Efficient Rule Ensemble Learning using Hierarchical Kernels

J. Saketha Nath
Collaboration: Pratik J. and Ganesh R.

Indian Institute of Technology — Bombay

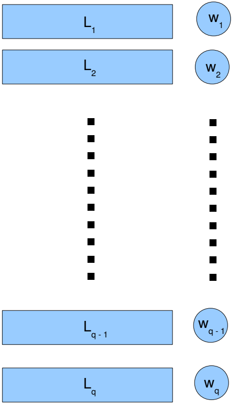
Google Talk

Rule Ensembles — Overview

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

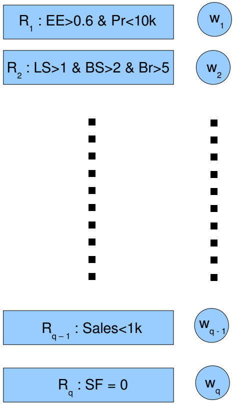
Rule Ensembles — Overview

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



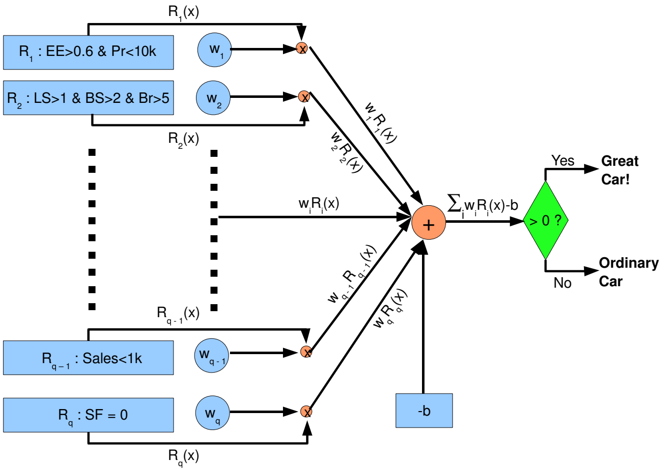
Rule Ensembles — Overview

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

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

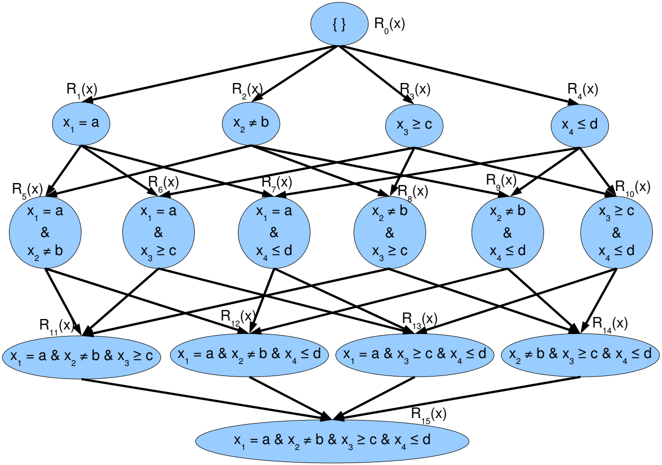
  Numeric e.g., $x_j \geq b$ and $x_j \leq b$

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 Reason for Efficiency:

(Large) sub-lattices with *long* rules are avoided

A Naive Formulation

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

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

A Naive Formulation

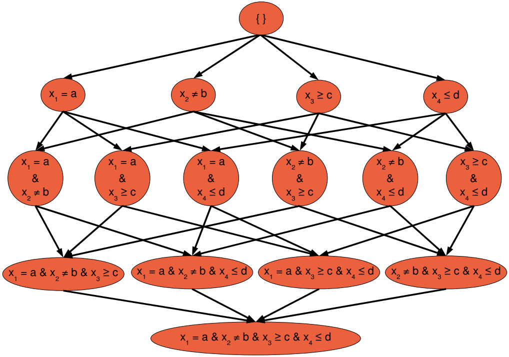
- Decision function[1]: sign $\left(\sum_{v\in\mathcal{V}} w_v R_v(\mathbf{x}) - b\right)$
- l_1 regularize to force many w_v to zero

l_1 regularized formulation:

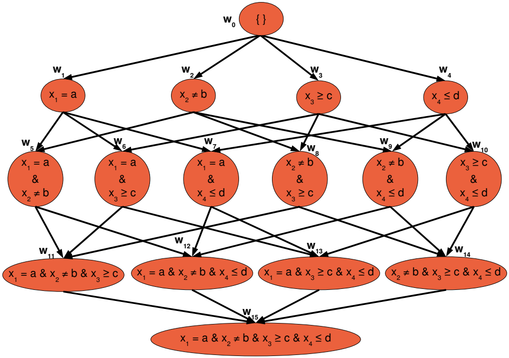
$$\min_{\mathbf{w},b} \frac{1}{2}\left(\sum_{v\in\mathcal{V}} |w_v|\right)^2 + C\sum_{i=1}^{m} L\left(y^i, \sum_{v\in\mathcal{V}} w_v R_v(\mathbf{x}^i) - b\right)$$

