Efficient Rule Ensemble Learning using Hierarchical Kernels

Ganesh Ramakrishnan

Collaboration: J. Saketha Nath, Pratik J., Naveen Nair and Amrita Saha.

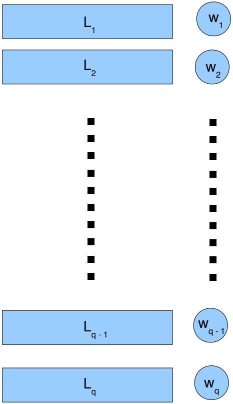
Indian Institute of Technology — Bombay

Rule Ensembles — Overview

- Ensembles with base learners as *simple* rules (Cohen&Singer, 99)

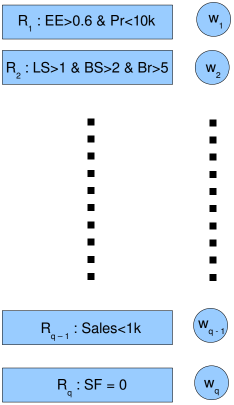
Rule Ensembles — Overview

- Ensembles with base learners as *simple* rules (Cohen&Singer, 99)



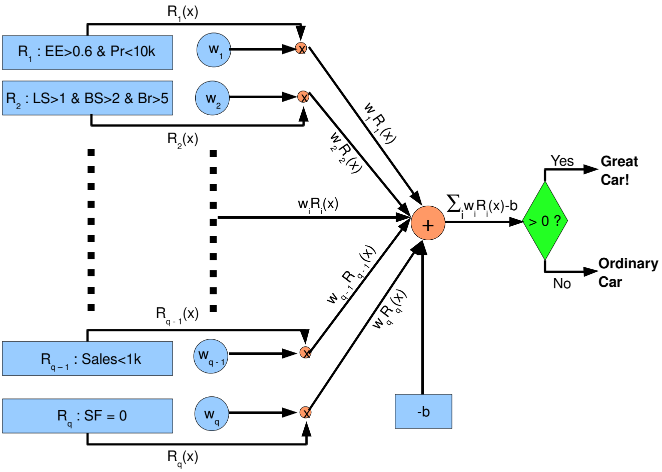
Rule Ensembles — Overview

- Ensembles with base learners as *simple* rules (Cohen&Singer, 99)



Rule Ensembles — Overview

- Ensembles with base learners as *simple* rules (Cohen&Singer, 99)



Rule Ensembles — Key Features

- Highly interpretable hypothesis
  - Small set of rules i.e., low q
  - *Simple* rules e.g., short conjunctive propositions

Rule Ensembles — Key Features

- Highly interpretable hypothesis
  - Small set of rules i.e., low q
  - *Simple* rules e.g., short conjunctive propositions
- Better generalization than conventional rule learners

Rule Ensemble Learning — Formal Definition

Input:

- Training Set: $\mathcal{D}=\left\{(\mathbf{x}^1,y^1),\ldots,(\mathbf{x}^m,y^m)\right\}$, $\mathbf{x}^i\in\mathbb{R}^n$ and $y^i\in\{-1,1\}$
- Basic propositions regarding input features (say, p in number)

  Nominal e.g., $x_i = a$ and $x_i \neq a$

  Numeric e.g., $x_j \geq b$ and $x_j \leq b$

Rule Ensemble Learning — Formal Definition

Input:

- Training Set: $\mathcal{D} = \left\{ (\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^m, y^m) \right\}$, $\mathbf{x}^i \in \mathbb{R}^n$ and $y^i \in \{-1, 1\}$
- Basic propositions regarding input features (say, p in number)

  Nominal e.g., $x_i = a$ and $x_i \neq a$

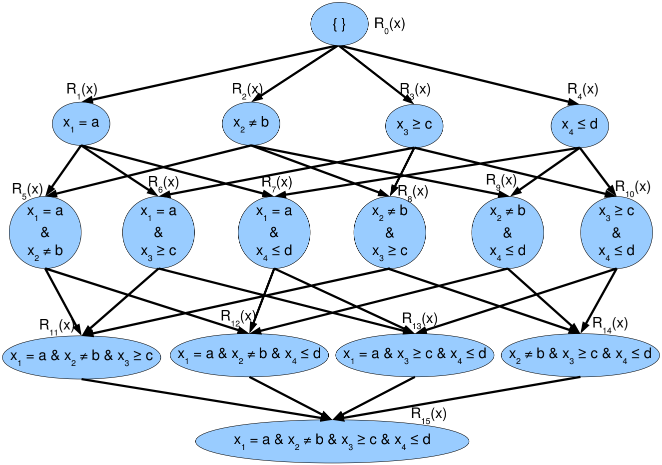
  Numeric e.g., $x_j \geq b$ and $x_j \leq b$

Goal:

- Construct conjunctive rules from basic propositions
  - Few in number
  - Short conjunctions
- Compute corresponding weights $(\mathbf{w}, b)$

Rule Ensemble Learning — Challenging task

Extremely large, atleast $O(2^n)$, rule space!



Rule Ensembles — Existing Methods

SLIPPER(Cohen&Singer, 99): AdaBoost + RIPPER — greedy

RuleFit(Friedman&Popescu, 08): ISLE + decision tree — greedy

ELCS(Gao et.al., 07): Genetic Alg. + post-pruning — sub-optimal

ENDER(Dembczynski et.al., 10): Minimization of empirical risk — greedy

Rule Ensembles — Existing Methods

SLIPPER(Cohen&Singer, 99): AdaBoost + RIPPER — greedy

RuleFit(Friedman&Popescu, 08): ISLE + decision tree — greedy

ELCS(Gao et.al., 07): Genetic Alg. + post-pruning — sub-optimal

ENDER(Dembczynski et.al., 10): Minimization of empirical risk — greedy

Proposed Methodology — Overview

Optimal search for rules over all conjunctions

- Regularized loss minimization
- Convex formulation
- Discovers compact ruleset (small set with short rules)

Proposed Methodology — Overview

Optimal search for rules over all conjunctions

- Regularized loss minimization
- Convex formulation
- Discovers compact ruleset (small set with short rules)

Technical Contribution:

Efficient mirror-descent based active set method

- Complexity: polynomial in active set size ($\ll 2^p$)

Proposed Methodology — Overview

Optimal search for rules over all conjunctions

- Regularized loss minimization
- Convex formulation
- Discovers compact ruleset (small set with short rules)

Technical Contribution:

Efficient mirror-descent based active set method

- Complexity: polynomial in active set size ($\ll 2^p$)

Key Structure Exploited:

Sub-lattices with long rules are discouraged.

A Primitive Formulation

- Decision function[1]: sign $\left(\sum_{v \in \mathcal{V}} w_v R_v(\mathbf{x}) - b\right)$
- l_1 regularize to force many w_v to zero

[1] $\mathcal{V}$ is index set for conjunctive lattice

A Primitive Formulation

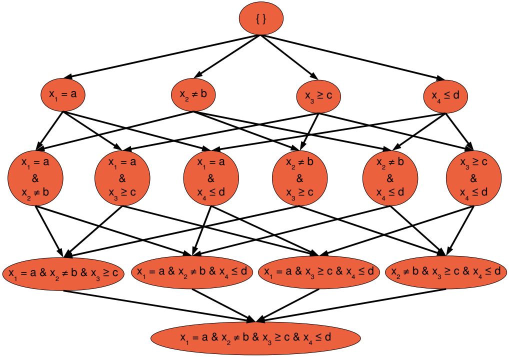
- Decision function[1]: sign $\left(\sum_{v \in \mathcal{V}} w_v R_v(\mathbf{x}) - b\right)$
- l_1 regularize to force many w_v to zero

l_1 regularized formulation:

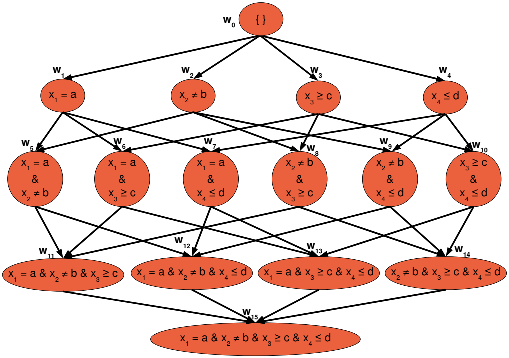
$$\min_{\mathbf{w}, b} \frac{1}{2} \left(\sum_{v \in \mathcal{V}} |w_v| \right)^2 + C \sum_{i=1}^{m} L \left(y^i, \sum_{v \in \mathcal{V}} w_v R_v(\mathbf{x}^i) - b \right)$$

[1] $\mathcal{V}$ is index set for conjunctive lattice

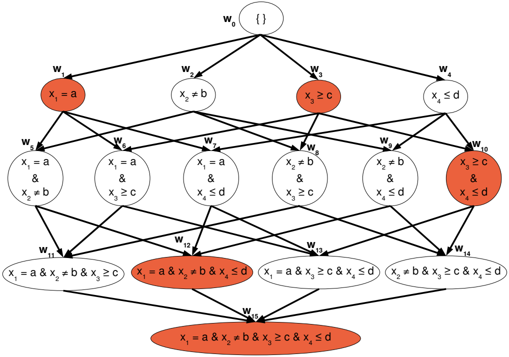
A Primitive Formulation



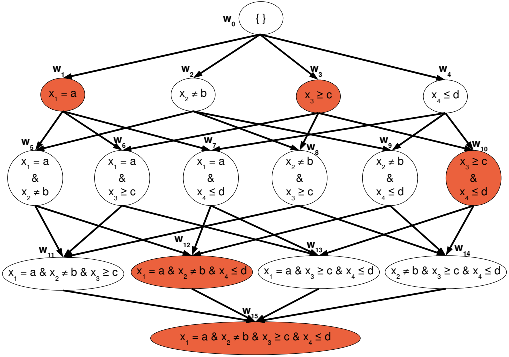
A Primitive Formulation



A Primitive Formulation



A Primitive Formulation



Short-comings:

