \max_{\eta \in \Delta_1} \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}. \tag{23}$$

Substituting $\mathcal{O}$ in (20) by the above (23) and again interchanging the min-max completes the proof. ■

A.6 Proof of Theorem 4

Proof We begin by noting that $\zeta_v(\eta)$ $(v \in \mathcal{V})$ is a concave function of η for all v (this is because when $\rho \in (1, 2]$, ζ_v is a weighted q-norm in η, where $q \in [-1, 0)$ and hence is concave in the first quadrant). By simple observations regarding operations preserving convexity we have that the objective in (13) is a convex function of η for a fixed value of α. Hence $g(\eta)$, which is a point-wise maximum over convex functions, is itself convex. The expression for $\nabla g(\eta)$ is computed by employing Danskin's theorem (Bertsekas, 1999, Proposition B.25) and is as follows:

$$(\nabla g(\eta))_i = -\frac{(1-\varepsilon)}{2\bar{\rho}} \times \overbrace{\left(\sum_{u \in D(i)} d_i^\rho \left((1-\varepsilon)\eta_i + \frac{\varepsilon}{|\mathcal{V}|} \right)^{-\rho} \zeta_u^s(\eta)^\rho \left(\bar{\alpha}^\top \mathbf{Y} H_u \mathbf{Y} \bar{\alpha} \right)^{\bar{\rho}} \right)}^{P_1} \tag{24}$$
$$\times \underbrace{\left(\sum_{w \in \mathcal{V}} \zeta_w^s(\eta) \left(\bar{\alpha}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}-1}}_{P_2},$$

where $\bar{\rho} = \frac{\rho}{2(\rho-1)}$, $\zeta_w^s(\eta) = \zeta_w((1-\varepsilon)\eta + \frac{\varepsilon}{|\mathcal{V}|})$, i.e., the smoothed $\zeta_w(\eta)$ and $\bar{\alpha}$ is an optimal solution of problem (13) with that η where the gradient is to be computed.

Next, we show that g is Lipschitz continuous by showing that its gradient is bounded. Firstly, $\rho \in (1, 2]$ and hence $\bar{\rho} \in [1, \infty)$. Next, let the minimum and maximum eigenvalues over all H_w $(w \in \mathcal{V})$ be θ and σ respectively. Then we have $\theta \|\bar{\alpha}\|^2 \leq \bar{\alpha}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha} \leq \sigma \|\bar{\alpha}\|^2$. Using this, we obtain: $\sum_{w \in \mathcal{V}} \zeta_w^s(\eta) \left(\bar{\alpha}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha} \right)^{\bar{\rho}} \geq \theta^{\bar{\rho}} \|\bar{\alpha}\|^{2\bar{\rho}} \sum_{w \in \mathcal{V}} \zeta_w^s(\eta)$. Note that $\sum_{w \in \mathcal{V}} \zeta_w^s(\eta) \geq \zeta_r^s(\eta)$ where $r \in sources(\mathcal{V})$ and $\zeta_r^s(\eta) \geq d_{max}^{\rho/(1-\rho)} \frac{\varepsilon}{|\mathcal{V}|}$ where d_{max} is the maximum of d_v $(v \in \mathcal{V})$. Thus we obtain: $P_2 \leq \left(\theta^{\bar{\rho}} \|\bar{\alpha}\|^{2\bar{\rho}} \varepsilon / |\mathcal{V}| \right)^{\frac{1}{\bar{\rho}}-1} d_{max}^{\frac{2-\rho}{\rho-1}}$.

