Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Rule Ensemble Learning Using Hierarchical Kernels in Structured Output Spaces

Naveen Nair
IITB-Monash Research Academy
Dept. of CSE, IIT Bombay
Faculty of IT, Monash University
naveennair@cse.iitb.ac.in

Amrita Saha
Dept. of CSE,
IIT Bombay
amrita@cse.iitb.ac.in

Ganesh Ramakrishnan
Dept. of CSE, IIT Bombay
IITB-Monash Research Academy
ganesh@cse.iitb.ac.in

Shonali Krishnaswamy
Institute for Infocomm Research (I2R), Singapore
Faculty of IT, Monash University
IITB-Monash Research Academy
Shonali.Krishnaswamy@monash.edu

Abstract

The goal in Rule Ensemble Learning (REL) is simultaneous discovery of a small set of simple rules and their optimal weights that lead to good generalization. Rules are assumed to be conjunctions of basic propositions concerning the values taken by the input features. It has been shown that rule ensembles for classification can be learnt optimally and efficiently using hierarchical kernel learning approaches that explore the exponentially large space of conjunctions by exploiting its hierarchical structure. The regularizer employed penalizes large features and thereby selects a small set of short features. In this paper, we generalize the rule ensemble learning using hierarchical kernels (RELHKL) framework to multi class structured output spaces. We build on the StructSVM model for sequence prediction problems and employ a ρ-norm hierarchical regularizer for observation features and a conventional 2-norm regularizer for state transition features. The exponentially large feature space is searched using an active set algorithm and the exponentially large set of constraints are handled using a cutting plane algorithm. The approach can be easily extended to other structured output problems. We perform experiments on activity recognition datasets which are prone to noise, sparseness and skewness. We demonstrate that our approach outperforms other approaches.

Keywords: *activity recognition, hidden markov models, hierarchical kernels, rule learning, structured output spaces, support vector machines.*

1 Introduction

Decision rules form one of the most expressive and human readable representations for learned hypotheses. An if-then decision rule (Rivest 1987) 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 part specifies a value of the function being learned. Rule Ensemble Learning (REL) is the problem of simultaneous discovery of a small set of simple rules and their optimal weights that lead to good generalization. This problem is very similar to that of feature induction (Pietra, Pietra, and Lafferty 1997; McCallum 2003; Nair, Ramakrishnan, and Krishnaswamy 2011), particularly when the features to be induced are conjunctions of basic features. Given this, we often refer to rules as feature conjuncts in the rest of the paper. Most REL and feature induction approaches explore the structured but exponentially large search space of conjunctions using greedy heuristics. Recently, it has been shown (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011) that rule ensembles for binary classification can be learnt optimally and efficiently using hierarchical kernel learning approaches (Bach 2009) that exploit the lattice structure of the search space. We seek to extend this approach to optimally learn feature conjunctions for problems with structured output spaces, such as sequence prediction tasks. Our motivating problem is that of activity recognition, which we briefly describe next.

Activity recognition systems help monitor activities of users in domicile environments. One specific application area is monitoring the daily activities of elderly people living alone, in order to estimate their health condition (Wilson 2005; van Kasteren et al. 2008; Gibson, van Kasteren, and Krose 2008). Such non-intrusive settings typically have on/off sensors installed at various locations in a home. Binary sensor values are recorded at regular time intervals. The joint state of these sensor values at time t form our observations and we will represent them as $\mathbf{x}_t$. The user activity at time t forms the hidden state, which we represent by y_t. The history of sensor readings and corresponding activities (as manually identified later) can be used to train prediction models such as the Hidden Markov Model (HMM) (Rabiner 1989), the Conditional Random Field (CRF) (Lafferty, McCallum, and Pereira 2001) or StructSVM (Tsochantaridis 2006; Tsochantaridis et al. 2004), which could be later used to predict activities based on sensor observations (van Kasteren et al. 2008; Nair, Ramakrishnan, and Krishnaswamy 2011). These approaches typically assume that y_t at time t is independent of all previous activities given y_{t-1} at time $t-1$ and $\mathbf{x}_t$ at time t is independent of all other variables given y_t. Prediction involves determining the label (activity) sequence that best explains the observation (joint state of sensors) sequence using dynamic programming (Forney 1973).

Activity recognition datasets tend to be sparse; that is, one could expect very few sensors to be on at any given time instance. Moreover, in a setting such as activity recognition, one can expect certain combinations of (sensor) readings to be directly indicative of certain activities. While HMMs, CRFs and StructSVM attempt to capture these relations indirectly, (Nair, Ramakrishnan, and Krishnaswamy 2011) illustrate that discovering activity specific conjunctions of sensor readings (as features) can improve the accuracy of prediction. McCallum (2003) follows a similar approach for inducing features for a CRF model. However, both these approaches greedily search the space of conjunctions, since an exhaustive search for the optimal features is exponential in the number of basic features (sensors in the case of activity recognition).

In this paper, we present a generalization of the optimal rule ensemble learning approach (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011) to multi-class structured output spaces, in particular, the sequence prediction problem. In doing so, we naturally extend the existing work on optimal rule ensemble learning to multi-class classification problems. We adopt the loss function from SVM on Structured Output Spaces (StructSVM) (Tsochantaridis et al. 2004; Tsochantaridis 2006) and refer to our approach as StructRELHKL. Our experiments show good improvement over existing approaches.