- long rules may be selected
- Computationally difficult problem

An Improved Formulation

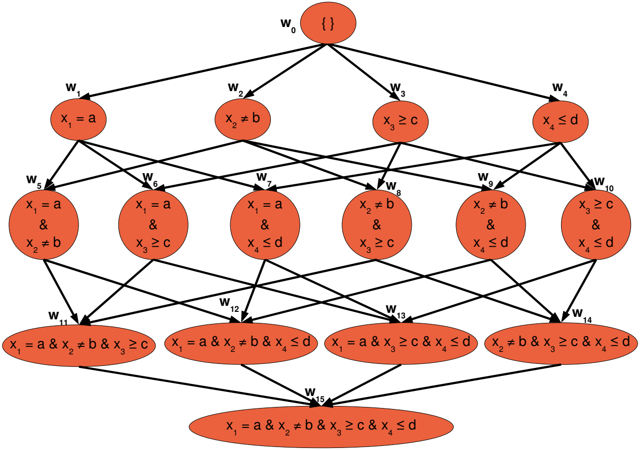
Key Idea:

Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$

An Improved Formulation

Key Idea:

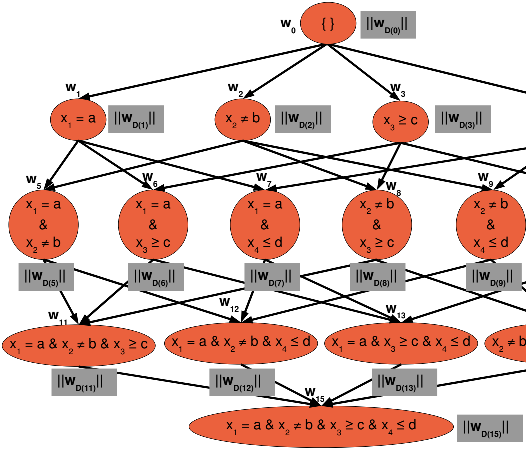
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

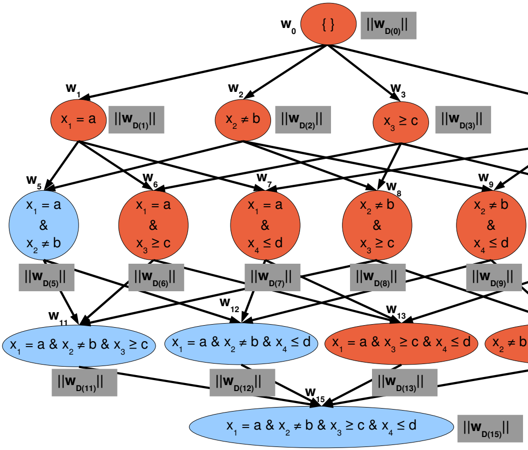
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

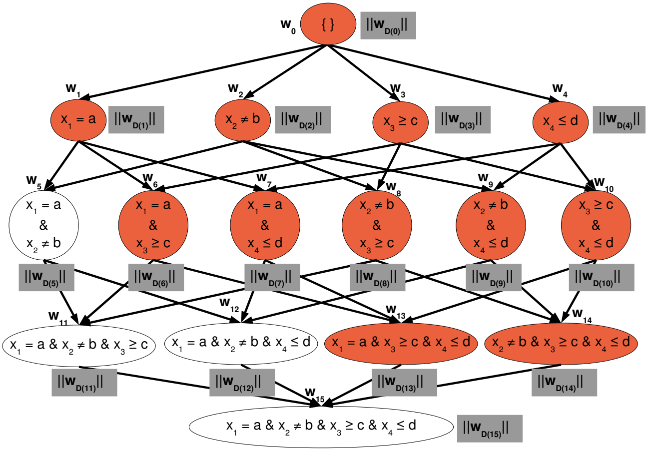
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

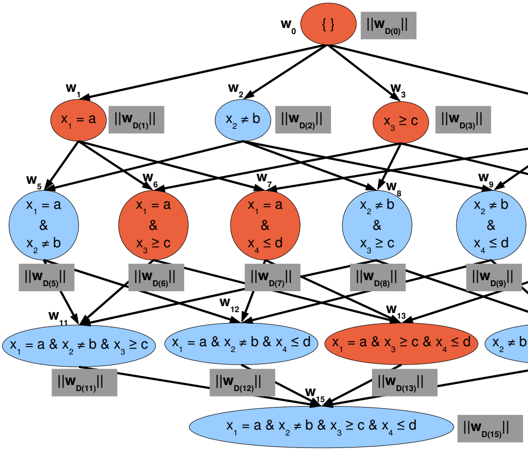
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

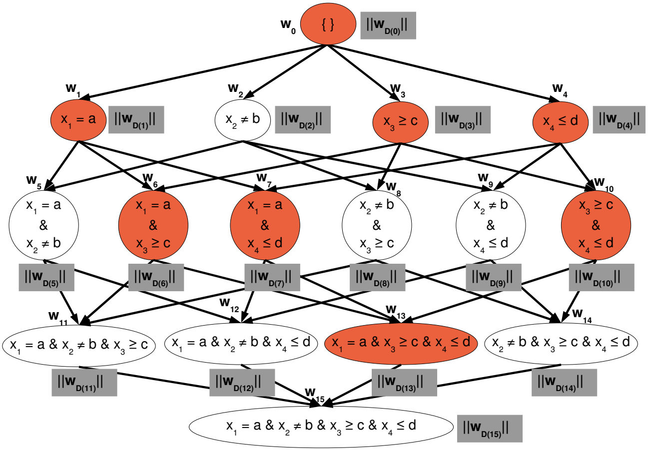
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

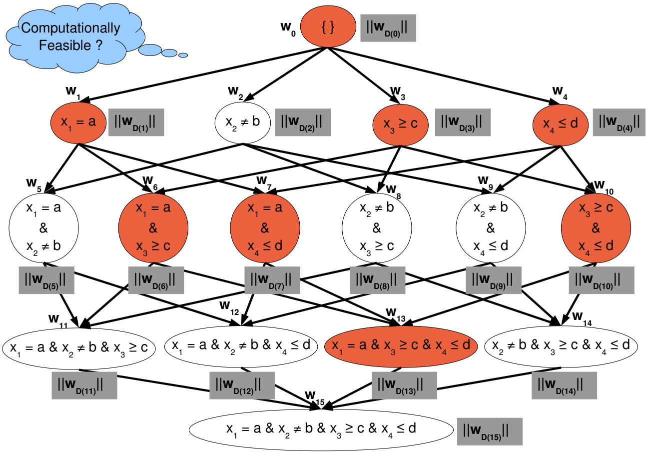
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

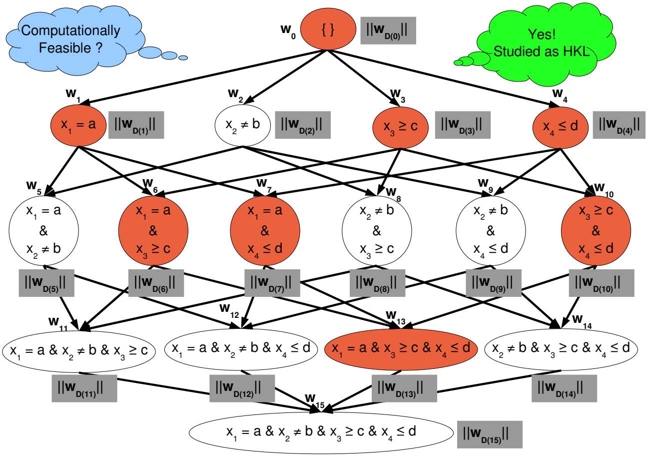
Block l_1 regularizer discourages long rules: $\left(\sum_{v\in\mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$

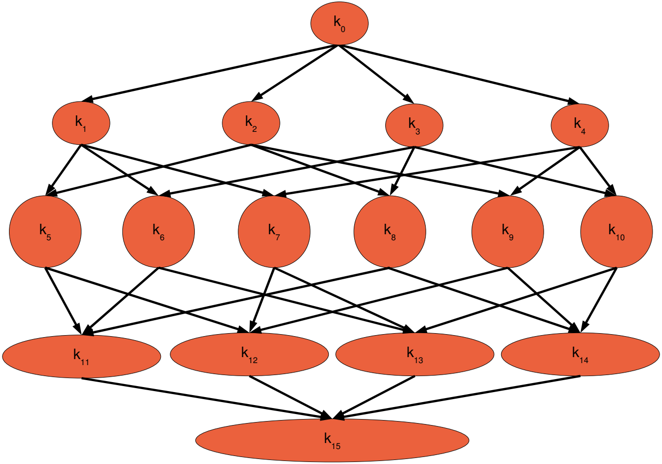


Hierarchical Kernel Learning (HKL) (Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)

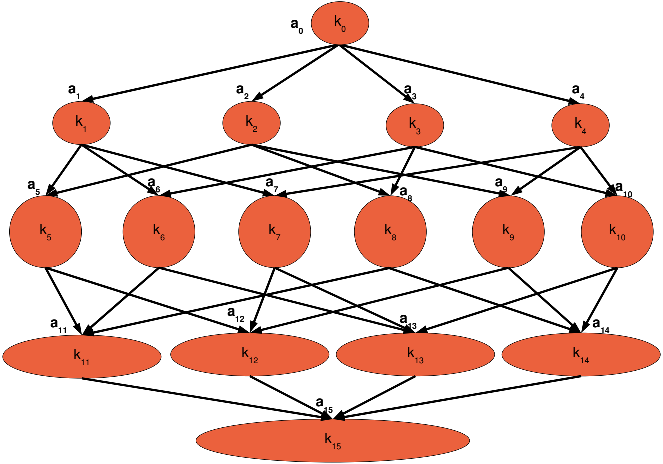
Hierarchical Kernel Learning (HKL)(Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)



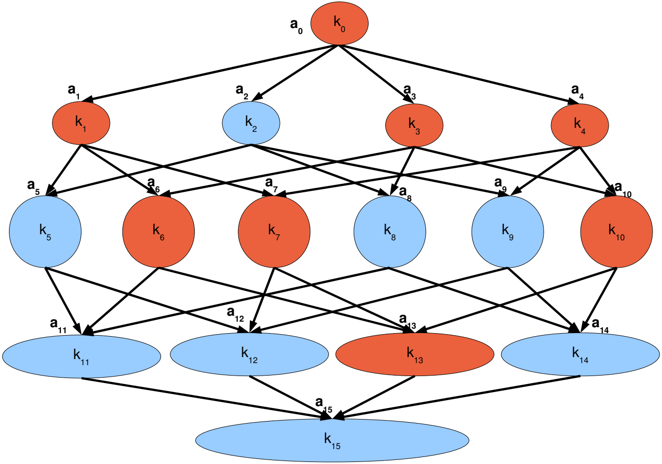
Hierarchical Kernel Learning (HKL)(Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)