Now, it is easy to see that $\forall u \in D(i)$, $d_i^\rho ((1-\varepsilon)\eta_i + \frac{\varepsilon}{|\mathcal{V}|})^{-\rho} \zeta_u(\eta)^\rho \leq d_i^{\frac{\rho}{1-\rho}} \leq d_{min}^{\frac{\rho}{1-\rho}}$, where d_{min} is the minimum of d_v $(v \in \mathcal{V})$. Hence $P_1 \leq |\mathcal{V}| \sigma^{\bar{\rho}} \|\bar{\alpha}\|^{2\bar{\rho}} d_{min}^{\frac{\rho}{1-\rho}}$. In addition, since $0 \leq \bar{\alpha} \leq C$, we have $\|\bar{\alpha}\| \leq \sqrt{mT}C$. Summarizing these findings, we obtain the following bound on the gradient:

$$\|\nabla g(\eta)\|_1 \leq \frac{(1-\varepsilon)}{2\bar{\rho}} mTC^2 \theta^{1-\bar{\rho}} \sigma^{\bar{\rho}} \varepsilon^{\frac{1-\bar{\rho}}{\bar{\rho}}} |\mathcal{V}|^{\frac{2}{\rho}+1} d_{min}^{\frac{\rho}{1-\rho}} d_{max}^{\frac{2-\rho}{\rho-1}}.$$

The proof will be similar for gHKL$_{\text{MT}}$ formulations in other learning settings. ■

A.7 Proof of Theorem 5

Proof Given a candidate solution η and $\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$ (with associated primal ($\mathbf{f} = (f_1, \ldots, f_T), b, \xi$)), the duality gap ($D$) between the two variational formulations in Lemma 9 is as follows:

$$\begin{aligned} D &= \max_{\hat{\alpha}_t \in S(\mathbf{y}_t, C) \forall t} \bar{G}(\eta, \hat{\alpha}) \; - \min_{\hat{\eta} \in \Delta_1} \; \bar{G}(\hat{\eta}, \alpha) \\ &\leq \frac{1}{2}\Omega_T(\mathbf{f})^2 + C\mathbf{1}^\top \xi \; - \min_{\hat{\eta} \in \Delta_1} \; \bar{G}(\hat{\eta}, \alpha) \\ &= \overbrace{\Omega_T(\mathbf{f})^2 + C\mathbf{1}^\top \xi - \mathbf{1}^\top \alpha}^{\text{Gap in solving with fixed } \eta} + \underbrace{\frac{1}{2}\left(\max_{\hat{\eta} \in \Delta_1} \; \left(\sum_{w \in \mathcal{V}} \zeta_w(\hat{\eta}) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} - \Omega_T(\mathbf{f})^2 \right)}_{\text{Gap in solving with fixed } \alpha}. \end{aligned}$$

With this upper bound on the duality gap, it is easy to see that the following condition is sufficient for the reduced solution (with active set $\mathcal{W}$) to have $D \leq \epsilon$:

$$\max_{\eta \in \Delta_1} \; \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(\alpha^\top \mathbf{Y} H_w \mathbf{Y} \alpha \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq \Omega_T(\mathbf{f}_{\mathcal{W}})^2 + 2(\epsilon - \epsilon_{\mathcal{W}}), \tag{25}$$

where $\epsilon_{\mathcal{W}}$ is the duality gap[9] associated with the computation of the dual variables $\alpha_{\mathcal{W}}$. Here as well as in the rest of the proof, the subscript $(\cdot)_{\mathcal{W}}$ implies the value of the variable obtained when the gHKL$_{\text{MT}}$ formulation is solved with $\mathcal{V}$ restricted to the active set $\mathcal{W}$. In Appendix A.5, we had proved that the L.H.S. of the above inequality is equal to the R.H.S. of (21), i.e.,

$$\max_{\eta \in \Delta_1} \; \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) \left(M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} = \min_{\kappa \in L} \; \max_{v \in \mathcal{V}} \; d_v^{-2} \left(\sum_{w \in D(v)} \left(\kappa_{vw}^2 M_w \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}, \tag{26}$$

where $M_w = \alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}$.