The paper is organized as follows. Related work is discussed in Section 2. Our proposed StructRELHKL formulation and the algorithms for solving it is elaborated in Section 3. In Section 4, we present our experimental results on the activity recognition problem. We conclude the paper in Section 5.

2 Related Work

Our related work falls into three categories: (i) work on learning for structured output spaces, with particular emphasis on StructSVM, which we build upon (ii) feature induction for structured output spaces and (iii) optimal rule ensemble learning for binary classification using hierarchical kernels. We discuss each of these in the subsections to follow.

Learning for Structured Output Spaces

The objective of learning with structured output spaces is to learn functions of the form $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ from training data, where $\mathcal{X}$ and $\mathcal{Y}$ are input and output sequence spaces respectively. A discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is learned from training data that consists of pairs of input and output sequences. The prediction is performed using the decision function $\mathcal{F}(X; \mathbf{f})$:

$$\mathcal{F}(X; \mathbf{f}) = \underset{Y \in \mathcal{Y}}{\arg\max}\, F(X, Y; \mathbf{f}), \qquad (1)$$

where $F(X, Y; \mathbf{f}) = \langle \mathbf{f}, \boldsymbol{\psi}(X, Y) \rangle$ represents a score which is a scalar value based on the features $\boldsymbol{\psi}$ involving input sequence X and output sequence Y values and parameterized by a parameter vector $\mathbf{f}$. In the case of sequence prediction, features are constructed to represent emission (observation) and transition distributions.

HMMs (Rabiner 1989) and CRFs (Lafferty, McCallum, and Pereira 2001) are traditionally being used in sequence prediction problems. Their ability to capture the state transition dependencies makes these approaches robust in noisy and sparse data. In an HMM setup, probability parameters that maximize the joint probability of input and output training sequences are learned during the training phase. In contrast, CRF learns parameters that maximize the conditional probability of output sequence given input sequence. Prediction is usually done by a dynamic programming algorithm called the Viterbi Algorithm (Forney 1973). Sequence prediction approaches have been used in the activity recognition domain, where the joint state of all the sensors at a time makes an instance of observation variable. The joint state of sensors makes the observation space exponentially large and people tend to assume independence among sensors given an activity, which enables naive factorization. Nair *et al.* (2011) report that the independence assumption could be wrong in many real world settings and in-turn can affect the efficiency of such systems in terms of prediction accuracy. Before explaining their solution in the next subsection, we discuss the StructSVM approach that models sequence prediction as a maximum margin problem.

Tsochantaridis *et al.* generalize the SVM framework to perform classification on structured outputs (Tsochantaridis et al. 2004; Tsochantaridis 2006). This builds on the conventional SVM formulation that assumes output as a single variable which can be a binary label or multi-class. The conventional SVM does not consider the dependencies between output variables and is not suitable for structured data such as sequential data, labeled trees, lattices, or graphs. StructSVM generalizes multi-class Support Vector Machine learning to incorporate features constructed from input and output variables and solve classification problems with structured output data. We now briefly explain their approach in the specific case of sequence prediction.

Loss functions in structured outputs have to measure the amount by which the prediction deviates from the actual value and hence the zero-one classification loss is not sufficient. In sequence prediction, the predicted sequence of labels that are different from the actual labels in a few time steps should be penalized less than those that differ from the actual labels in majority of the time steps. While any decomposable loss-function that holds the above property fits in this approach, the micro-average of wrong predictions is used in this paper. A loss function is represented as $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. $\Delta(Y, \hat{Y})$ is the loss value when the true output is Y and the prediction is $\hat{Y}$.

The SVM formulation for structured output spaces can thus be written as

$$\min_{\mathbf{f},\boldsymbol{\xi}} \ \frac{1}{2} \parallel \mathbf{f} \parallel^2 \ + \frac{C}{m}\sum_{i=1}^{m}\xi_i, \qquad s.t.\ \forall i:\ \xi_i \geq 0$$

$$\forall i,\ \forall\, Y \in \mathcal{Y} \setminus Y_i:\ \langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y)\rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)}. \quad (2)$$

where m is the number of examples, C is the regularization parameter, ξ's are the slack variables introduced to allow errors in the training set in a soft margin SVM formulation, and $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$ represent the i^{th} input and output sequence respectively (Subscript i here is to denote i^{th} example sequence and should not be confused with the i^{th} element of a vector). $\langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y)\rangle$ represents the value $\langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y_i)\rangle - \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y)\rangle$.

In cases where the sequence length is large, the number of constraints in (2) can be extremely large. To solve this problem, an algorithm based on the cutting plane method is proposed by Tsochantaridis *et al.* (*c.f.* algorithm 1 in (Tsochantaridis et al. 2004)) to find a polynomially sized subset of constraints that ensures a solution very near to the optimum. We now discuss two approaches that learn features by a guided search in the lattice to enhance sequence prediction.

Feature Induction for Structured Outputs

In this subsection, we briefly discuss prior work on inducing features for sequence prediction problems, *viz.*, feature induction assisted HMM and feature induction for conditional random fields (CRF).