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

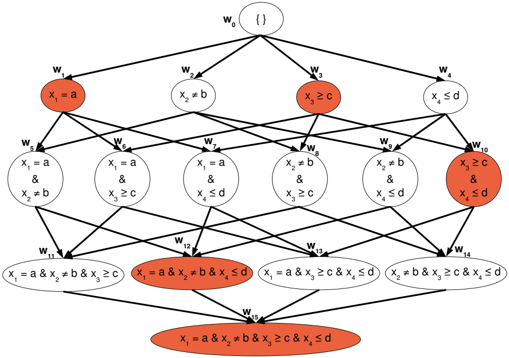
A Naive Formulation



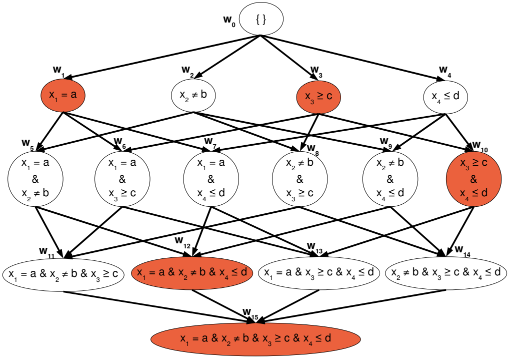
A Naive Formulation



A Naive Formulation



A Naive Formulation



Short-comings:

- long rules may be selected
- Computationally difficult problem

An Improved Formulation

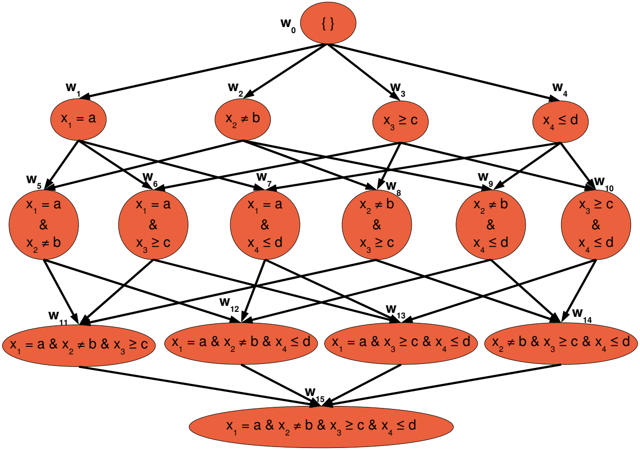
Key Idea:

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:

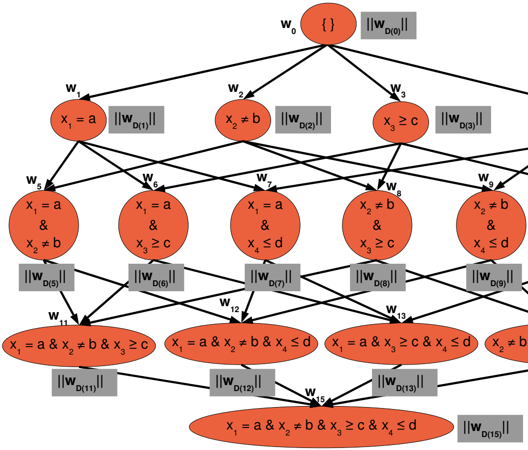
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

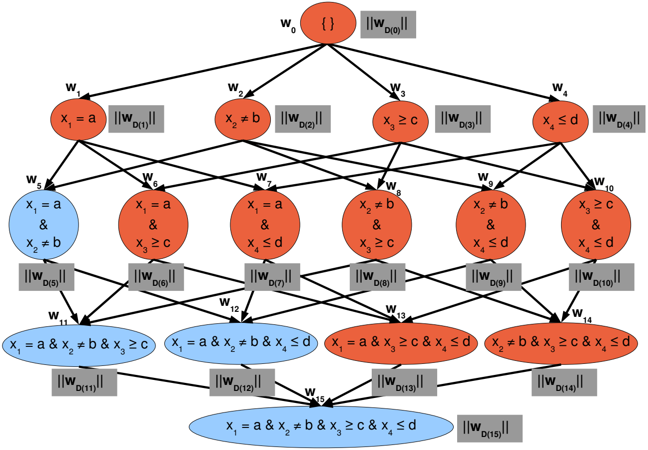
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

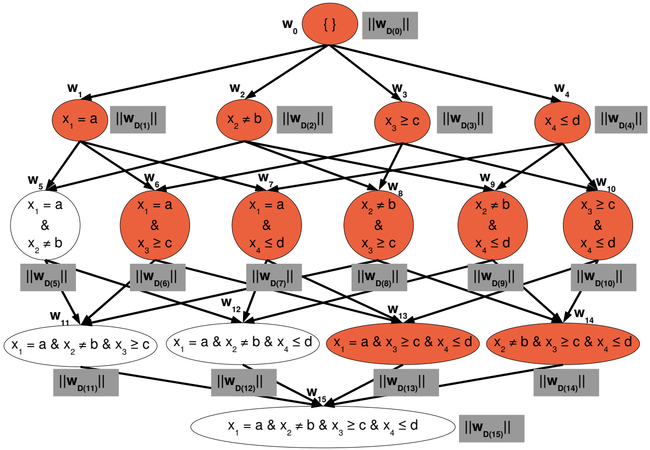
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