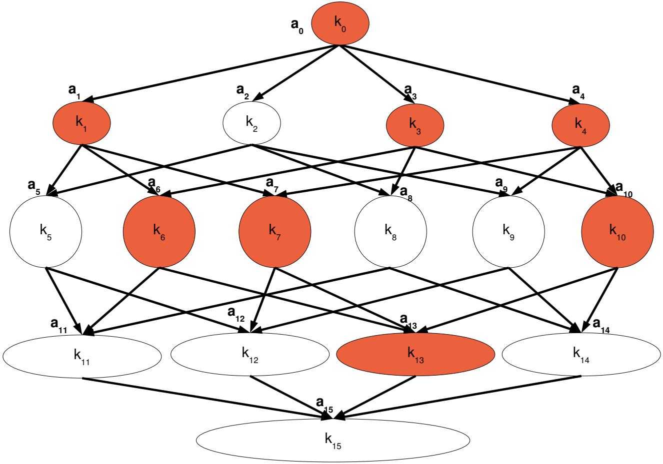
Hierarchical Kernel Learning (HKL)(Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)



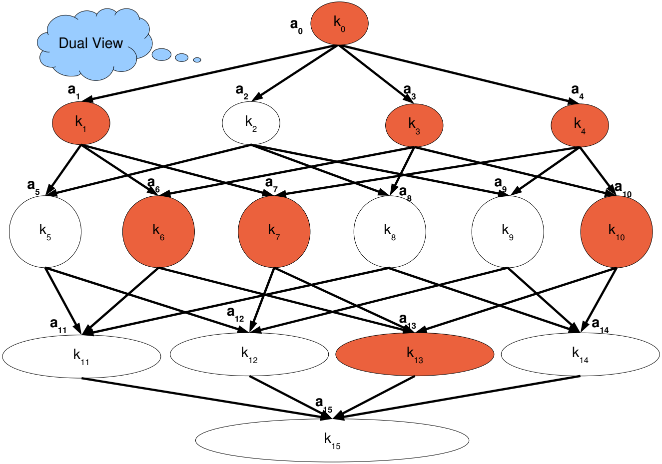
Hierarchical Kernel Learning (HKL)(Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)



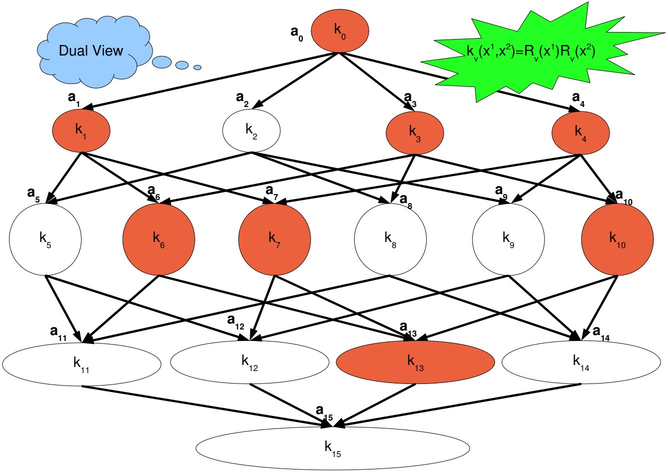
Hierarchical Kernel Learning (HKL) (Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)



Hierarchical Kernel Learning (HKL)(Bach, 08)

- Kernels arranged on DAG (lattice) are given
- Optimal combination of kernels (Multiple Kernel Learning)



HKL — Key Result

Active Set Algorithm:

- Complexity: Polynomial in number of selected kernels
- Condition: kernels are summable in *linear* time over a sub-lattice

HKL — Key Result

Active Set Algorithm:

- Complexity: Polynomial in number of selected kernels
- Condition: kernels are summable in *linear* time over a sub-lattice

Our case:

- Kernels indeed easily summable
  - R_v is nothing but product of few base proposition evaluations
  - Sum of exponential no. terms = Product of linear no. terms
  - E.g., $1 + R_1 + R_2 + R_1 R_2 = (1 + R_1)(1 + R_2)$
  - Our problem can be solved in reasonable time

Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL
TIC-TAC-TOE	0.652 ± 0.068	0.747 ± 0.026	0.633 ± 0.011	$\mathbf{0.889} \pm 0.029$
BALANCE	0.835 ± 0.034	0.856 ± 0.027	0.827 ± 0.013	$\mathbf{0.893} \pm 0.027$
HABERMAN	0.512 ± 0.072	0.565 ± 0.066	0.424 ± 0.000	$\mathbf{0.594} \pm 0.056$
CAR	0.913 ± 0.033	0.895 ± 0.024	0.755 ± 0.028	$\mathbf{0.943} \pm 0.024$
BLOOD TRANS.	0.549 ± 0.092	0.559 ± 0.100	0.489 ± 0.054	$\mathbf{0.594} \pm 0.009$
CMC	0.632 ± 0.013	0.601 ± 0.041	0.644 ± 0.026	$\mathbf{0.656} \pm 0.014$

Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL
TIC-TAC-TOE	0.652 ± 0.068 (2.51)	0.747 ± 0.026 (2.35)	0.633 ± 0.011 (2.46)	$\mathbf{0.889} \pm 0.029$ (**1.85**)
BALANCE	0.835 ± 0.034 (2.18)	0.856 ± 0.027 (1.88)	0.827 ± 0.013 (1.99)	$\mathbf{0.893} \pm 0.027$ (**1.65**)
HABERMAN	0.512 ± 0.072 (1.68)	0.565 ± 0.066 (**1.14**)	0.424 ± 0.000 (1.87)	$\mathbf{0.594} \pm 0.056$ (1.27)
CAR	0.913 ± 0.033 (3.12)	0.895 ± 0.024 (2.27)	0.755 ± 0.028 (1.85)	$\mathbf{0.943} \pm 0.024$ (**1.78**)
BLOOD TRANS.	0.549 ± 0.092 (1.99)	0.559 ± 0.100 (**1.07**)	0.489 ± 0.054 (1.5)	$\mathbf{0.594} \pm 0.009$ (1.64)
CMC	0.632 ± 0.013 (2.41)	0.601 ± 0.041 (2.13)	0.644 ± 0.026 (2.65)	$\mathbf{0.656} \pm 0.014$ (**1.96**)

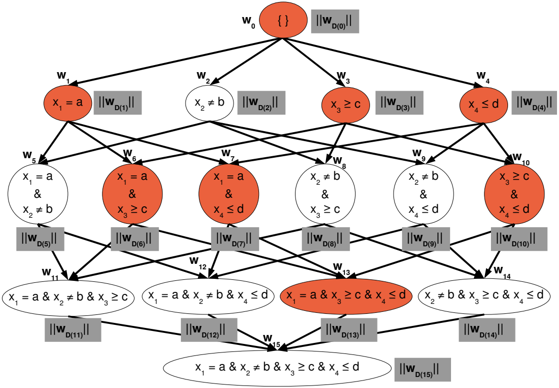
Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL
TIC-TAC-TOE	0.652 ± 0.068 (**40**, 2.51)	0.747 ± 0.026 (59, 2.35)	0.633 ± 0.011 (111, 2.46)	$\mathbf{0.889} \pm 0.029$ (129, **1.85**)
BALANCE	0.835 ± 0.034 (**17**, 2.18)	0.856 ± 0.027 (25, 1.88)	0.827 ± 0.013 (64, 1.99)	$\mathbf{0.893} \pm 0.027$ (65, **1.65**)
HABERMAN	0.512 ± 0.072 (**6**, 1.68)	0.565 ± 0.066 (8, **1.14**)	0.424 ± 0.000 (18, 1.87)	$\mathbf{0.594} \pm 0.056$ (32, 1.27)
CAR	0.913 ± 0.033 (**34**, 3.12)	0.895 ± 0.024 (141, 2.27)	0.755 ± 0.028 (80, 1.85)	$\mathbf{0.943} \pm 0.024$ (87, **1.78**)
BLOOD TRANS.	0.549 ± 0.092 (18, 1.99)	0.559 ± 0.100 (**6**, **1.07**)	0.489 ± 0.054 (58, 1.5)	$\mathbf{0.594} \pm 0.009$ (242, 1.64)
CMC	0.632 ± 0.013 (39, 2.41)	0.601 ± 0.041 (**13**, 2.13)	0.644 ± 0.026 (74, 2.65)	$\mathbf{0.656} \pm 0.014$ (127, **1.96**)

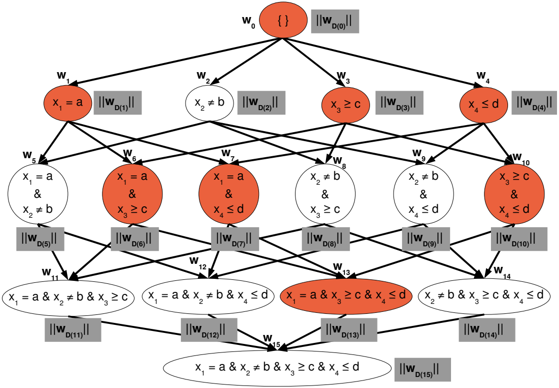
Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL
TIC-TAC-TOE	0.652 ± 0.068 (**40**, 2.51)	0.747 ± 0.026 (59, 2.35)	0.633 ± 0.011 (111, 2.46)	$\mathbf{0.889} \pm 0.029$ (129, **1.85**)
BALANCE	0.835 ± 0.034 (**17**, 2.18)	0.856 ± 0.027 (25, 1.88)	0.827 ± 0.013 (64, 1.99)	$\mathbf{0.893} \pm 0.027$ (65, **1.65**)
HABERMAN	0.512 ± 0.072 (**6**, 1.68)	0.565 ± 0.066 (8, **1.14**)	0.424 ± 0.000 (18, 1.87)	$\mathbf{0.594} \pm 0.056$ (32, 1.27)
CAR	0.913 ± 0.033 (**34**, 3.12)	0.895 ± 0.024 (141, 2.27)	0.755 ± 0.028 (80, 1.85)	$\mathbf{0.943} \pm 0.024$ (87, **1.78**)
BLOOD TRANS.	0.549 ± 0.092 (18, 1.99)	0.559 ± 0.100 (**6**, **1.07**)	0.489 ± 0.054 (58, 1.5)	$\mathbf{0.594} \pm 0.009$ (242, 1.64)
CMC	0.632 ± 0.013 (39, 2.41)	0.601 ± 0.041 (**13**, 2.13)	0.644 ± 0.026 (74, 2.65)	$\mathbf{0.656} \pm 0.014$ (217, **1.96**)