Next, we obtain an upper bound of the above by substituting $\kappa \in L$ in the R.H.S of (26). In particular, we employ the following: the value of κ_{vw} $v, w \in \mathcal{W}$ is obtained by solving the small[10] problem (14). This is fine because $\mathcal{W} = hull(\mathcal{W})$. For $v \in \mathcal{W}^c$ and $w \in \mathcal{W}$, by definition of L and $\mathcal{W}$, we have $\kappa_{vw} = 0$. Next, κ_{vw} is set to zero $\forall$ $v \in \mathcal{W}, w \in \mathcal{W}^c$. For the remaining κ_{vw}, $v \in \mathcal{W}^c$ and $w \in \mathcal{W}^c$, we use the value of κ obtained by solving (21) with $\rho = 1$, i.e., $\kappa_{vw} = d_v \left(\sum_{u \in A(v) \cap \mathcal{W}^c} d_u \right)^{-1}$ (also see Section A.5 Bach, 2009). Note that the above constructed value of κ is feasible in the set L. With this choice of κ substituted in

9. This is given by the gap associated with the $\hat{\rho}$-norm MKL solver employed in the mirror descent algorithm for solving the small problem (14).
10. The value of κ_{vw} ($\forall v, w \in \mathcal{W}$) obtained in this manner satisfy the constraint set L restricted to $\mathcal{W}$, i.e., $L_{\mathcal{W}} = \{\kappa \in \mathbb{R}^{|\mathcal{W}| \times |\mathcal{W}|} \mid \kappa \geq 0, \; \sum_{v \in A(w)} \kappa_{vw} = 1, \; \kappa_{vw} = 0 \; \forall \; v \in A(w)^c \cap \mathcal{W}, \; \forall \; w \in \mathcal{W}\}$

the R.H.S. of (26), we have the following inequalities:

$$\begin{aligned}
&\max_{\eta\in\Delta_1} \left(\sum_{w\in\mathcal{V}} \zeta_w(\eta)\left(\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}} \\
&\quad \leq \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in\mathcal{W}^c}\left(\sum_{w\in D(u)}\left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap\mathcal{W}^c} d_v\right)^2}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}}\right\} \\
&\qquad (\text{ Specific choice of } \kappa) \\
&\quad = \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in sources(\mathcal{W}^c)}\left(\sum_{w\in D(u)}\left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap\mathcal{W}^c} d_v\right)^2}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}}\right\} \\
&\qquad (\because\ \mathcal{W} = hull(\mathcal{W})) \\
&\quad \leq \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in sources(\mathcal{W}^c)}\left(\sum_{w\in D(u)}\left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap D(u)} d_v\right)^2}\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}}\right\} \\
&\qquad (\because\ \textstyle\sum_{v\in A(w)\cap\mathcal{W}^c} d_v \geq \sum_{v\in A(w)\cap D(u)} d_v) \\
&\quad \leq \max\left\{\Omega_T(\mathbf{f}_{\mathcal{W}})^2, \max_{u\in sources(\mathcal{W}^c)}\sum_{w\in D(u)}\frac{\alpha_{\mathcal{W}}^\top \mathbf{Y} H_w \mathbf{Y} \alpha_{\mathcal{W}}}{\left(\sum_{v\in A(w)\cap D(u)} d_v\right)^2}\right\} \\
&\qquad (\because\ \|\beta\|_1 \geq \|\beta\|_{\bar\rho}\ \forall\ \bar\rho \geq 1)
\end{aligned}$$

Employing the above upper bound in (25) leads to the result in Theorem 5. Note that in practice, the last upper bound is not loose for Rule Ensemble Learning (REL) application. This is because most of the matrices, especially near the bottom of the lattice, will be (near) zero-matrices — larger the conjunctive rule, the fewer are the examples which may satisfy it. ■

A.8 gHKL$_{\mathrm{MT}}$ with General Convex Loss Functions

In this section, we present extension of the proposed algorithm to other learning settings like regression. In particular, we consider the case where the loss function $\ell(\cdot,\cdot)$ is a general convex loss function such as the hinge loss, the square loss, the Huber loss, etc.