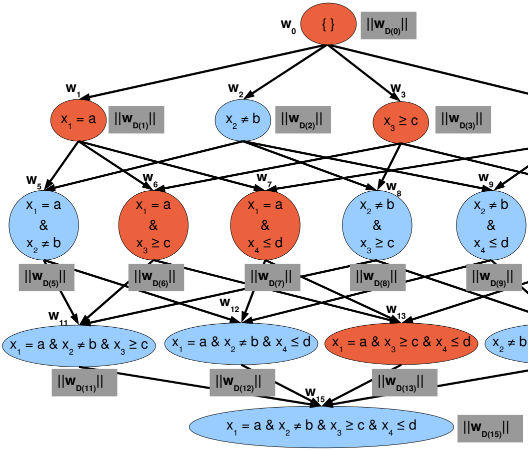
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

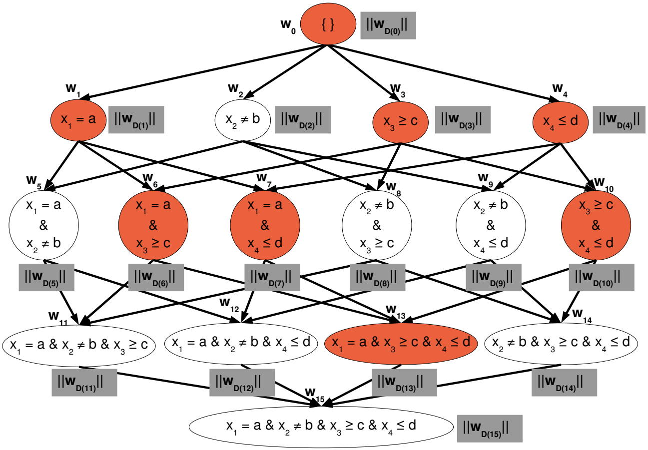
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

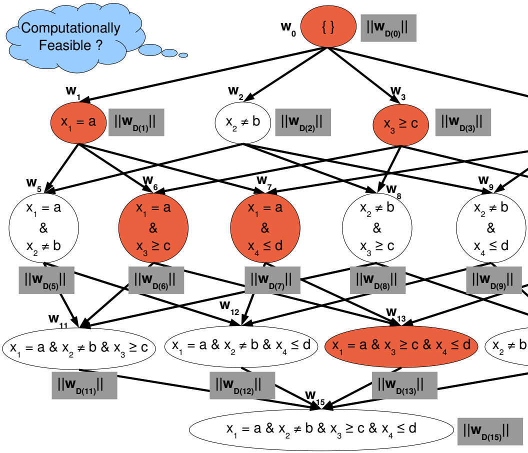
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} ||\mathbf{w}_{D(v)}||_2\right)^2$



An Improved Formulation

Key Idea:

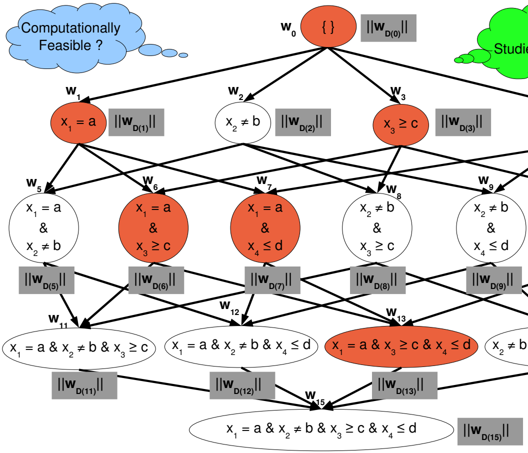
Block l_1 regularizer discourages long rules: $\left(\sum_{v \in \mathcal{V}} \|\mathbf{w}_{D(v)}\|_2\right)^2$



An Improved Formulation

Key Idea:

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

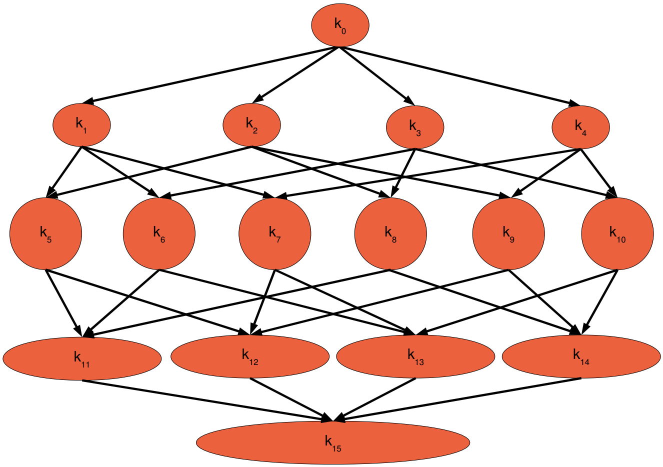


Hierarchical Kernel Learning (HKL)(Bach, 08)