HKL — Introspection

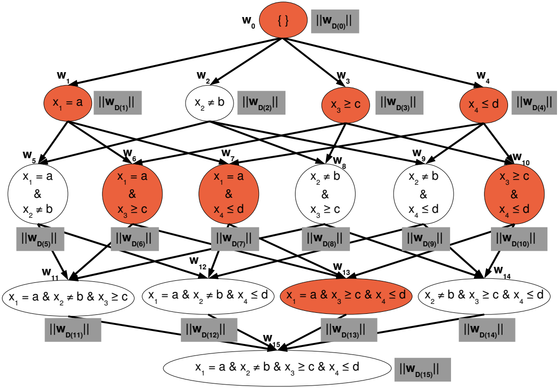


HKL — Introspection



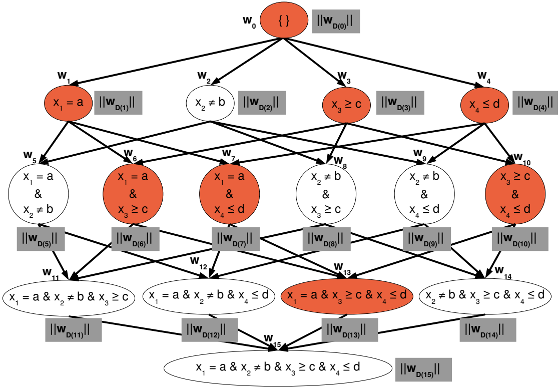
- Node selected only if all its ancesters are!

HKL — Introspection



- Node selected only if all its ancesters are!
- l_1 promotes sparsity.
- l_2 promotes non-sparsity. Employ sparsity inducing norm!

HKL — Introspection



- Node selected only if all its ancesters are!
- l_1 promotes sparsity.
- l_2 promotes non-sparsity. Employ sparsity inducing norm!

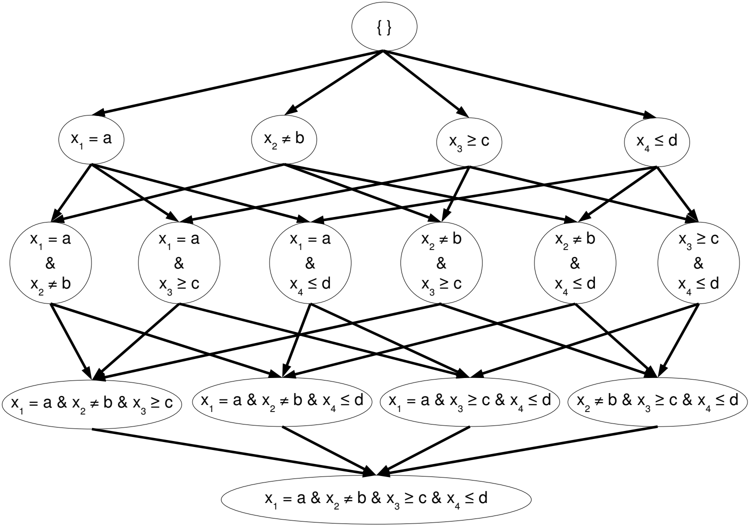
Proposed Formulation

Generalized HKL

$$\min_{\mathbf{w},b} \frac{1}{2}\left(\sum_{v\in\mathcal{V}} d_v \|\mathbf{w}_{D(v)}\|_\rho\right)^2 + C\sum_{i=1}^{m} L\left(y^i, \sum_{v\in\mathcal{V}} w_v R_v(\mathbf{x}^i) - b\right)$$

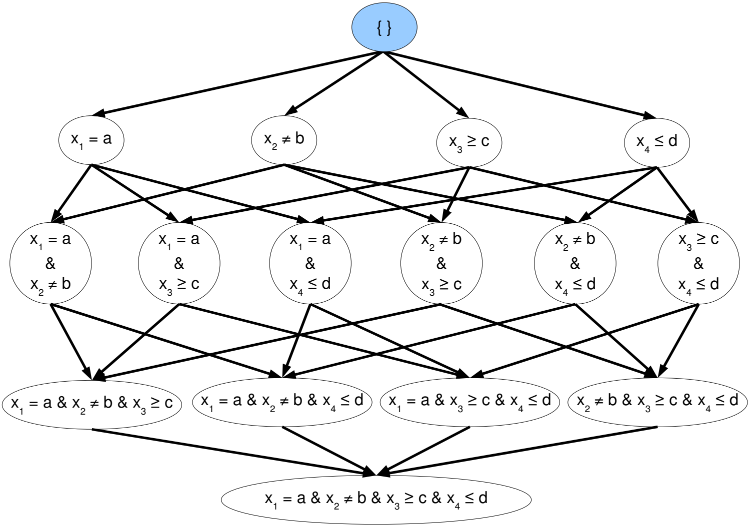
where $1 < \rho \leq 2$.

Active Set Method



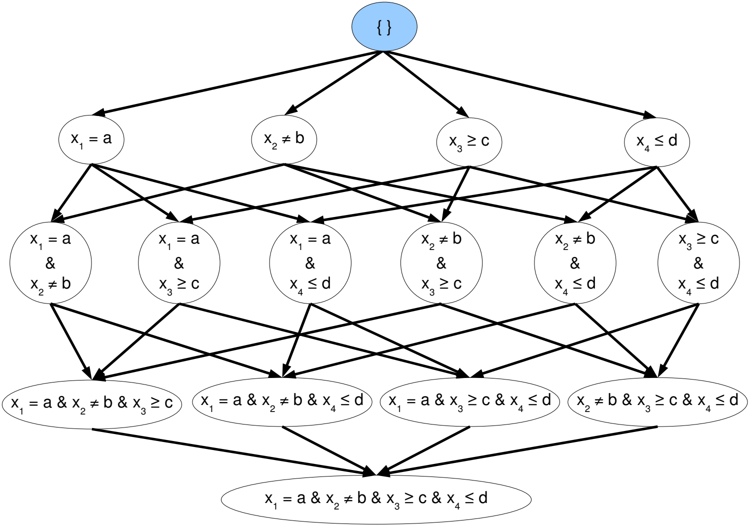
Active Set Method

Initialize active set with root node ($\mathcal{W} = \{0\}$).



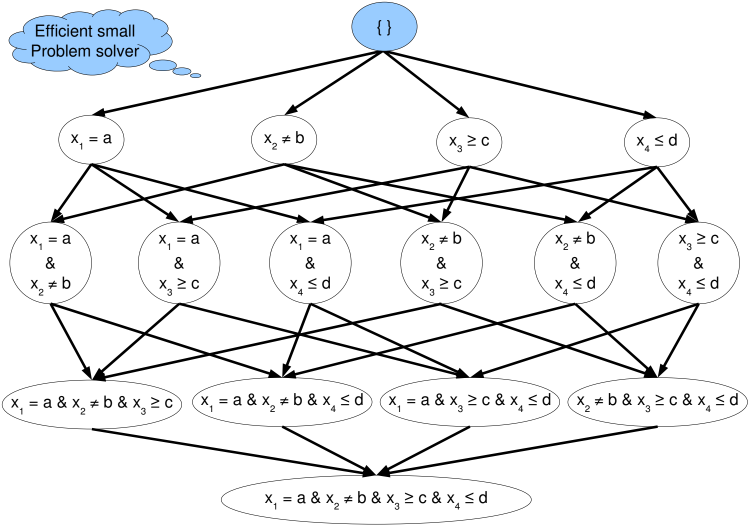
Active Set Method

Solve small problem



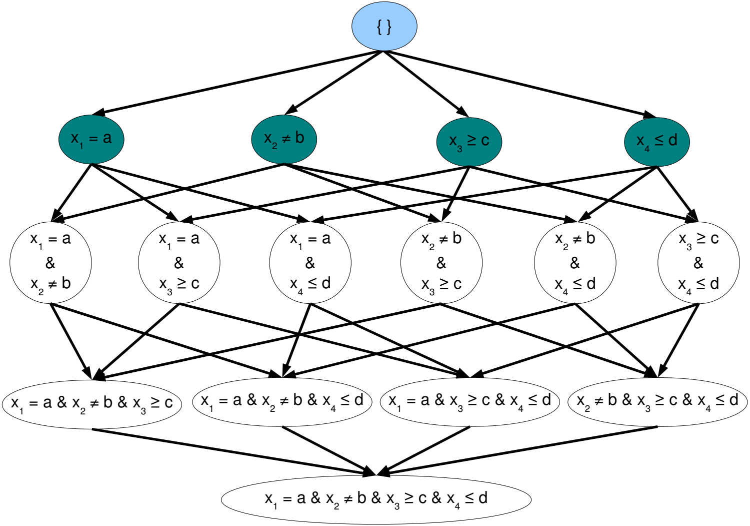
Active Set Method

Solve small problem



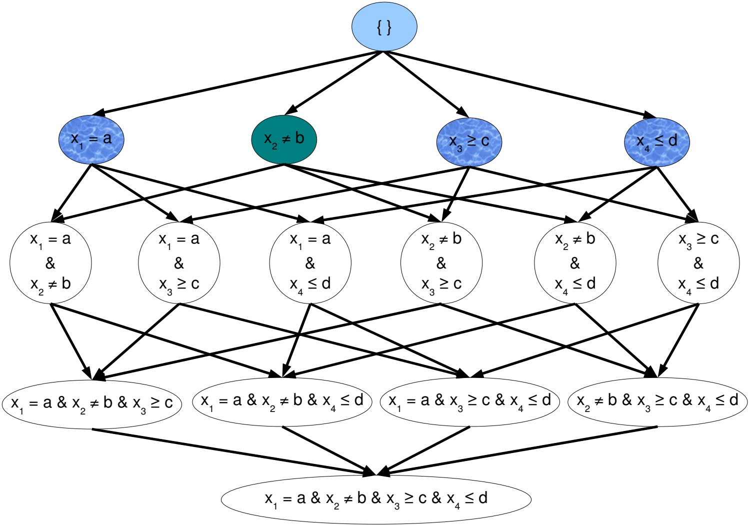
Active Set Method

Identify potential active set entries (i.e., $sources(\mathcal{W}^c)$)