The gHKL$_{\mathrm{MT}}$ primal formulation with a general convex loss function $\ell(\cdot,\cdot)$ was given in equation (6). The specialized gHKL$_{\mathrm{MT}}$ dual formulation corresponding to (6) is as follows:

$$\min_{\eta\in\Delta_1}\ \max_{\alpha_t\in\mathbb{R}^m, \mathbf{1}^\top\alpha_t=0\ \forall t}\ -C\sum_{t=1}^{T}\sum_{i=1}^{m}\varphi_{ti}^*\left(-\frac{\alpha_{ti}}{C}\right) - \frac{1}{2}\left(\sum_{w\in\mathcal{V}}\zeta_w(\eta)\left(\alpha^\top H_w \alpha\right)^{\bar\rho}\right)^{\frac{1}{\bar\rho}},$$

where $\alpha = [\alpha_1^\top, \ldots, \alpha_T^\top]^\top$, $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho}\right)^{\frac{1}{1-\rho}}$ (refer Theorem 3 for details) and φ_{ti}^* denotes the Fenchel[11] conjugate (Boyd and Vandenberghe, 2004) of the function $\varphi_{ti} : z \to \ell(y_{ti}, z)$.

A.9 Prediction Function for gHKL$_{\mathbf{MT}}$ with the Hinge Loss Function

Let the final active set be $\mathcal{W}$ and $(\bar{\eta}_\mathcal{W}, \bar{\alpha}_\mathcal{W})$ be the optimal solution of (12). Then the prediction function for an instance $\mathbf{x}_{tj}$ belonging to the t^{th} task is given by

$$F_t(\mathbf{x}) = (\bar{\alpha}_\mathcal{W} \odot \mathbf{y})^\top \left(\sum_{w \in \mathcal{W}} \bar{\theta}_w (\zeta_w(\bar{\eta}_\mathcal{W}))^{\frac{1}{\bar{\rho}}} H_w(\cdot, \mathbf{x}_{tj}) \right), \tag{27}$$

where symbol $\odot$ denote element-wise product, H_w is the kernel matrix corresponding to the multi-task kernel (11), $H_w(\cdot, \mathbf{x}_{tj}) = ((H_w(\mathbf{x}_{t'i}, \mathbf{x}_{tj}))_{i=1}^m)_{t'=1}^T$ and

$$\bar{\theta}_w = \left(\frac{(\zeta_w(\bar{\eta}_\mathcal{W}))^{\frac{1}{\bar{\rho}}} \bar{\alpha}_\mathcal{W}^\top \mathbf{Y} H_w \mathbf{Y} \bar{\alpha}_\mathcal{W}}{\left(\sum_{v \in \mathcal{W}} \left((\zeta_v(\bar{\eta}_\mathcal{W}))^{\frac{1}{\bar{\rho}}} \bar{\alpha}_\mathcal{W}^\top \mathbf{Y} H_v \mathbf{Y} \bar{\alpha}_\mathcal{W} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}} \right)^{\frac{1}{\bar{\rho}-1}}.$$

A.10 Proof of Corollary 6

Note that proving the computational complexity of the matrix $\mathcal{K}_u$ ($u \in \mathit{sources}(\mathcal{W}^c)$) in (15) to be polynomial time in size of the active set and the training set dimensions suffices to prove the corollary. This is because all the other steps in Algorithms 3 and 2 are of polynomial time complexity (discussed in Section 4).

We begin the proof by introducing some indexing notations related to the multi-task matrices. Let the entries in H_w, the $mT \times mT$ multi-task kernel matrix, be arranged in the following form: the entry corresponding to the input pair $(\mathbf{x}_{t_1 i}, \mathbf{x}_{t_2 j})$ be in the $((t_1 - 1) * m + i)^{th}$ row and $((t_2 - 1) * m + j)^{th}$ column of H_w.