McCallum *et al.* (2003) as well as Nair *et al.* (2011) propose feature induction methods that iteratively construct feature conjunctions that increase an objective. These approaches start with no features and at each step, consider a set of candidate features (conjunctions or atomic). The features, whose inclusion will lead to maximum increase in the objective are selected. Weights for the new features are trained. The steps are iterated until convergence. While McCallum *et al.* (2003) trains a CRF model and uses conditional log-likelihood as the objective for the greedy induction, Nair *et al.* (2011) train an HMM and use prediction accuracy on a held out dataset (part of the training data) as the objective. This effectively solves the problem of incorrect independence assumption while not dealing with exponential observation space.

Although these greedy feature induction approaches have been shown to improve performance, they cannot guarantee an optimal solution. An exhaustive search to find the optimal solution is expensive due to the exponential size of the search space. We next discuss an approach for optimal induction of feature conjunctions, developed for binary classification problems.

Rule Ensemble Learning using Hierarchical Kernels

Jawanpuria *et al.* (2011) make use of the Hierarchical Kernel Learning (HKL) framework introduced by Bach (2009) to simultaneously learn sparse rule ensembles and their optimal weights. We will refer to their approach as RELHKL. The regularizer used in HKL discourages selection of rules that involve large number of basic features. Jawanpuria *et al.*, prove that although HKL discourages selection of large rules, it redundantly selects all the rules that are subsets of the chosen rules. As a solution, they generalize HKL with a convex formulation using a (1,2] norm (ρ-norm) regularizer that ensures a set of sparse and non redundant rules. A mirror descent based active set algorithm is employed to solve the convex formulation. We briefly discuss their approach in the following paragraphs.

The prime objective of Rule Ensemble Learning (REL) is to learn small set of simple rules and their optimal weights. The set of rules that can be constructed from basic features follow a partial order and can be visualized as a lattice (conjunction lattice when the features are conjunctions of basic features). The set of indices of the nodes in the lattice are represented by $\mathcal{V}$. The model is additive in nature and the weighted sum of the features decides the output. To learn sparse sets of rules, the regularizer $\Omega(\mathbf{f})$ is modified in the following way (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011),

$$\Omega(\mathbf{f}) = \sum_{v\in\mathcal{V}} d_v \parallel \mathbf{f}_{D(v)} \parallel_\rho \qquad (3)$$

where $\mathbf{f}$ is the feature weight vector corresponding to the feature nodes in the lattice, $d_v \geq 0$ is a prior parameter showing usefulness of the feature conjunctions, $\mathbf{f}_{D(v)}$ is the vector with elements as $\parallel f_w \parallel_2 \ \forall w \in D(v)$, and $\parallel . \parallel_\rho$ represents the ρ-norm. In rule ensemble learning d_v is defined as b^i, where b is a constant and $i = |v|$. The optimization problem, with hinge loss, can now be written as,

$$\begin{aligned} &\min_{\mathbf{f},\mathbf{b},\boldsymbol{\xi}} \quad \frac{1}{2}\Omega(\mathbf{f})^2 \ + \ C\sum_{i=1}^{m}\xi_i, \\ &s.t.\ \ y_i\Big(\sum_{v\in\mathcal{V}}\langle f_v, \psi_v(\mathbf{x}_i)\rangle - b\Big) \geq 1 - \xi_i, \\ &\qquad \xi_i \geq 0 \end{aligned} \qquad (4)$$

where y_i and $\mathbf{x}_i$ are output label and input vector respectively for i^{th} example. b is the bias. Other notations are as defined in previous subsections.

Since the 1-norm induces sparsity (Rakotomamonjy et al. 2008; Bach 2009), for most of the $v \in \mathcal{V}$, $\parallel \mathbf{f}_{D(v)} \parallel_\rho = 0$ and this implies $f_w = 0 \ \forall w \in D(v)$. Since norms between $(1, 2)$ promote sparsity (Szafranski and Rakotomamonjy 2008; Jawanpuria, Jagarlapudi, and Ramakrishnan 2011), even if $\parallel \mathbf{f}_{D(v)} \parallel_\rho$ is not forced to zero, many of the $f_w = 0$ for $w \in D(v)$.

At optimality, only a few features are expected to be nonzero. The solution obtained with these non zero features will be the same as the solution obtained with the original set of features. Therefore for computational efficiency, an active set algorithm can be employed (*c.f.* algorithm 1 of (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011)) which starts with an initial set of non zero features. At every step, it solves an optimization problem; a sufficiency condition is checked and it terminates if satisfied. Otherwise the nodes violating the sufficiency condition are added to the active set and the algorithm moves on to the next iteration.

The RELHKL approach is specific to the single variable binary classification problem and cannot be trivially applied to problems involving multi class structured output spaces. In the next section, we develop our approach, that derives from the norm employed in RELHKL and uses the loss function of StructSVM.