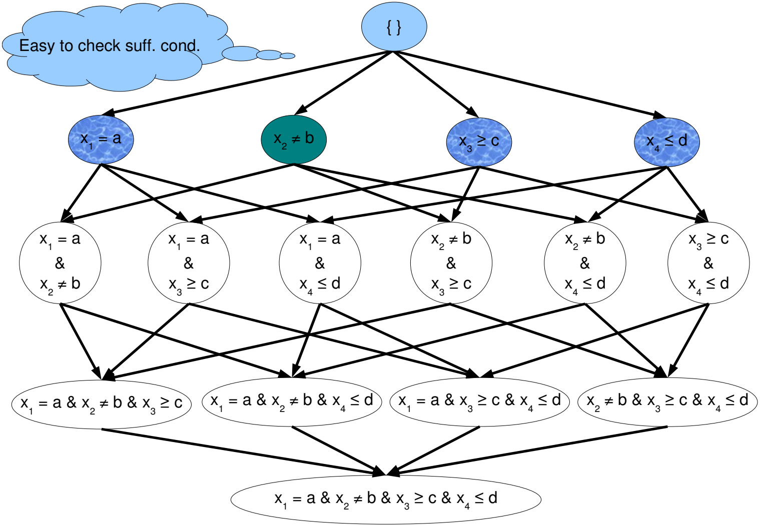
Active Set Method

Among them, optimality condition violators



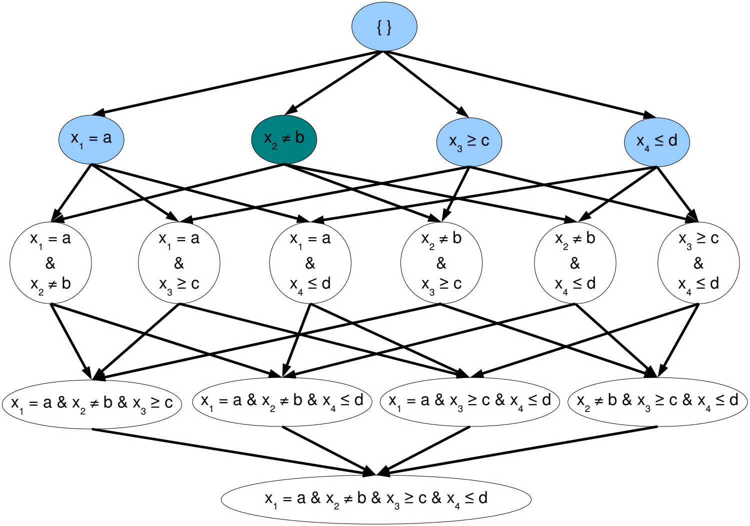
Active Set Method

Among them, optimality condition violators



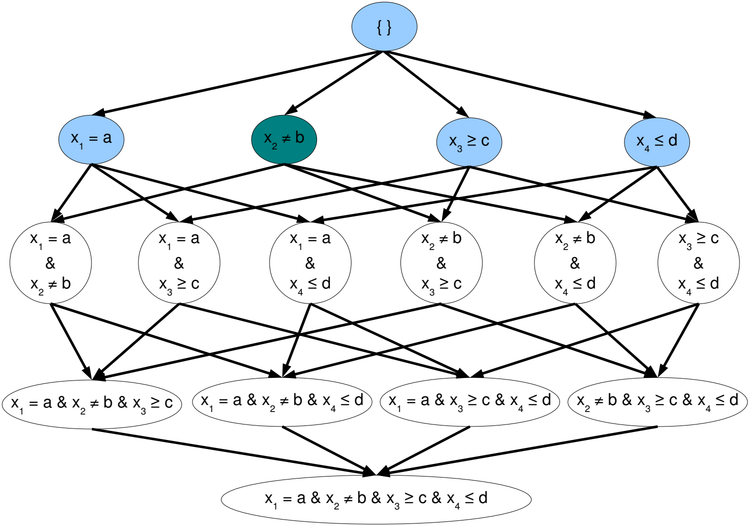
Active Set Method

Append them to active set ($\mathcal{W} = \{0, 1, 3, 4\}$).



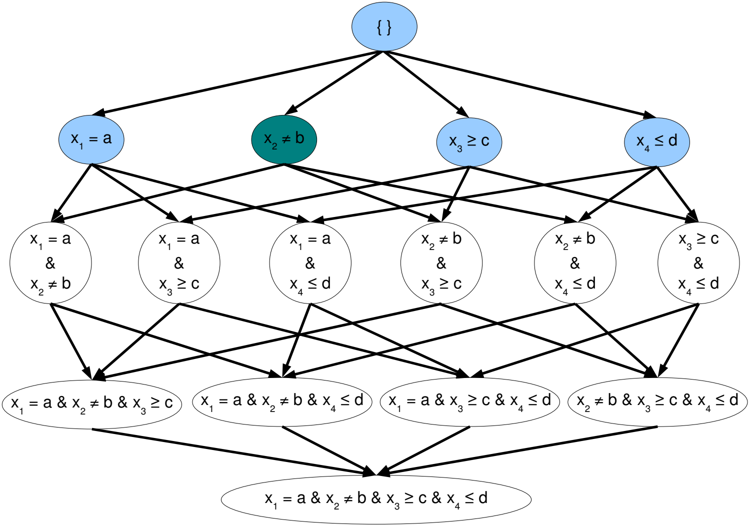
Active Set Method

Append them to active set ($\mathcal{W} = \{0, 1, 3, 4\}$). (repeat until suff. cond. satisfied)



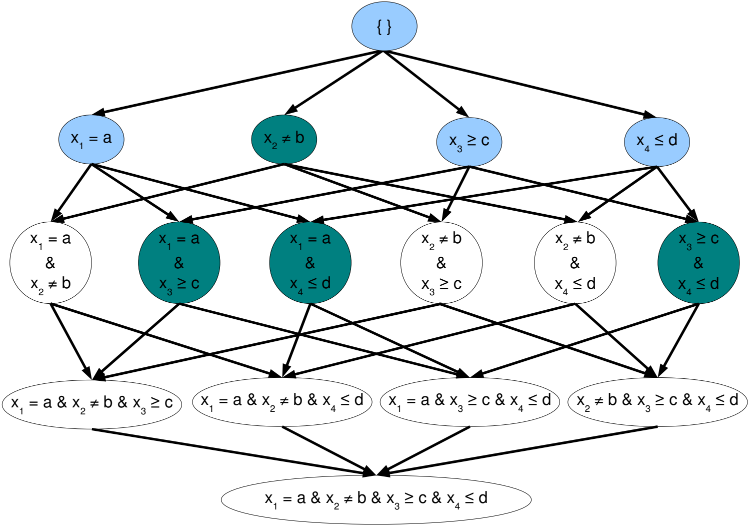
Active Set Method

Solve small problem



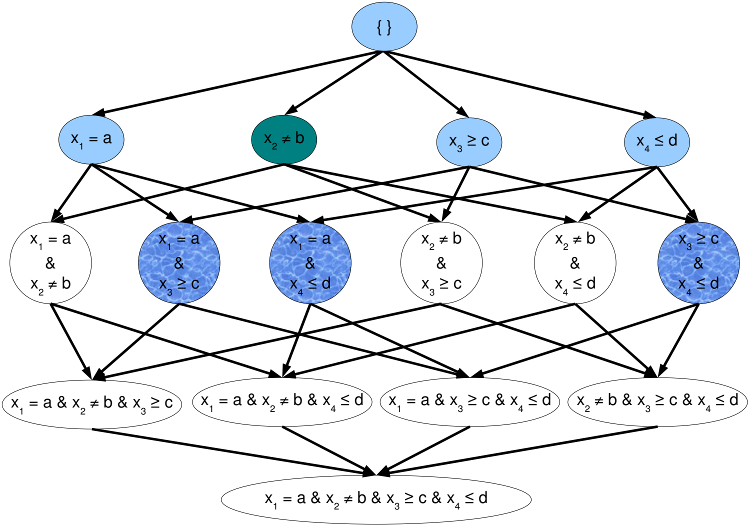
Active Set Method

Identify potential active set entries (i.e., $sources(\mathcal{W}^c)$)



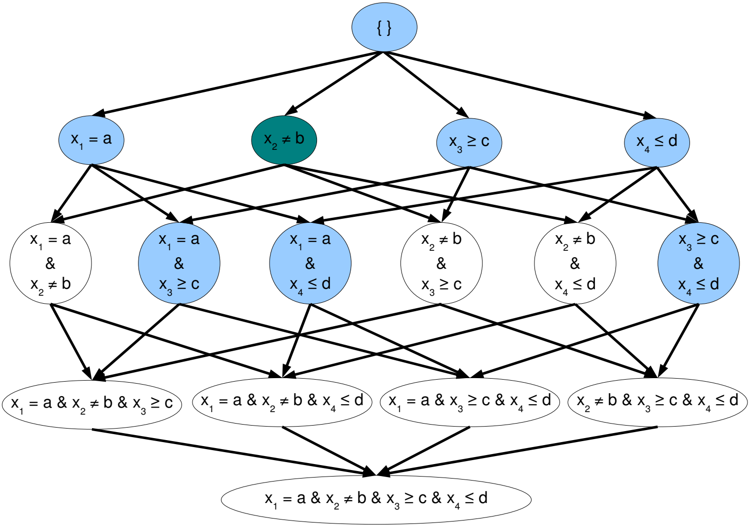
Active Set Method

Among them, optimality condition violators



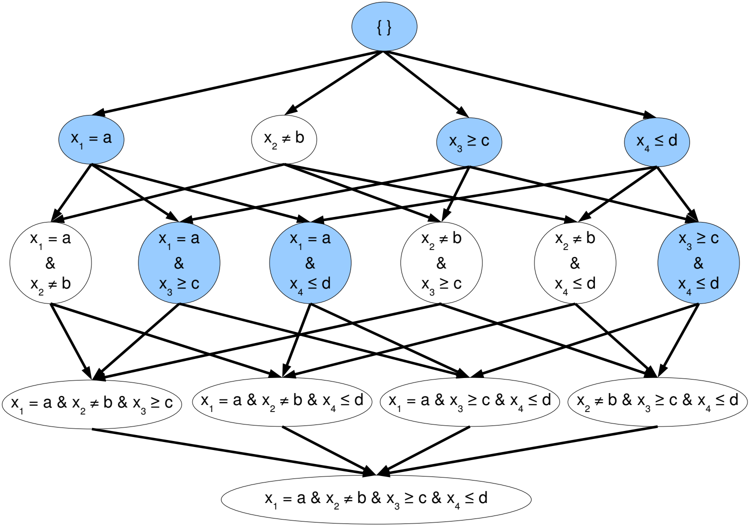
Active Set Method

Append them to active set ($\mathcal{W} = \{0, 1, 3, 4, 6, 7, 10\}$)