Next we observe that the expression for $\mathcal{K}_u$ in Theorem 5 may be rewritten as:

$$\mathcal{K}_u = \underbrace{\left(\sum_{w \in D(u)} \frac{K_w}{\left(\sum_{v \in A(w) \cap D(u)} d_v \right)^2} \right)}_{T_u} \odot K_T,$$

where: i) K_w is a $mT \times mT$ matrix corresponding to the base kernel k_w and constructed from the inputs from all the tasks, ii) K_T is a $mT \times mT$ such that the entry corresponding to the $((t_1 - 1) * m + i)^{th}$ row and $((t_2 - 1) * m + j)^{th}$ column ($1 \leq i, j \leq m$) of K_T is $B(t_1, t_2)$, and iii) $\odot$ is the symbol for element-wise product (Hadamard product).

11. Fenchel conjugate $\varphi^*(z)$ of a convex function $\varphi(u)$ is given by $\varphi^*(z) = \sup_u z^\top u - \varphi(u)$. As an example, for hinge loss $\varphi(u) = \ell(u, y) = \max(0, 1 - uy)$, $\varphi^*(z) = \begin{cases} zy & \text{if } zy \in [-1, 0] \\ \infty & \text{otherwise} \end{cases}$

In the above expression, $\mathcal{K}_u$ is computable in polynomial time if and only if T_u is computable in polynomial time. The proof of the corollary follows from observing the expression of the sufficient condition for optimality of the HKL (Bach, 2009, Equation 21), which also involves the term T_u.

A.11 Proof of Theorem 7

Given an active set $\mathcal{W}$ of size W, proving that the computational complexity of the verification of the sufficient condition of optimality (15) is polynomial in terms of the active set and the training set sizes suffices to prove Theorem 7. This is because all the other steps in Algorithms 3 and 2 are of polynomial time complexity (discussed in Section 4).

In the REL setup, the DAG is the conjunction lattice and the embedded kernels k_v $v \in \mathcal{V}$ may be rewritten as:

$$k_v(\mathbf{x}_i, \mathbf{x}_j) = \phi_v(\mathbf{x}_i) \cdot \phi_v(\mathbf{x}_j) = \left(\prod_{c \in S_v} \phi_c(\mathbf{x}_i) \right) \cdot \left(\prod_{c \in S_v} \phi_c(\mathbf{x}_j) \right) = \bigodot_{c \in S_v} k_c(\mathbf{x}_i, \mathbf{x}_j),$$

where S_v is the set of basic propositions involved in the conjunction ϕ_v and $\odot$ is the symbol for element-wise product (Hadamard product). The kernels corresponding to the basic propositions are in fact the base kernels embedded in the second level nodes of the lattice $\mathcal{V}$. Employing the above definition of $k_v(\mathbf{x}_i, \mathbf{x}_j)$, the matrices $\mathcal{K}_u$ (in L.H.S. of (15)) are computed as:

$$\mathcal{K}_u = \sum_{w \in D(u)} \frac{K_w}{\left(\sum_{v \in A(w) \cap D(u)} d_v \right)^2} = \left(\bigodot_{c \in S_u} \frac{K_c}{a^2} \right) \odot \left(\bigodot_{c \in B/S_u} \left(\frac{K_c}{(1+a)^2} + \mathbf{1}\mathbf{1}^\top \right) \right),$$

where K_c is the kernel matrix corresponding to the basic proposition ϕ_c, B is the set of all basic propositions and the parameters d_v ($v \in \mathcal{V}$) are defined as $d_v = a^{|S_v|}$ ($a > 0$).