3 Rule Ensemble Learning using Hierarchical Kernels in Structured Output Spaces for Sequence Prediction.

We start with the StructSVM formulation for sequence prediction in (2). Let the input/observation at p^{th} time step of the i^{th} example be $\mathbf{x}_i^p$, where $\mathbf{x}_i^p$ is a vector of binary sensor values (each element of the vector represents the value of a sensor fixed at a location such as groceries cupboard, bathroom door etc. at that time step). Similarly, output (activity) at p^{th} time step of the i^{th} example is represented by y_i^p. Let y_i^p can take any of n values. A feature vector, $\boldsymbol{\psi}$, contains entries for emission/observation and the transition distribution. To learn the emission structure, the feature vector has to be modified to include the emission lattice defined in (Nair, Ramakrishnan, and Krishnaswamy 2011). The emission lattice has conjunctions of basic features (sensors) as nodes and obeys a partial order. The top node is the empty conjunction and the bottom node is the conjunction of all the basic features. The nodes at level 1, denoted by B, are basic features themselves. As followed in (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011), $D(v)$ and $A(v)$ represent the set of descendants and ancestors of the node v in the lattice. Both $D(v)$ and $A(v)$ include node v. The hull and the sources of any subset of nodes $\mathcal{W} \subset \mathcal{V}$ are defined as $hull(\mathcal{W}) = \bigcup_{w\in\mathcal{W}} A(w)$ and $sources(\mathcal{W}) = \{w \in \mathcal{W} | A(w) \bigcap \mathcal{W} = \{w\}\}$ respectively. The size of set $\mathcal{W}$ is denoted by $|\mathcal{W}|$. $\mathbf{f}_{\mathcal{W}}$ is the vector with elements as $f_v, v \in \mathcal{W}$. Also let the complement of $\mathcal{W}$ denoted by $\mathcal{W}^c$ be the set of all nodes belonging to the same activity that are not in $\mathcal{W}$. For the sake of visualization, we assume there is a lattice for each label. Therefore, elements of $\boldsymbol{\psi}$ vector correspond to the nodes in the conjunction lattice of each label and the transition features. We represent the emission and transition parts of the vector $\boldsymbol{\psi}$ as $\boldsymbol{\psi_E}$ and $\boldsymbol{\psi_T}$ respectively. We assume that both $\boldsymbol{\psi_E}$ and $\boldsymbol{\psi_T}$ are of dimension equal to the dimension of $\boldsymbol{\psi}$ with zero values for all elements not in their context. That is, $\boldsymbol{\psi_E}$ has dimension of $\boldsymbol{\psi}$, but has zero values corresponding to transition elements. In similar spirit, we split the feature weight vector $\mathbf{f}$ to $\mathbf{f_E}$ and $\mathbf{f_T}$. Similarly, $\mathcal{V}$, the indices of the elements of $\boldsymbol{\psi}$, is split into $\mathcal{V}_\mathbf{E}$ and $\mathcal{V}_\mathbf{T}$.

To use the ρ-norm introduced in (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011) on the feature weights corresponding to the lattice nodes, we separate the regularizer term into those corresponding to emission and transition features and construct the following SVM formulation,

$$\min_{\mathbf{f},\boldsymbol{\xi}} \frac{1}{2}\Omega_E(\mathbf{f_E})^2 \ + \ \frac{1}{2}\Omega_T(\mathbf{f_T})^2 \ + \ \frac{C}{m}\sum_{i=1}^{m}\xi_i,$$
$$\forall i, \forall Y \in \mathcal{Y} \setminus Y_i : \ \langle \mathbf{f}, \boldsymbol{\psi}_i^\delta(Y)\rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)}$$
$$\forall i : \ \xi_i \geq 0 \tag{5}$$

where $\Omega_E(\mathbf{f_E})$ is defined in (3) as $\sum_{v\in\mathcal{V}_\mathbf{E}} d_v \parallel \mathbf{f}_{\mathbf{E}D(v)} \parallel_\rho, \ \ \rho \in (1,2]$ and $\Omega_T(\mathbf{f_T})$ is the 2-norm regularizer $\left(\sum_i f_{Ti}^2\right)^{\frac{1}{2}}$

The 1-norm in $\Omega_E(\mathbf{f_E})$ forces many of the $\parallel \mathbf{f}_{\mathbf{E}D(v)} \parallel_\rho$ to be zero. Even in cases where $\parallel \mathbf{f}_{\mathbf{E}D(v)} \parallel_\rho$ is not forced to zero, the ρ-norm forces many of node v's descendants to zero. As transition feature space is not exponential, no sparsity is desired and therefore a 2-norm regularizer is sufficient for transition. The above SVM setup has two inherent issues which makes it a hard problem to solve. The first is that the regularizer, $\Omega_E(\mathbf{f_E})$, consists of ρ-norm over descendants of each lattice node, which makes it exponentially expensive. The second problem is the exponential number of constraints for the objective. The rest of the section discusses how to solve the problem efficiently.

By solving (5), we expect most of the emission feature weights to be zero. As illustrated by Jawanpuria *et al.*, the solution to the problem when solved with the original set of features is the same but requires less computation when solved only with features having non zero weights at optimality. Therefore, an active set algorithm can be employed to incrementally find the optimal set of non zero weights (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011). In each iteration of the active set algorithm, since the constraint set in (5) is exponential, a cutting plane algorithm has to be used to find a subset of constraints of polynomial size so that the corresponding solution satisfies all the constraints with an error not more than ϵ. We now modify (5) to consider only the active set of features $\mathcal{W}$.