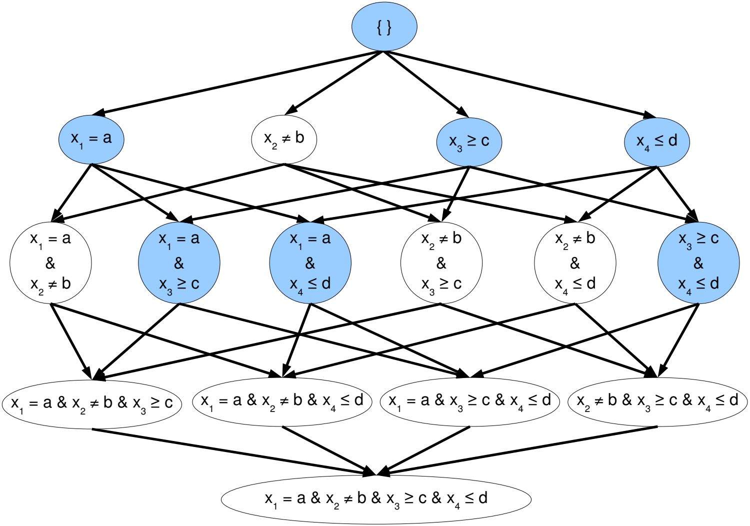
Active Set Method

Final active set: $\mathcal{W} = \{0, 1, 3, 4, 6, 7, 10\}$



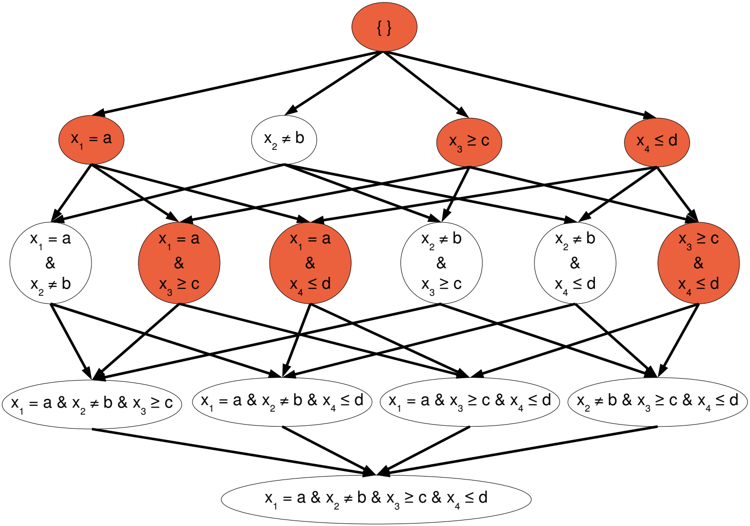
Active Set Method

Final active set: $\mathcal{W} = \{0, 1, 3, 4, 6, 7, 10\}$ (Complexity: Polynomial in active set size)



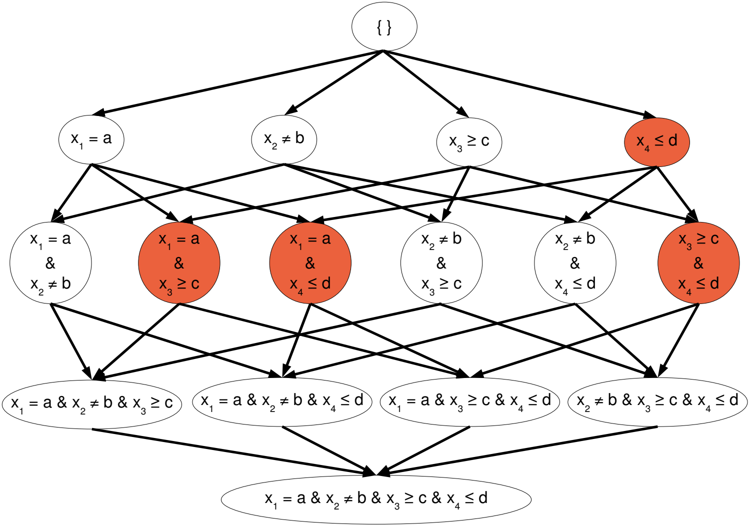
Active Set Method

Solution with HKL



Active Set Method

Key difference from HKL: Node selected without its ancestor!



Key Technical Result

Theorem

A highly specialized partial dual of generalized HKL is:

$$\begin{aligned} \min_{\eta \in \mathcal{R}^{|\mathcal{V}|}} \quad & g(\eta) \\ \text{s.t.} \quad & \eta \geq 0,\ \textstyle\sum_{v \in \mathcal{V}} \eta_v = 1 \end{aligned}$$

Key Technical Result

Theorem

A highly specialized partial dual of generalized HKL is:

$$\min_{\eta \in \mathcal{R}^{|\mathcal{V}|}} \quad g(\eta)$$
$$\text{s.t.} \quad \eta \geq 0,\ \sum_{v\in\mathcal{V}} \eta_v = 1$$

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

$$\max_{\alpha\in\mathcal{R}^m} \quad \sum_{i=1}^m \alpha_i - \frac{1}{2}\left(\sum_{v\in\mathcal{V}} \zeta_v(\eta)\left(\alpha^\top \mathbf{K}_v \alpha\right)^{\bar{\rho}}\right)^{\frac{1}{\bar{\rho}}} \text{ s.t. } 0\leq\alpha_i\leq C,\ \sum_{i=1}^m \alpha_i y^i = 0.$$

where $\zeta_v(\eta) = \left(\sum_{u\in A(v)} d_u^\rho \eta_u^{1-\rho}\right)^{\frac{1}{1-\rho}}$, $\bar{\rho} = \frac{\rho}{2(\rho-1)}$ and $\mathbf{K}_v$ is matrix with entries: $y^i y^j k_v(\mathbf{x}^i, \mathbf{x}^j)$.

Solving small problem

- Dual is min. of convex, Lipschitz conts., sub-differential objective over a simplex.
- Mirror-descent — highly scalable alg. for such problems.
- Sub-gradient — solve l_p-MKL (Vishwanathan et.al., 10).

Key Technical Result

Theorem

Suppose the active set $\mathcal{W}$ is such that $\mathcal{W} = A(\mathcal{W})$. Let the reduced solution with this $\mathcal{W}$ be $(\mathbf{w}_{\mathcal{W}}, b_{\mathcal{W}})$ and the corresponding dual variables be $(\eta_{\mathcal{W}}, \alpha_{\mathcal{W}})$. Then the reduced solution is a solution to the full problem with a duality gap less than ϵ if:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^{\top} \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

where $\epsilon_{\mathcal{W}}$ is a duality gap term associated with the computation of the reduced solution.

Complexity: Polynomial in size of $\mathcal{W}$?

Sufficiency Condition:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^{\top} \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

Complexity: Polynomial in size of $\mathcal{W}$?

Sufficiency Condition:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

- $\rho \to 1$ ($\bar{\rho} \to \infty$), suff. cond. tight

Complexity: Polynomial in size of $\mathcal{W}$?

Sufficiency Condition:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

- $\rho \rightarrow 1$ ($\bar{\rho} \rightarrow \infty$), suff. cond. tight
- $\rho = 2$ ($\bar{\rho} = 1$), suff. cond. loose; computationally feasible

Complexity: Polynomial in size of $\mathcal{W}$?

Sufficiency Condition:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^{\top} \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

- $\rho \to 1$ ($\bar{\rho} \to \infty$), suff. cond. tight
- $\rho = 2$ ($\bar{\rho} = 1$), suff. cond. loose; computationally feasible
- How much ground lost by replacing l_∞ with l_1 ?

Complexity: Polynomial in size of $\mathcal{W}$?

Sufficiency Condition:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right)^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}} \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

- $\rho \to 1$ ($\bar{\rho} \to \infty$), suff. cond. tight
- $\rho = 2$ ($\bar{\rho} = 1$), suff. cond. loose; computationally feasible
- How much ground lost by replacing l_∞ with l_1 ?
  - Not much: As kernels near bottom are extremely sparse!

Complexity: Polynomial in size of $\mathcal{W}$?

Final Sufficiency Condition:

$$\max_{t \in sources(\mathcal{W}^c)} \left(\sum_{v \in D(t)} \left(\frac{\alpha_{\mathcal{W}}^\top \mathbf{K}_v \alpha_{\mathcal{W}}}{\left(\sum_{u \in A(v) \cap D(t)} d_u \right)^2} \right) \right) \leq (\Omega(\mathbf{w}_{\mathcal{W}}))^2 + 2(\epsilon - \epsilon_{\mathcal{W}})$$

- $\rho \to 1$ ($\bar{\rho} \to \infty$), suff. cond. tight
- $\rho = 2$ ($\bar{\rho} = 1$), suff. cond. loose; computationally feasible
- How much ground lost by replacing l_∞ with l_1 ?
  - Not much: As kernels near bottom are extremely sparse!

Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL	$\text{HKL}_{\rho=1.1}$
TIC-TAC-TOE	0.652 ± 0.068 (**40**, 2.51)	0.747 ± 0.026 (59, 2.35)	0.633 ± 0.011 (111, 2.46)	0.889 ± 0.029 (129, 1.85)	$\mathbf{0.935} \pm 0.043$ (79, **1.77**)
BLOOD TRANS.	0.549 ± 0.092 (18, 1.99)	0.559 ± 0.100 (**6**, **1.07**)	0.489 ± 0.054 (58, 1.5)	$\mathbf{0.594} \pm 0.009$ (242, 1.64)	0.593 ± 0.011 (7,1.40)
BALANCE	0.835 ± 0.034 (**17**, 2.18)	0.856 ± 0.027 (25, 1.88)	0.827 ± 0.013 (64, 1.99)	0.893 ± 0.027 (65, 1.65)	$\mathbf{0.899} \pm 0.023$ (28,**1.23**)
HABERMAN	0.512 ± 0.072 (**6**, 1.68)	0.565 ± 0.066 (8, **1.14**)	0.424 ± 0.000 (18, 1.87)	$\mathbf{0.594} \pm 0.056$ (32, 1.27)	$\mathbf{0.594} \pm 0.056$ (12,1.20)
CAR	0.913 ± 0.033 (**34**, 3.12)	0.895 ± 0.024 (141, 2.27)	0.755 ± 0.028 (80, 1.85)	$\mathbf{0.943} \pm 0.024$ (87, 1.78)	0.935 ± 0.036 (50,**1.68**)
CMC	0.632 ± 0.013 (39, 2.41)	0.601 ± 0.041 (**13**, 2.13)	0.644 ± 0.026 (74, 2.65)	0.656 ± 0.014 (127, 1.96)	$\mathbf{0.659} \pm 0.008$ (43,**1.70**)

Summary

- Applied HKL to rule ensemble learning
  - Improved generalization
  - Bridged gap between kernel and rule learning communities

Summary

- Applied HKL to rule ensemble learning
  - Improved generalization
  - Bridged gap between kernel and rule learning communities
- Generalized HKL
  - Generalizes well while learning compact ruleset
  - Sometimes 25% improvement in generalization
  - Applicable elsewhere

Summary

- Applied HKL to rule ensemble learning
  - Improved generalization
  - Bridged gap between kernel and rule learning communities
- Generalized HKL
  - Generalizes well while learning compact ruleset
  - Sometimes 25% improvement in generalization
  - Applicable elsewhere
- Efficient mirror-descent based active set method
  - Complexity: polynomial in active set size ($\ll O(2^n)$)
  - Searched rule space size $\sim 2^{50}$ in $\sim$ 10 min.

Rule Ensemble Learning using Hierarchical Kernels framework for Structured Output Spaces.

REL-HKL on structured output spaces

- Output is a structure.
- SVM maximizes the margin of true output with all possible outputs in output space.
- HMM is a structured output problem (which we explore in this work).

SVM for structured output spaces

Notations

- $\mathcal{X}$: input sequence space, $\mathcal{Y}$: output sequence space.
- $X_i \in \mathcal{X}$: an instance of input sequence.
- $Y_i \in \mathcal{Y}$: an instance of output sequence[2].
- $\mathbf{x}_i^p$: joint state of feature values at p^{th} position of the i^{th} example.
- y_i^p: output at p^{th} position of the i^{th} example.
- ψ: feature vector.
- $\mathbf{f}$: feature weights vector.

[2]Subscript i is used to denote i^{th} example sequence and should not be confused with the i^{th} element of a vector.

SVM for structured output spaces [4]

- Generalize multiclass support vector machine learning.
- Features constructed from input and output variables.
- In case of HMM, features constructed from emission and transition distribution.

Define discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, such that, $F(X, Y; \mathbf{f}) = \langle \mathbf{f}, \psi(X, Y) \rangle$ [3]

and prediction is given by

$$\hat{Y} = \mathcal{F}(X; \mathbf{f}) = \arg\max_{Y \in \mathcal{Y}} F(X, Y; \mathbf{f})$$

Loss function for HMM

- Predicted sequences that deviate more from the actual should be penalized more.
- Loss function, $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. $\Delta(Y, \hat{Y})$ is the loss value when the true output is Y and the prediction is $\hat{Y}$.

---

[3] $F(X, Y; \mathbf{f})$ represents a score which is a scalar value based on the features ψ involving input sequence X and output sequence Y values and parameterised by a parameter vector $\mathbf{f}$.

[4] [Tsochantaridis et. al.,2004,2006]

SVM for structured output spaces

SVM formulation for structured output spaces (HMM)

SVM_0:

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

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

- C is the regularization parameter.
- ξ s are the slack variables introduced to allow errors in the training set in a soft margin SVM.
- $\langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y) \rangle = \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y_i) \rangle - \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y) \rangle$.

When the sequence length is large, the number of constraints in SVM_0 can be extremely large. A cutting plane method can be used to find a polynomially sized subset of constraints that ensures a solution very near to the optimum [Tsochantarids et. al.].

SVM for structured output spaces: Remarks

To learn optimum structure and parameters of HMM (structSVM)

1. Modify StructSVM to include features that can be ordered in the form of a lattice.
2. Include the ρ-norm regularizer (as 'in RELHKL) for emission features and 2-norm for transition features.
3. Derive a dual for the new formulation that can be computed efficiently.
4. Derive a sufficiency condition to stop the active set algorithm.

REL-HKL on structured output spaces for learning optimum HMM model

Notations

- ψ: feature vector containing emission and transition features.
- ψ_E: part of ψ corresponding to emission features.
- ψ_T: part of ψ corresponding to transition features.[5]
- $\mathbf{f}$: feature weights vector.
- $\mathbf{f_E}$: feature weights vector corresponding to emission.
- $\mathbf{f_T}$: feature weights vector corresponding to transition.
- $\mathcal{V}$: indices of the elements of ψ.
- $\mathcal{V}_\mathbf{E}$: indices corresponding to emission elements.
- $\mathcal{V}_\mathbf{T}$: indices corresponding to transition elements.

[5] For convenience we assume ψ_E and ψ_T as two vectors of dimension same as ψ, but with non zero elements to features only on their context.

REL-HKL on structured output spaces for learning optimum HMM model

- Regularizer used in RELHKL is for the features obeying lattice structure.
- Also that We are not interested in learning sparse transition features.
- Therefore, We separate the regularizer into two, viz, emission and transition.

SVM formulation after separating the regularizer.

SVM_1:

$$\min_{\mathbf{f},\boldsymbol{\xi}} \frac{1}{2}\Omega_E(\mathbf{f_E})^2 \;+\; \frac{1}{2}\Omega_T(\mathbf{f_T})^2 \;+\; \frac{C}{m}\sum_{i=1}^{m}\xi_i,$$

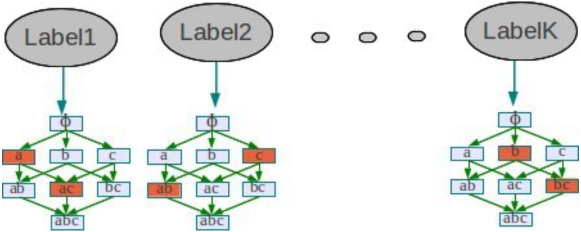
$$\forall i, \forall Y \in \mathcal{Y} \setminus Y_i : \langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y)\rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)}$$

$$\forall i : \xi_i \geq 0$$

- $\Omega_E(\mathbf{f_E}) = \sum_{v \in \mathcal{V}_\mathbf{E}} d_v \parallel \mathbf{f}_{\mathbf{E}D(v)} \parallel_\rho, \quad \rho \in (1, 2]$
- $\Omega_T(\mathbf{f_T}) = \left(\sum_i f_{Ti}^2\right)^{\frac{1}{2}}$
- $\langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y)\rangle = \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y_i)\rangle - \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y)\rangle$

REL-HKL on structured output spaces for learning optimum HMM model

- At optimality, most of the emission feature weights are expected to be zero [Ganesh et. al.,2011].
- Therefore an active set algorithm can be employed to solve efficiently.
- In each iteration, a subset of featues ($\mathcal{W}$) is considered to be active.



REL-HKL on structured output spaces for learning optimum HMM model

SVM formulation considering only the featues in $\mathcal{W}$ (reduced problem),

SVM_2

$$\min_{\mathbf{f},\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{v\in\mathcal{W}} d_v \parallel \mathbf{f}_{\mathbf{E}D(v)\bigcap\mathcal{W}} \parallel_\rho\right)^2 + \frac{1}{2} \parallel \mathbf{f_T} \parallel_2^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i,$$

$$\forall i, \forall Y \in \mathcal{Y} \setminus Y_i :$$

$$-\left(\sum_{v\in\mathcal{W}} \langle f_{Ev}, \psi^\delta_{Evi}(Y)\rangle + \sum_{v\in\mathcal{V}_\mathbf{T}} \langle f_{Tv}, \psi^\delta_{Tvi}(Y)\rangle + \frac{\xi_i}{\Delta(Y_i, Y)} - 1\right) \leq 0$$

$$\forall i : \; -\xi_i \leq 0$$

REL-HKL on structured output spaces for learning optimum HMM model

Applying variational characterization[6] on $\Omega_E(\mathbf{f_E})^2$

Partial dual (wrt. $\mathbf{f}, \boldsymbol{\xi}$) of SVM_1

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

where

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

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

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

---

[6] Micchelli&Pontil,2005,Bach,2009,Jawanpuria et.al.,2011

REL-HKL on structured output spaces for learning optimum HMM model

Sufficiency condition for the reduced solution to have a duality gap less than ϵ

$$\max_{u \in sources(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \boldsymbol{\alpha}_{\mathcal{W}iY}^{\top} \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \Bigg(\prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{b^2} \Bigg)$$

$$\Bigg(\prod_{k \notin u} \Big(1 + \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{(1+b)^2} \Big) \Bigg) \boldsymbol{\alpha}_{\mathcal{W}jY'} \leq \boldsymbol{\Omega}_E(\mathbf{f}_{\mathbf{E}\mathcal{W}})^2 + \boldsymbol{\Omega}_T(\mathbf{f}_{\mathbf{T}\mathcal{W}})^2 + 2(\epsilon - e_{\mathcal{W}})$$

where $e_{\mathcal{W}} = \boldsymbol{\Omega}_E(\mathbf{f}_{\mathbf{E}\mathcal{W}})^2 + \boldsymbol{\Omega}_T(\mathbf{f}_{\mathbf{T}\mathcal{W}})^2 + \frac{C}{m} \sum_i \xi_i + \frac{1}{2} \boldsymbol{\alpha}_{\mathcal{W}}^{\top} \boldsymbol{\kappa}_{\mathbf{T}} \boldsymbol{\alpha}_{\mathcal{W}} - \sum_{i, Y \neq Y_i} \boldsymbol{\alpha}_{\mathcal{W}iY}$.