It is obvious that a trivial computational complexity of computing $\mathcal{K}_u$ ($u \in \mathcal{V}$) is $O(pm^2)$. In practice, this complexity can be reduced to $O(m^2)$ by caching the matrices $\mathcal{K}_u$. For illustration, suppose $\mathcal{K}_{u_1}$ needs to be computed, given that $\mathcal{K}_{u_0}$ is cached and u_0 is a parent of u_1. Let the extra basic proposition contained in ϕ_{u_1} (with respect to ϕ_{u_0}) be ϕ_e. Then $\mathcal{K}_{u_1}$ can be calculated as follows:

$$\mathcal{K}_{u_1} = \mathcal{K}_{u_0} \odot \left(\frac{K_e}{a^2} \right) \oslash \left(\frac{K_e}{(1+a)^2} + \mathbf{1}\mathbf{1}^\top \right),$$

where $\oslash$ is the symbol for element-wise division of matrices.

Hence, plugging the REL specific values in the runtime complexity of the gHKL algorithm, ω =constant and $z = p$, the runtime complexity of the gHKL based REL algorithm is $O(m^3W^3 \log(W) + m^2W^2p)$.

A.12 REL Binary Classification Results in AUC

Table 5 reports the REL binary classification results in AUC (area under the ROC curve). The experimental details (and results measured in F1-score) are discussed in Section 6.

	RuleFit	**SLI**	**ENDER**	**HKL-ℓ_1-MKL**	**gHKL$_\rho$**		
					$\rho = 2$	$\rho = 1.5$	$\rho = 1.1$
TIC	0.736 ± 0.05	0.482 ± 0.21	0.783 ± 0.036	0.836 ± 0.024	0.967 ± 0.023	0.973 ± 0.02	$\mathbf{0.975 \pm 0.018}$
BCW	0.941 ± 0.011	0.917 ± 0.051	0.958 ± 0.039	0.981 ± 0.008	$\mathbf{0.984 \pm 0.005}$	0.93 ± 0.099	0.93 ± 0.099
DIA	0.67 ± 0.027	0.576 ± 0.115	0.761 ± 0.02	0.746 ± 0.050	$\mathbf{0.766 \pm 0.046}$	0.733 ± 0.058	0.636 ± 0.118
HAB	0.537 ± 0.054	0.17 ± 0.155	$\mathbf{0.575 \pm 0.039}$	0.524 ± 0.078	0.556 ± 0.07	0.482 ± 0.11	0.383 ± 0.166
HTC	0.764 ± 0.03	0.541 ± 0.215	0.805 ± 0.031	0.802 ± 0.085	$\mathbf{0.837 \pm 0.035}$	0.763 ± 0.12	0.753 ± 0.118
BLD	0.546 ± 0.06	0.175 ± 0.256	$\mathbf{0.68 \pm 0.028}$	0.660 ± 0.025	0.667 ± 0.034	0.634 ± 0.028	0.519 ± 0.079
HTS	0.765 ± 0.028	0.712 ± 0.085	0.801 ± 0.022	0.825 ± 0.032	$\mathbf{0.849 \pm 0.021}$	0.83 ± 0.027	0.811 ± 0.056
MK3	0.972	0.632	0.998	0.995	**1**	0.998	0.957
VTE	0.955 ± 0.022	0.919 ± 0.048	0.965 ± 0.014	$\mathbf{0.977 \pm 0.009}$	0.972 ± 0.016	0.948 ± 0.015	0.945 ± 0.016
BCC	0.578 ± 0.05	0.469 ± 0.078	0.622 ± 0.043	0.627 ± 0.063	$\mathbf{0.637 \pm 0.055}$	0.576 ± 0.089	0.513 ± 0.124
MAM	0.818 ± 0.02	0.763 ± 0.08	$\mathbf{0.887 \pm 0.006}$	0.866 ± 0.028	0.882 ± 0.023	0.85 ± 0.032	0.839 ± 0.03
LIV	0.607 ± 0.017	0.093 ± 0.168	0.619 ± 0.038	0.619 ± 0.074	$\mathbf{0.623 \pm 0.038}$	0.583 ± 0.11	0.565 ± 0.109

Table 5: Results on binary REL classification. We report the average AUC along with standard deviation, over ten random train-test splits.