$$\min_{\mathbf{f},\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{v\in\mathcal{W}} d_v \parallel \mathbf{f}_{\mathbf{E}D(v)\bigcap\mathcal{W}} \parallel_\rho\right)^2 \ + \ \frac{1}{2} \parallel \mathbf{f_T} \parallel_2^2 \ + \frac{C}{m}\sum_{i=1}^{m}\xi_i,$$
$$\forall i, \forall Y \in \mathcal{Y} \setminus Y_i :$$
$$-\left(\sum_{v\in\mathcal{W}} \langle f_{Ev}, \psi_{Evi}^\delta(Y)\rangle + \sum_{v\in\mathcal{V}_\mathbf{T}} \langle f_{Tv}, \psi_{Tvi}^\delta(Y)\rangle + \frac{\xi_i}{\Delta(Y_i, Y)} - 1\right) \leq 0$$
$$\forall i : \ -\xi_i \leq 0 \tag{6}$$

where $\rho \in$ (1,2]

The active set algorithm can be terminated when the solution to the small problem (reduced solution) is the same as the solution to the original problem; otherwise the active set has to be updated. We follow a similar approach to that defined in (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011) for deriving a sufficiency condition to check optimality, which we discuss in the following paragraphs.

Applying lemma 26 in (Micchelli and Pontil 2005), the regularizer term corresponding to the emission weights in (5) can be written as,

$$\Omega_E(\mathbf{f_E})^2 = \underset{\gamma\in\Delta_{|\mathcal{V}_\mathbf{E}|,1}}{min} \ \underset{\lambda_v\in\Delta_{|D(v)|,\hat{\rho}}\forall v\in\mathcal{V}_\mathbf{E}}{min} \sum_{w\in\mathcal{V}_\mathbf{E}} \delta_w^{-1}(\gamma,\lambda) \parallel f_{Ew} \parallel_2^2$$

where, $\hat{\rho} = \frac{\rho}{2-\rho}$, $\Delta_{d,r} = \left\{\eta \in \mathbb{R}^d | \eta \geq 0, \sum_{i=1}^{d} \eta_i^r = 1\right\}$, and $\delta_w(\gamma,\lambda)^{-1} = \sum_{v\in A(w)} \frac{d_v^2}{\gamma_v \lambda_{wv}}$.

Using the variational characterization and representer theorem (Rakotomamonjy et al. 2008), the partial dual (wrt. $\mathbf{f}, \boldsymbol{\xi}$) of (5) can be derived as,

$$\min_{\gamma\in\Delta_{|\mathcal{V}_\mathbf{E}|,1}} \ \min_{\lambda_v\in\Delta_{|D(v)|,\hat{\rho}}\forall v\in\mathcal{V}_\mathbf{E}} \ \max_{\alpha\in S(\mathcal{Y},C)} \ G(\gamma,\lambda,\alpha) \tag{7}$$

where

$$G(\gamma, \lambda, \alpha) = \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \boldsymbol{\alpha}^\top \left(\sum_{w \in \mathcal{V}_\mathbf{E}} \delta_w(\gamma, \lambda) \boldsymbol{\kappa}_{\mathbf{E}w} \right) \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{\kappa}_\mathbf{T} \boldsymbol{\alpha}$$

and

$$S(\mathcal{Y}, C) = \{\boldsymbol{\alpha} \in \mathbb{R}^{m(n^l - 1)} \mid \alpha_{i,Y} \geq 0,\ m \sum_{Y \neq Y_i} \frac{\alpha_{iY}}{\Delta(Y, Y_i)} \leq C,\ \forall i, Y\}$$

.

Let the duality gap with $(\gamma, \lambda, \alpha)$ in (7) be given by

$$\begin{aligned} \max_{\hat{\alpha} \in S(\mathcal{Y}, C)} G(\gamma, \lambda, \hat{\alpha}) - \min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_\mathbf{E}|, 1}} \min_{\hat{\lambda_v} \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_\mathbf{E}} G(\hat{\gamma}, \hat{\lambda}, \alpha) \\ \leq \quad \frac{1}{2} \Omega_E(\mathbf{f_E})^2 + \frac{1}{2} \Omega_T(\mathbf{f_T})^2 + \frac{C}{m} \sum_i \xi_i - \\ \Big(\mathop{min}_{\hat{\gamma} \in \Delta_{|\mathcal{V}|, 1}} \mathop{min}_{\hat{\lambda_v} \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_\mathbf{E}} \sum_{i, Y \neq Y_i} \alpha_{iY} - \\ \frac{1}{2} \sum_{w \in \mathcal{V}_\mathbf{E}} \delta_w(\gamma, \lambda) \boldsymbol{\alpha}^\top \mathbf{K}_{\mathbf{E}w} \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{K}_\mathbf{T} \boldsymbol{\alpha} \Big) \end{aligned}$$

From this, we can derive a sufficient condition for the reduced solution with $\mathcal{W}$ to have a duality gap less than ϵ as,

$$\begin{aligned} \max_{u \in sources(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \boldsymbol{\alpha}_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \Big(\prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{b^2} \Big) \\ \Big(\prod_{k \notin u} (1 + \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{(1+b)^2}) \Big) \boldsymbol{\alpha}_{\mathcal{W}jY'} \\ \leq \Omega_E(\mathbf{f}_{\mathbf{E}\mathcal{W}})^2 + \Omega_T(\mathbf{f}_{\mathbf{T}\mathcal{W}})^2 + 2(\epsilon - e_\mathcal{W}) \end{aligned} \tag{8}$$