REL-HKL on structured output spaces for learning optimum HMM model

Final dual

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

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

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

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

- Equation (1) is solved using mirror descent algorithm.
- For a given η, equation (2) is solved using a cutting plane algorithm.

For mirror descent algorithm, the i^{th} subgradient is computed using,

$$(\nabla g(\eta))_i = -\frac{d_i^\rho \eta_i^{-\rho}}{2\hat{\rho}} \left(\sum \zeta_w(\eta) (\bar{\boldsymbol{\alpha}}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\boldsymbol{\alpha}})^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}-1} \left(\sum \zeta_w(\eta)^\rho (\bar{\boldsymbol{\alpha}}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\boldsymbol{\alpha}})^{\hat{\rho}} \right)$$

REL-HKL on structured output spaces for learning optimum HMM model

Active set algorithm

Input: Training data D, Oracle for computing kernels, Maximum tolerance ϵ
1. Initialize $\mathcal{W} = Top\ nodes$ in the lattice as the active set
2. Compute η, α by solving (1) using mirror descent
3. **while** sufficiency condiiton is not satisfied, **do**
4. Add sufficiency condition violating nodes to active set $\mathcal{W}$
5. Recompute η, α by solving (1)
6. **end while**
7. **Output:** active-set W, η, α

Step 2 and 5 are solved as

- For a fixed η, an optimum α is computed by solving (2) using cutting plane algorithm.
- Update η using the gradient computed using the obtained α.
- Repeat above two steps until convergence.

Cutting plane algorithm

Input: kernels, C, ϵ_{margin} (allowed violation of margin)

$S_i \leftarrow \phi \quad \forall i = 1, ..., m$
repeat
    for $i = 1, ..., m$ **do**
        $\forall Y : H(Y)$ is computed using (3).
        compute $\hat{Y} = \arg\max_{Y} H(Y)$.
        compute $\xi_i = \max\{0, \max_{Y \in S_i} H(Y)\}$.
        if $H(\hat{Y}) > \xi_i + \epsilon_{margin}$, **then**
            $S_i \leftarrow S_i \bigcup \{\hat{Y}\}$.
            compute $\boldsymbol{\alpha}$ using $S = \bigcup_i S_i$ in (2).
        end if
    end for
until no S_i has changed during the iteration.

where cost for boundary violation,

$$H(Y) \equiv \left[1 - \langle \mathbf{f}, \psi_i^{\delta}(Y) \rangle \right] \Delta(Y_i, Y) \quad (3)$$

RELHKL on Structured Output Spaces Results

Dataset [7]	Std HMM		Greedy feature induction		RELHKL on StructSVM [8]	
	Timeslice	class	Timeslice	class	Timeslice	class [9]
`Raw`	25.4	21.75	26.88	21.33	63.96	32.01
`Change`	23.64	25.99	44.39	31.42	56.74	33.85
`Last`	51.83	38.56	49.74	27.76	92.57	53.91
`Change + Last`	37.86	30.12	37.29	27.67	94.47	55.82

[7] Activity recognition dataset, Kasteren et. al.

[8] Greedy feature induction and RELHKL on StructSVM consider positive conjunctions

[9] Timeslice accuracy is percentage of time the prediction is correct. Class accuracy is the average percentage of time a class is predicted correctly

Hierarchical Kernel learning For Propositional Features

Applications of Hierarchical Kernel learning For Propositional Features

Learning rule ensembles

- Conjunctive propositional features ✓ [6]
- Disjunctive propositional features

Disjunctive propositional features

- Since HKL follows a top-down approach $\rightarrow$ descendant norm is more suitable
- Top node in lattice is the most general, i.e. disjunction of all basic features $\bigvee_{n=1}^{N} \phi_n$
- descendant of node is a more specialized node; got by removing one of the features of its parent.
- Only sufficiency condition changes; everything else remains same.

Disjunctive propositional features

- feature map $\phi(x)$ as $(1 - \overline{\phi}(x))$ ($\overline{\phi}(x)$ is boolean complement of $\phi(x)$).
- a disjunctive feature corresponding to $\bigvee_{n=1}^{N} \phi_n(x_i) = (1 - \Pi_{n=1}^{N} \overline{\phi_n}(x_i))$
- kernel corresponding to the disjunctive feature is $(1 - \Pi_{n=1}^{N} \overline{\phi_n}(x_i))(1 - \Pi_{n=1}^{N} \overline{\phi_n}(x_j))$

Hierarchical Kernel learning For Disjunctive Features

- sum of exponential kernels of the entire lattice:
  $\sum_{v \in V} K_v(x_i, x_j) =$
  $1 + 2^N + \Pi_{n=1}^{N}(1 + \overline{\phi}_n(x_i)\overline{\phi}_n(x_j)) - \prod_n (1 + \overline{\phi}_n(x_i)) - \prod_n (1 + \overline{\phi}_n(x_j))$

Hierarchical Kernel learning For Disjunctive Features

- sufficiency condition:
  $$\max_{t \in sources(W^C)} \sum_{i,j} \alpha_{Wi} Q(t)_{ij} \alpha_{Wj} \leq \Omega_s(f)^2 + \epsilon$$
  where $Q(t)_{ij} =$
  $$\frac{1}{(1+b)^{2|t|}}\left(\left(1 + \left(\frac{1+b}{b}\right)^2\right)^{|t|} - \prod_{k \in t}\left(1 + \frac{\overline{\phi}_k(x_i)}{(\frac{b}{b+1})^2}\right) - \prod_{k \in t}\left(1 + \frac{\overline{\phi}_k(x_j)}{(\frac{b}{b+1})^2}\right)\right.$$
  $$\left. + \prod_{k \in t}\left(1 + \frac{\overline{\phi}_k(x_i)}{\frac{b}{1+b}} \frac{\overline{\phi}_k(x_j)}{\frac{b}{1+b}}\right)\right)$$

Hierarchical Kernel learning For Learning Taxonomies

- Inherent hierarchical structure exploited
- Vocabulary consisiting of important sense tagged words
- Every sense of every word becomes a basic feature of HKL
- Syntagmatic Information (context-co-occurrence): conjunctive lattice
- Paradigmatic Information (synonymous words): disjunctive lattice

Conclusion

- Hierarchical Kernel Learning: Large features are discarded.
- Rule Ensemble Learning using Hierarchical Kernels: Large features are discarded and sparcity among small features selected.
- REL-HKL framework in structured output spaces
- Hierarchical Kernel Learning for disjunctive features.

Bibliography

- Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, Matthai Philipose: Common sense based joint training of human activity recognizers. In: 20th International Joint Conference on Artifical Intelligence (2007)
- John Lafferty, Andrew McCallum, Fernando Pereira: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: International Conference on Machine Learning (2001)
- Niels Landwehr, Andrea Passerini, Luc De Raedt, Paolo Frasconi: KFOIL: Learning Simple Relational Kernels. In: 21st National Conference on Artificial Intelligence (2006)
- Bernd Gutmann, Kristian Kersting: TildeCRF: Conditional Random Fields for Logical Sequences. In: 15th European Conference on Machine Learning (2006)
- N. Di Mauro, T.M.A. Basile, S. Ferilli, F. Esposito: Feature Construction for Relational Sequence Learning. In: Technical Report, arXiv:1006.5188 (2010)
- Ashwin Srinivasan: The Aleph Manual. Technical Report, University of Oxford (2007)

Bibliography

- Niels Landwehr, Bernd Gutmann, Ingo Thon, Luc De Raedt, Matthai Philipose: Relational Transformation-based Tagging for Activity Recognition. Progress on Multi-Relational Data Mining 89(1):111-129 (2009)
- Henri Binsztok, Thierry Artieres, Patrick Gallinari: A model-based approach to sequence clustering. In: European Conference on Artificial Intelligence (2004)
- Andrew McCallum: Efficiently Inducing Features of Conditional Random Fields. In: Nineteenth Conference on Uncertainty in Artificial Intelligence (2003)
- S. Siegel: Nonparametric statistics for the behavioural sciences. New York: McGraw-Hill (1956)
- Naveen Nair, Ganesh Ramakrishnan and Shonali Krishnaswamy, *Enhancing Activity Recognition in Smart Homes Using Feature Induction*, International Conference on Data Warehousing and Knowledge Discovery, 2011.
- Pratik Jawanpuria, Saketha Nath Jagarlapudi and Ganesh Ramakrishnan, *Efficient Rule Ensemble Learning using Hierarchical Kernels*, International Conference on Machine Learning, 2011.
- Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims and Yasemin Altun, *Support Vector Machine Learning for Interdependent and Structured Output Spaces*, International Conference on Machine Learning, 2004.

Bibliography

- Tsochantaridis Ioannis, *Support vector machine learning for interdependent and structured output spaces*,2006.
- Charles Micchelli and Massimiliano Pontil, *Learning the Kernel Function via Regularization*, Journal of Machine Learning Research,2005.
- Daniel H. wilson, *Assistive Intelligent Environments for Automatic Health Monitoring*, PhD Thesis, Carnegie Mellon University, 2005.
- Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne and Ben krose, *Accurate activity recognition in a home setting*, 10th International conference on Ubiquitous computing, 2008.
- C.H.S. Gibson, T.L.M. van Kasteren and Ben Krose, *Monitoring Homes with Wireless Sensor Networks*, Proceedings of the International Med-e-Tel Conference, 2008.
- R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proceedings of the IEEE, 77(2):257–286, 1989.
- Lise Getoor and Ben Taskar, *Statistical Relational Learning*, MIT Press, 2006.
- Forney GD, *The viterbi algorithm*, Proceedings of IEEE, 61(3):268–278, 1973

Bibliography

Bach F., *High-Dimensional Non-Linear Variable Selection through Hierarchical Kernel Learning*, Technical report, INRIA, France, 2009.

Rakotomamonjy A., Bach F., Canu S., and Grandvalet Y., *SimpleMKL*, JMLR, 9:2491-2521, 2008.

Szafranski M., Grandvalet Y., and Rakotomanmonjy A., *Composite Kernel Learning*, ICML, 2008.

Kloft M., Brefeld U., Sonnenburg S., Laskov P., Muller K. R., and Zien A., *Efficient and Accurate p-Norm Multiple Kernel Learning*, NIPS, 2009.

Sion M., *On General Minimax Theorem*, Pacific Journal of Mathematics, 1958.

Thanks