References

J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. Saketha Nath, and S. Raman. Variable sparsity kernel learning. *Journal of Machine Learning Research*, 12:565–592, 2011.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4:195–266, 2012.

R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

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

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems*, 2008.

F. Bach. High-dimensional non-linear variable selection through hierarchical kernel learning. Technical report, INRIA, France, 2009.

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

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

S. Ben-David and R. Schuller. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73:273–287, 2008.

A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization: Analysis, algorithms and engineering applications. *MPS/ SIAM Series on Optimization*, 1, 2001.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

K. Blake and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

R. Caruana. Mutitask learning. *Machine Learning*, 28:41–75, 1997.

P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

W. W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *AAAI Conference on Artificial Intelligence*, 1999.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

K. Dembczyński, W. Kotłowski, and R. Słowiński. Maximum likelihood rule ensembles. In *Proceedings of the International Conference of Machine Learning*, 2008.

K. Dembczyński, W. Kotłowski, and R. Słowiński. ENDER - A statistical framework for boosting decision rules. *Data Mining and Knowledge Discovery*, 21:52–90, 2010.

T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

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

J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2:916–954, 2008.

L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems*, 2008.

A. Jain, S. V. N. Vishwanathan, and M. Varma. SPG-GMKL: Generalized multiple kernel learning with a million kernels. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

P. Jawanpuria and J. S. Nath. Multi-task multiple kernel learning. In *SIAM International Conference on Data Mining*, 2011.

P. Jawanpuria and J. S. Nath. A convex feature learning formulation for latent task structure discovery. In *Proceedings of the International Conference on Machine Learning*, 2012.

P. Jawanpuria, J. S. Nath, and G. Ramakrishnan. Efficient rule ensemble learning using hierarchical kernels. In *Proceedings of the International Conference of Machine Learning*, 2011.

P. Jawanpuria, M. Varma, and J. S. Nath. On p-norm path following in multiple kernel learning for non-linear feature selection. In *Proceedings of the International Conference of Machine Learning*, 2014.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_p-norm multiple kernel learning. *Journal of Maching Learning Research*, 12:953–997, 2011.

G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5: 27–72, 2004.

H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, 2007.

J. Liu and J. Ye. Efficient ℓ_1/ℓ_q norm regularization. Technical Report arXiv:1009.4766, 2010.

K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer. Taking advantage of sparsity in multi-task learning. In *Proceedings of the Annual Conference on Learning Theory*, 2009.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

R. S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111–161, 1983.

S. Negahban and M. Wainwright. Phase transitions for high-dimensional joint support recovery. In *Advances in Neural Information Processing Systems*, 2009.

G. Obozinski, Martin J. Wainwright, and M.I. Jordan. Support union recovery in high-dimensional multivariate regression. *Annals of Statistics*, 39:1–17, 2011.

F. Orabona, J. Luo, and B. Caputo. Multi kernel learning with online-batch optimization. *Journal of Machine Learning Research*, 13:227–253, 2012.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*, 1999.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

R. L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT press, Cambridge, 2002.

D. Sheldon. Graphical multi-task learning. Technical report, Cornell University, 2008.

M. Sion. On general minimax theorem. *Pacific Journal of Mathematics*, 1958.

M. Szafranski, Y. Grandvalet, and P. M. Mahoudeaux. Hierarchical penalization. In *Advances in Neural Information Processing Systems*, 2007.

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. *Machine Learning*, 79:73–103, 2010.

B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 47:349–363, 2005.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

S. V. N. Vishwanathan, Z. Sun, N. T.-Ampornpunt, and M. Varma. Multiple kernel learning and the SMO algorithm. In *Advances in Neural Information Processing Systems*, 2010.

S. M. Weiss and N. Indurkhya. Lightweight rule induction. In *Proceedings of the International Conference of Machine Learning*, 2000.

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

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:49–67, 2006.