where
$e_\mathcal{W} = \Omega_E(\mathbf{f}_{\mathbf{E}\mathcal{W}})^2 + \Omega_T(\mathbf{f}_{\mathbf{T}\mathcal{W}})^2 + \frac{C}{m} \sum_i \xi_i + \frac{1}{2} \boldsymbol{\alpha}_\mathcal{W}^\top \boldsymbol{\kappa}_\mathbf{T} \boldsymbol{\alpha}_\mathcal{W} - \sum_{i, Y \neq Y_i} \alpha_{\mathcal{W}iY}$.
If the current solution satisfies the above condition in any iteration of the active set, the algorithm terminates; else the active set is updated by adding the nodes in $sources(\mathcal{W}^c)$ which violate the condition. To solve the optimization problem efficiently, we now derive the complete dual of (5) from the partial dual (7) as,

$$\min_{\eta \in \Delta_{|\mathcal{V}|, 1}} g(\eta) \tag{9}$$

where $g(\eta)$ is defined as,

$$\max_{\alpha \in S(\mathcal{Y}, C)} \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{\kappa}_\mathbf{T} \boldsymbol{\alpha} - \frac{1}{2} \Big(\sum_{w \in \mathcal{V}} \zeta_w(\eta) (\boldsymbol{\alpha}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \boldsymbol{\alpha})^{\bar{\rho}} \Big)^{\frac{1}{\bar{\rho}}}, \tag{10}$$

$\zeta_w(\eta) = \big(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \big)^{\frac{1}{1-\rho}}$ and $\bar{\rho} = \frac{\rho}{2(\rho - 1)}$.

Since (9) is a 1-norm regularized problem, many of the ηs are expected to be zero at optimality. A zero value for η at node v makes the weights $\zeta_w(\eta)$ of all of v's descendant nodes w to be equal to zero and essentially discourages selection of kernels near the bottom of the lattice (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011). It can be shown that the maximization term is similar to a $\hat{\rho}$-norm ($\hat{\rho} = \frac{\rho}{2-\rho}$) MKL formulation (Kloft et al. 2009). If $\rho \in (1, 2)$, the kernel κ_w may not be selected even in cases when $\zeta_w(\eta) \neq 0$ (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011). Therefore the formulation ensures that large conjunctive features are not selected and that selection of a feature does not warrant selection of its subsets.

```
Input: Training data D, Oracle for computing kernels, Maximum tolerance $\epsilon$
Initialize $\mathcal{W} = Top\ nodes$ in the lattice as the active set
Compute $\eta, \alpha$ by solving (9) using mirror descent
while sufficiency condition is not satisfied, do
    Add sufficiency condition violating nodes to active set $\mathcal{W}$
    Recompute $\eta, \alpha$ by solving (9)
end while
Output: Active-set $W, \eta, \alpha$
```

Figure 1: Active set algorithm for solving RELHKL in structured output spaces.

The solution to the dual problem in (9) with $\mathcal{V}$ restricted to active set $\mathcal{W}$ gives the solution to the restricted primal problem given in (6). The active set algorithm, adapted from (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011), is briefly outlined in Figure 1. It starts with top nodes in the lattice and iteratively adds new nodes that violate the sufficiency condition. Parameters are updated in each step of active set by solving (9). We follow the mirror descent approach to solve (9) (Jawanpuria, Jagarlapudi, and Ramakrishnan 2011).

Let $\bar{\alpha}$ be the optimal solution to (10) with some η, then the i^{th} sub-gradient is computed as

$$\begin{aligned} (\triangledown g(\eta))_i = -\frac{d_i^\rho \eta_i^{-\rho}}{2\bar{\rho}} \Big(\sum_{w \in \mathcal{V}_\mathbf{E}} \zeta_w(\eta) (\bar{\boldsymbol{\alpha}}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\boldsymbol{\alpha}})^{\bar{\rho}} \Big)^{\frac{1}{\bar{\rho}} - 1} \\ \Big(\sum_{w \in D(i)} \zeta_w(\eta)^\rho (\bar{\boldsymbol{\alpha}}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\boldsymbol{\alpha}})^{\bar{\rho}} \Big) \end{aligned} \tag{11}$$

To compute the gradient, $\bar{\alpha}$ is to be obtained by solving (10) using the cutting plane method. The cutting plane algorithm, adapted from (Tsochantaridis et al. 2004), is outlined in Figure 2. The algorithm starts with no constraints for (6) and in each step, adds a constraint that most violates the margin. The dual problem (10) is then solved and the process is continued. The algorithm stops when there are no more margin violations.

In this section, we derived an approach that builds on structSVM framework and uses a structured output extension of RELHKL to construct a small set of simple emission features. We use the active set algorithm to handle exponential feature conjunction space. Each active set iteration solves the dual formulation using mirror descent algorithm. The subproblem in mirror descent step is solved efficiently by a cutting plane algorithm that handles the exponential constraint space.