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

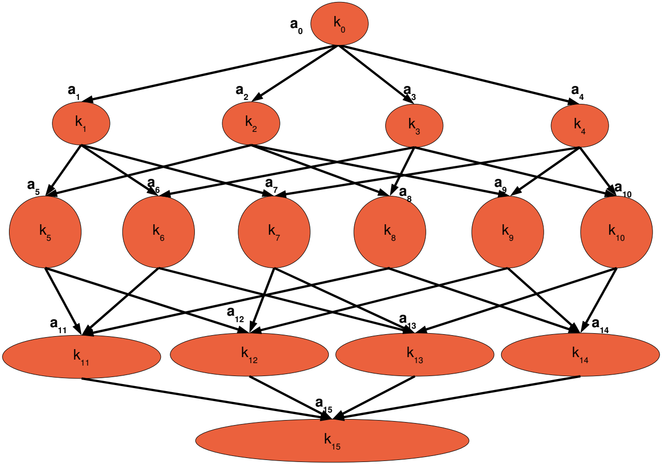
Hierarchical Kernel Learning (HKL)(Bach, 08)

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



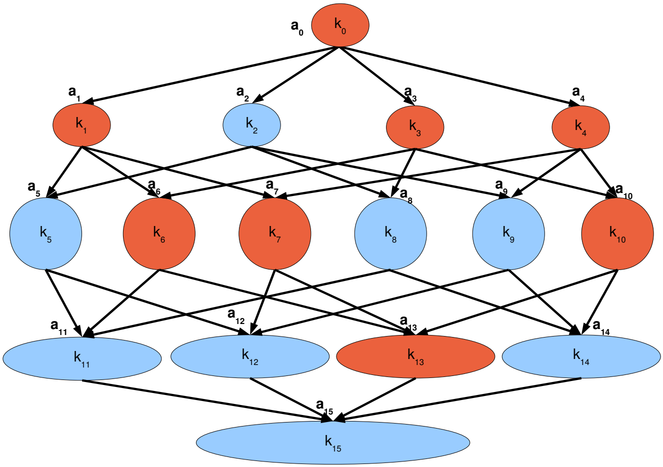
Hierarchical Kernel Learning (HKL) (Bach, 08)

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



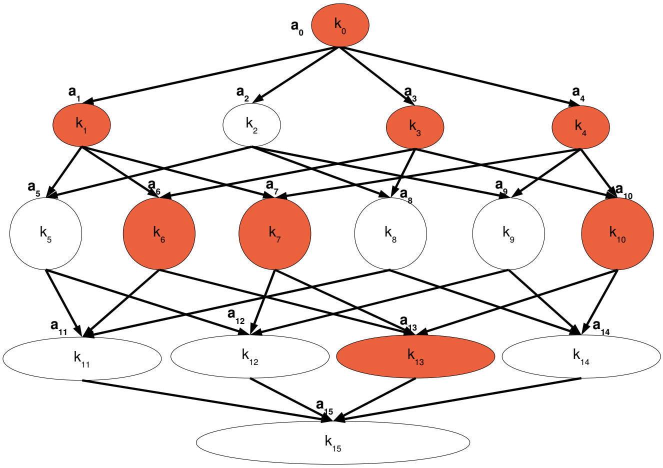
Hierarchical Kernel Learning (HKL)(Bach, 08)

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



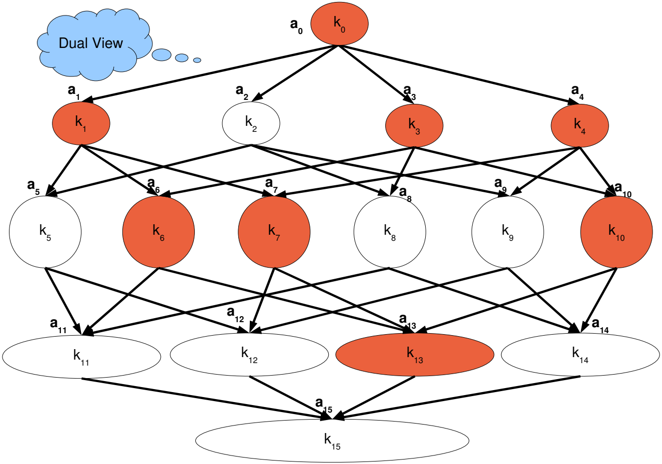
Hierarchical Kernel Learning (HKL)(Bach, 08)