4 Experiments and Results

Our entire implementation is in Java. Our experiments are carried out on two publicly available activity recognition datasets. The first is the data provided by Kasteren *et al.* (2008). The dataset is extracted from a household fitted with 14 binary sensors. Eight activities have been annotated for 4 weeks. Activities are daily house hold activities like $sleeping$, $usingToilet$, $preparingDinner$, $preparingBreakfast$, $leavingOut$, *etc.* There are 40006 data instances. Since the authors of the dataset are from the University of Amsterdam, we will refer to the dataset as the

```
Input: kernels, C, $\epsilon_{margin}$
$S_i \leftarrow \phi \quad \forall i = 1, ..., m$
repeat
    for $i = 1, ..., m$ do
        Define $H(Y) \equiv \left[1 - \langle \mathbf{f}, \psi_i^{\delta}(Y) \rangle\right] \Delta(Y_i, Y)$
        Compute $\hat{Y} = \arg\max_{Y \in \mathcal{Y}} \ H(Y)$.
        Compute $\xi_i = \max\{0, \underset{Y \in S_i}{max} \ H(Y)\}$.
        if $H(\hat{Y}) > \xi_i + \epsilon_{margin}$, then
            $S_i \leftarrow S_i \bigcup \{\hat{Y}\}$.
            $\boldsymbol{\alpha} \leftarrow$ optimize dual over $S$, $S = \bigcup_i \ S_i$.
        end if
    end for
until no $S_i$ has changed during the iteration.
```

Figure 2: Cutting plane algorithm for solving dual with a fixed η

	Micro avg.	Macro avg.
Std. HMM	25.40 (±18.55)	21.75 (±12.12)
B&B HMM	29.54 (±20.70)	16.39 (±02.74)
Greedy FIHMM	58.08 (±10.14)	26.84 (±04.41)
StructSVM	58.02 (±11.87)	35.00 (±05.24)
CRF	48.49 (±05.02)	20.65 (±04.82)
FICRF	59.52 (±11.76)	33.60 (±07.38)
RELHKL	46.28 (±11.44)	23.11 (±07.46)
StructRELHKL	63.96 (±05.74)	32.01 (±03.04)

Table 1: Micro average accuracy and macro average accuracy of classification in percentage using standard HMM, B&B learning assisted HMM, greedy feature induction assisted HMM, StructSVM, CRF, CRF with feature induction, RELHKL without transitions and the proposed StructRELHKL approach on UA dataset.

UA data. The second data is recorded at MIT Place-Lab by Tapia *et al.* (2003; 2004) (we call the dataset PlaceLab data). The data is extracted from two single-person's apartments (subject one and subject two). The apartments are fitted with 76 and 70 sensors for subject one and two respectively and data is collected for two weeks. Annotated activities are categorized into nine high level activities such as $employmentRelated$, $personalNeeds$, $domesticWork$, $educational$, $entertainment$, *etc.*

We use 25% of data for training and the rest for testing and report all accuracies by average across the four folds (the dataset is split into different sequences and each sequence is treated as an example). We report both micro-average and macro-average prediction accuracies. The micro-average accuracy is referred to as time-slice accuracy in (van Kasteren et al. 2008), and is the weighted average of per-class accuracies, weighted by the number of instances of the class. Macro-average accuracy, referred to as class accuracy in (van Kasteren et al. 2008), is simply the average of the per-class accuracies. Micro-averaged accuracy is typically used as the performance evaluation measure. However in data that is biased towards some classes, macro-average also is an indicator of the quality of the model.

In a typical activity recognition setting, observations are sparse and therefore, the systems that do not consider the temporal dependencies between activities fail to give comparable results, as observed in our experiments. For example, activities like sleeping may cause a sensor at bedroom door to fire only at the start and end of the sleeping period. It is intuitive to think that a person most likely will be sleeping at the current time step if he was sleeping at the previous time step. This dependency is taken care by transition distribution. In the following paragraphs, we compare our approach with other approaches that gave comparable results.

For UA data, we compare our results with seven other approaches: (a) standard HMM, (b) Branch and Bound structure learning assisted HMM model construction (B&BHMM), where the rules learned by Aleph (Srinivasan 2007) (an ILP system which learns definite rules from examples) for each activity determine the HMM emission structure, (c) greedy feature induction assisted HMM approach (Greedy FIHMM), (d) StructSVM approach, (e) Conditional Random Field (CRF), (f) Conditional Random Field with Feature Induction (FICRF) (McCallum 2003; 2002), (g) RELHKL (without considering transitions). While standard approaches such as HMM, CRF and structSVM use basic features (binary sensor values) as emission features, feature induction approaches such as Greedy FIHMM and FICRF use conjunctions of basic features as emission features. In contrast to greedy feature induction approaches, RELHKL and StructRELHKL find the feature conjunctions efficiently and optimally. While RELHKL without the transition features does not consider the structure in output space, StructRELHKL does the classification in structured output space and performs better. The results are summarized in Table 1. We observed that the proposed StructRELHKL approach outperforms all the other approaches in micro-averaged accuracy. While our macro average accuracy is comparable to FICRF, StructSVM and outperforms others, our standard deviation is much less. This suggests that the model is not skewed towards any particular activity. In general, our approach exhibits much lower standard deviation, reflecting its consistency.

In our experiments on PlaceLab dataset, we observed that the performance of standard HMM, B&B structure learning assisted HMM, and RELHKL without transition features was poor and the greedy feature induction assisted HMM did not converge at all. Hence we compare our results with (a) StructSVM approach, (b) Conditional Random Field (CRF), and (c) Conditional Random Field with Feature Induction (FICRF) (McCallum 2003; 2002). The results are summarized in Table 2. Our results show that StructRELHKL performs better than other approaches in micro-averaged accuracy for both subject one and two, while maintaining comparable macro-averaged class accuracies. Our approach shows less standard deviation in subject one data while giving comparable standard deviation in subject two data.