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



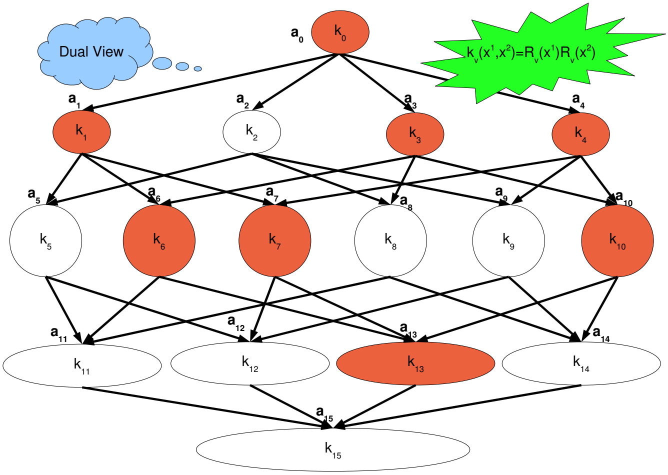
Hierarchical Kernel Learning (HKL) (Bach, 08)

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



Hierarchical Kernel Learning (HKL) (Bach, 08)

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



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 $m = 96, p = 27$	0.652 ± 0.068	0.747 ± 0.026	0.633 ± 0.011	$\mathbf{0.889} \pm 0.029$
BALANCE $m = 28, p = 51$	0.835 ± 0.034	0.856 ± 0.027	0.827 ± 0.013	$\mathbf{0.893} \pm 0.027$
HABERMAN $m = 31, p = 28$	0.512 ± 0.072	0.565 ± 0.066	0.424 ± 0.000	$\mathbf{0.594} \pm 0.056$
CAR $m = 159, p = 21$	0.913 ± 0.033	0.895 ± 0.024	0.755 ± 0.028	$\mathbf{0.943} \pm 0.024$
BLOOD TRANS. $m = 75, p = 32$	0.549 ± 0.092	0.559 ± 0.100	0.489 ± 0.054	$\mathbf{0.594} \pm 0.009$
CMC $m = 114, p = 38$	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 $m = 96, p = 27$	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 $m = 28, p = 51$	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 $m = 31, p = 28$	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 $m = 159, p = 21$	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. $m = 75, p = 32$	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 $m = 114, p = 38$	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**)

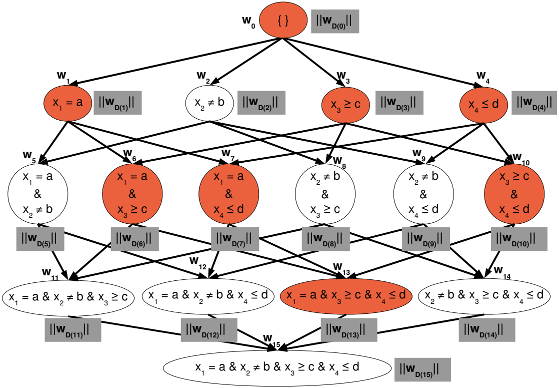
Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL
TIC-TAC-TOE $m = 96, p = 27$	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 $m = 28, p = 51$	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 $m = 31, p = 28$	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 $m = 159, p = 21$	0.913 ± 0.033 (**34**, 3.12)	0.895 ± 0.024 (141, 2.27)	0.755 ± 0.028 (80, 1.85)	**0.943** ± 0.024 (87, **1.78**)
BLOOD TRANS. $m = 75, p = 32$	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 $m = 114, p = 38$	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**)

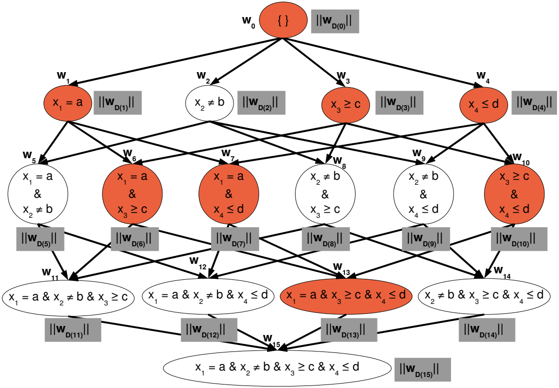
Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL
TIC-TAC-TOE $m = 96, p = 27$	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 $m = 28, p = 51$	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 $m = 31, p = 28$	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 $m = 159, p = 21$	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. $m = 75, p = 32$	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 $m = 114, p = 38$	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

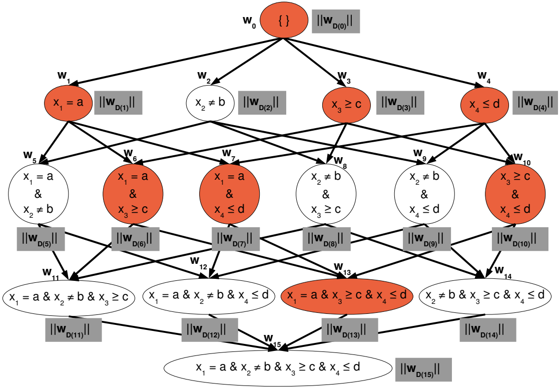


HKL — Introspection



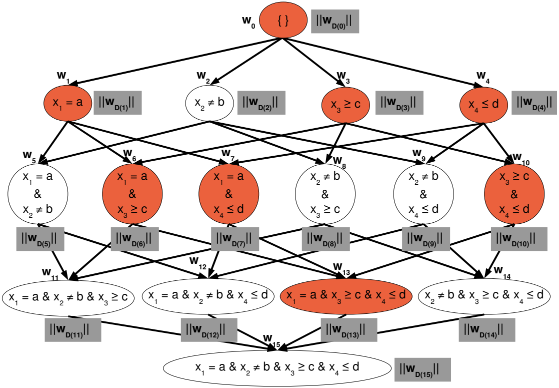
- Node selected only if all its ancestors are!

HKL — Introspection



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

HKL — Introspection



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

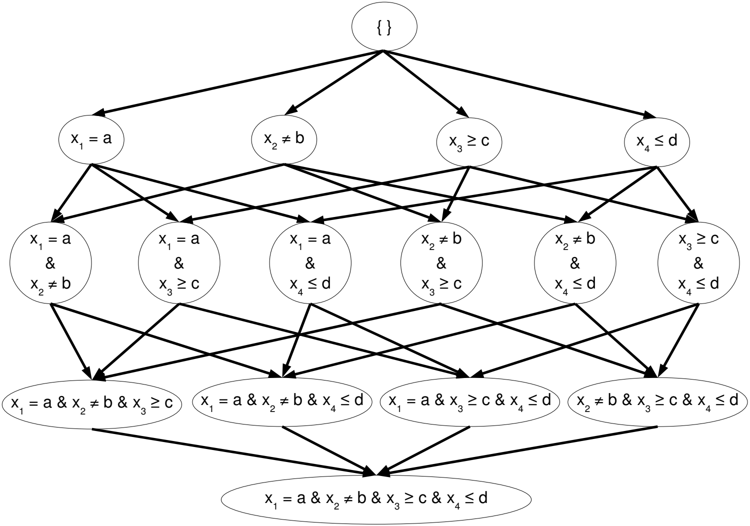
Proposed Formulation

Generalized HKL

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

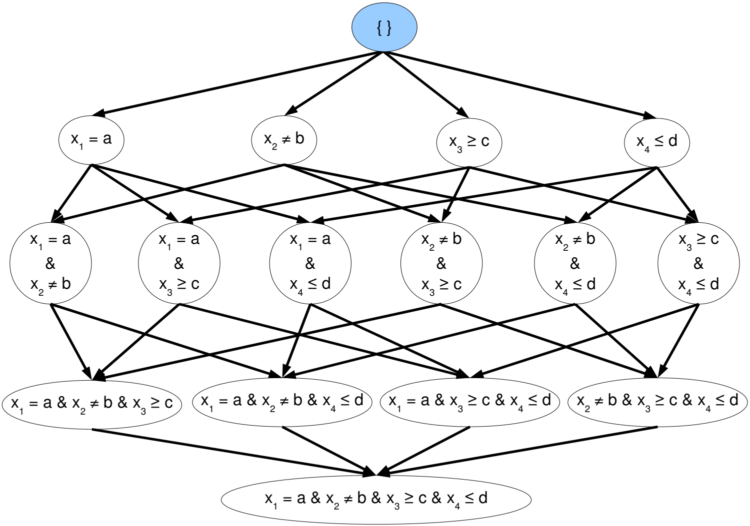
where $1 < \rho \leq 2$.