In a setting with n activities and s sensors, an exhaustive search for optimum features needs evaluation at $n \times 2^s$ nodes. This amounts to 131072 nodes in UA data and to the order of 10^{22} in PlaceLab data, which is computationally in-

		Micro avg.	Macro avg.
Subject 1	StructSVM	75.03 (±04.51)	26.99 (±07.73)
	CRF	65.54 (±06.80)	31.19 (±07.39)
	FICRF	68.52 (±07.19)	29.77 (±03.59)
	StructRELHKL	82.88 (±0.43)	28.92 (±01.53)
Subject 2	StructSVM	63.49 (±02.75)	25.33 (±05.8)
	CRF	50.23 (±06.80)	27.42 (±07.65)
	FICRF	51.86 (±07.35)	26.11 (±05.89)
	StructRELHKL	67.16 (±08.64)	24.32 (±02.12)

Table 2: Micro average accuracy and macro average accuracy of classification in percentage using StructSVM, CRF, CRF with feature induction and the proposed StructRELHKL approach on PlaceLab dataset. (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either failed to give comparable results or did not converge)

feasible. In contrast, due to the active-set algorithm and sufficiency condition check, our approach explores only a few thousand nodes and converges in 24 hours approximately. Time for prediction is as fast as other approaches we compare against.

The approach discovered rules such as $usingToilet \leftarrow bathroomDoor \wedge toiletFlush$, $sleeping \leftarrow bedroomDoor \wedge toiletDoor \wedge bathroomDoor$, $preparingDinner \leftarrow groceriesCupboard$, *etc.* The conjunction $bathroomDoor \wedge toiletFlush$ strongly indicates that the activity is $usingToilet$ while $groceriesCupboard$ indicates a higher chance of $preparingDinner$. Similarly $bedroomDoor \wedge toiletDoor \wedge bathroomDoor$ increases the chance of predicting $sleeping$ as the activity. This is reasonable as people access these doors during night before going to sleep and the sensors at $bedroomDoor$, $toiletDoor$, and $bathroomDoor$ fire once when the person access the door and goes to off-mode while s/he is sleeping. But since, the conjunction just before sleep gives a higher weight to the activity $sleeping$, the weight gets accrued and gets combined with transition weights to accurately predict the activity as $sleeping$.

5 Conclusion

Rule Ensemble Learning using Hierarchical Kernels (RELHKL) has been proved to be effective in learning optimal feature conjuncts for binary classification problems. In this paper, we presented a generalization of the RELHKL framework to multi class structured output spaces, and in particular, for the sequence prediction problem. We have demonstrated the effectiveness of our approach in activity recognition settings.

Acknowledgments

We thank Microsoft Research India for providing partial travel grant support to attend the conference.

References

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

Forney, G.D., J. 1973. The viterbi algorithm. *Proceedings of IEEE* 61(3):268–278.

Gibson, C.; van Kasteren, T.; and Krose, B. 2008. Monitoring homes with wireless sensor networks. *Proceedings of the International Med-e-Tel Conference*.

Jawanpuria, P.; Jagarlapudi, S. N.; and Ramakrishnan, G. 2011. Efficient rule ensemble learning using hierarchical kernels. *International Conference on Machine Learning*.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; Laskov, P.; Muller, K. R.; and Zien, A. 2009. Efficient and accurate p-norm multiple kernel learning. *NIPS*.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

McCallum, A. K. 2003. Efficiently inducing features of conditional random fields. Proceedings of the Nineteenth Conference Annual Conference on Uncertainty in Artificial Intelligence.

Micchelli, C., and Pontil, M. 2005. Learning the kernel function via regularization. *Journal of Machine Learning Research*.

Nair, N.; Ramakrishnan, G.; and Krishnaswamy, S. 2011. Enhancing activity recognition in smart homes using feature induction. *International Conference on Data Warehousing and Knowledge Discovery*.

Pietra, S. D.; Pietra, V. J. D.; and Lafferty, J. D. 1997. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(4):380–393.

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–286.

Rakotomamonjy, A.; Bach, F.; Canu, S.; and Grandvalet, Y. 2008. Simplemkl. *JMLR* 9:2491–2521.

Rivest, R. L. 1987. Learning decision lists. *Machine Learning* 2(3):229–246.

Srinivasan, A. 2007. The aleph manual. *Technical Report, University of Oxford*.

Szafranski, M., and Rakotomamonjy, A. 2008. Composite kernel learning. *ICML*.

Tapia, E. M. 2003. Activity recognition in the home setting using simple and ubiquitous sensors. *S.M. Thesis, Massachusetts Institute of Technology*.

Tapia, E. M. 2004. Activity recognition in the home using simple and ubiquitous sensors. *International Conference on Pervasive Computing*.

Tsochantaridis, I.; Hofmann, T.; Joachims, T.; and Altun, Y. 2004. Support vector machine learning for interdependent and structured output spaces. *International Conference on Machine Learning*.

Tsochantaridis, I. 2006. Support vector machine learning for interdependent and structured output spaces.

van Kasteren, T.; Noulas, A.; Englebienne, G.; and krose, B. 2008. Accurate activity recognition in a home setting. *10th International conference on Ubiquitous computing*.

Wilson, D. H. 2005. Assistive intelligent environments for automatic health monitoring. *PhD Thesis, Carnegie Mellon University*.