Active Set Method



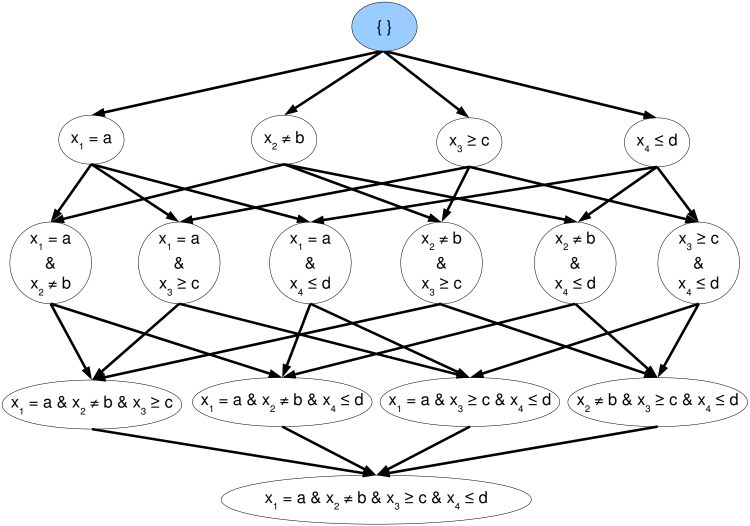
Active Set Method

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



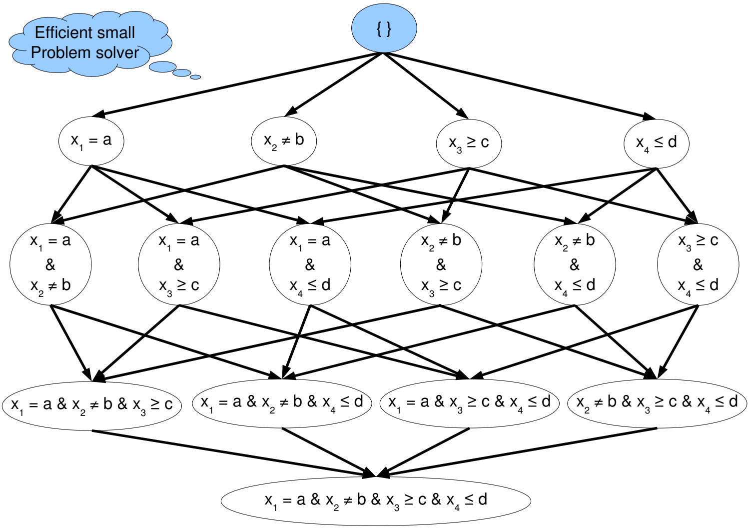
Active Set Method

Solve small problem



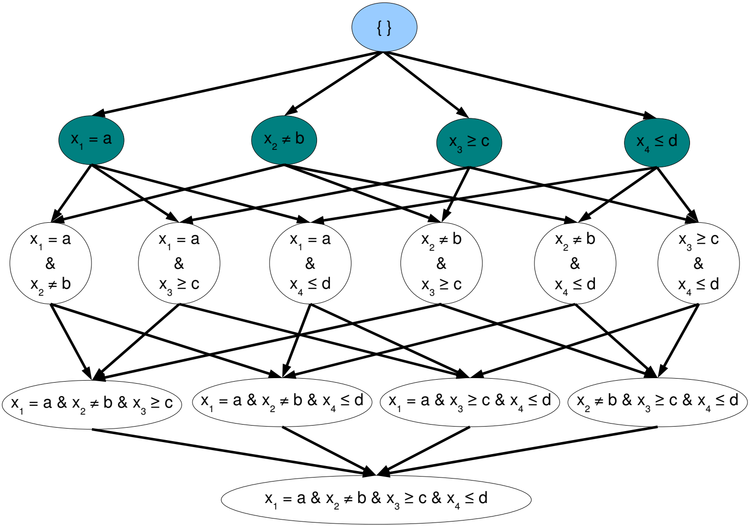
Active Set Method

Solve small problem



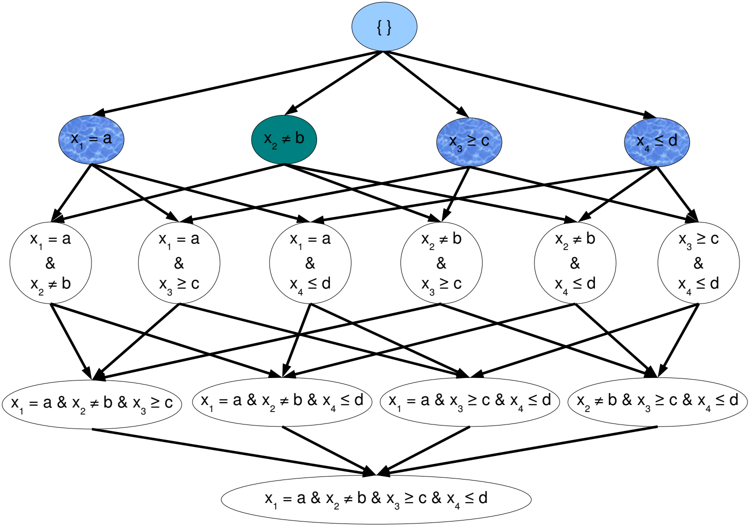
Active Set Method

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



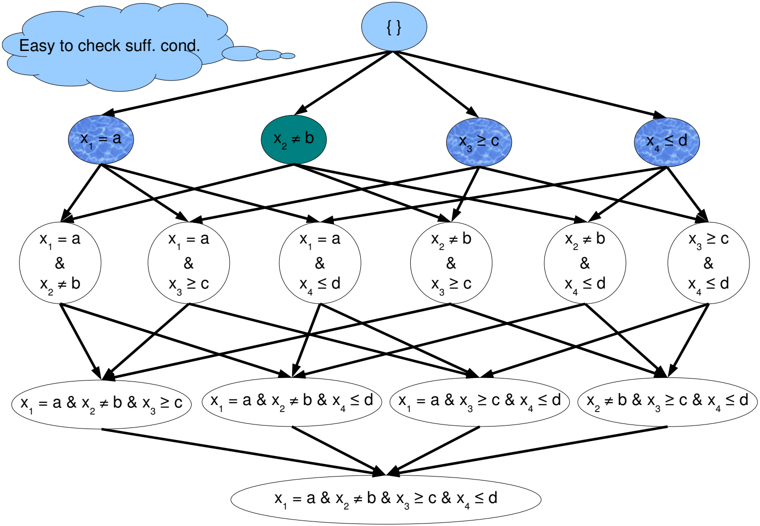
Active Set Method

Among them, optimality condition violators



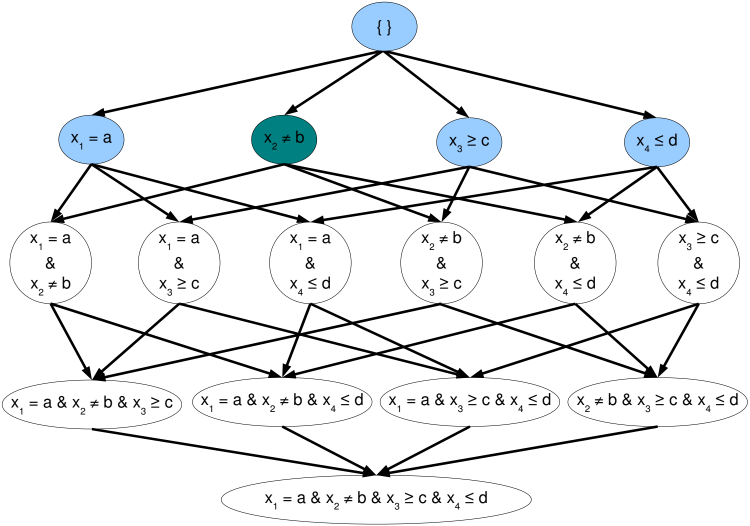
Active Set Method

Among them, optimality condition violators



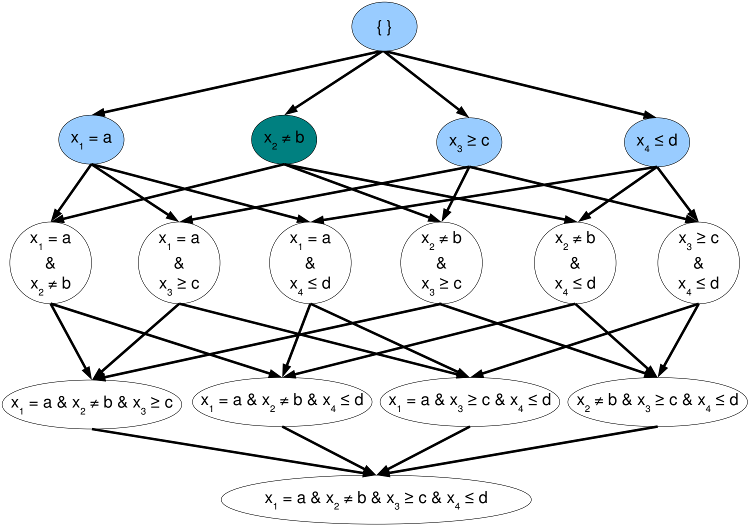
Active Set Method

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



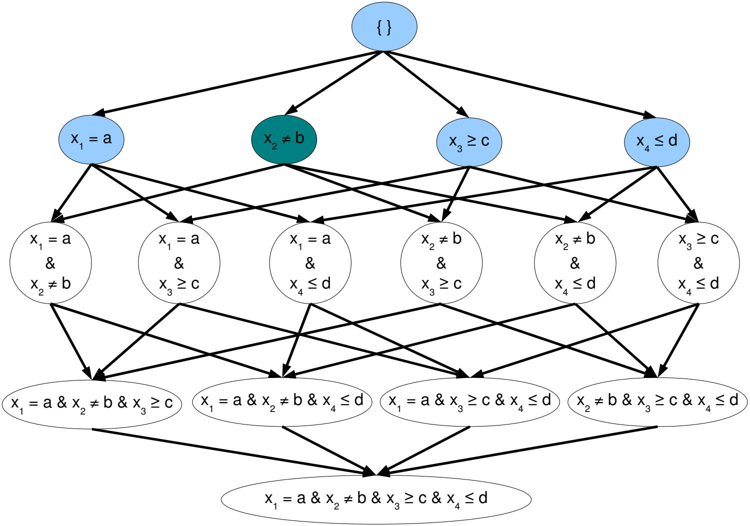
Active Set Method

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



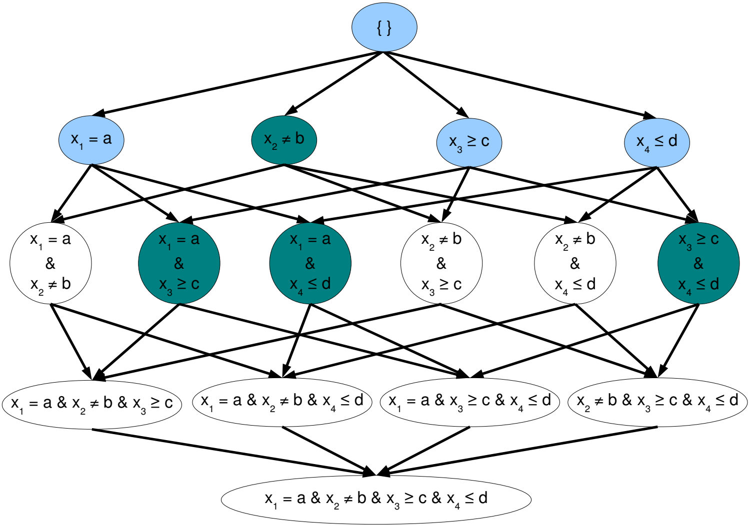
Active Set Method

Solve small problem



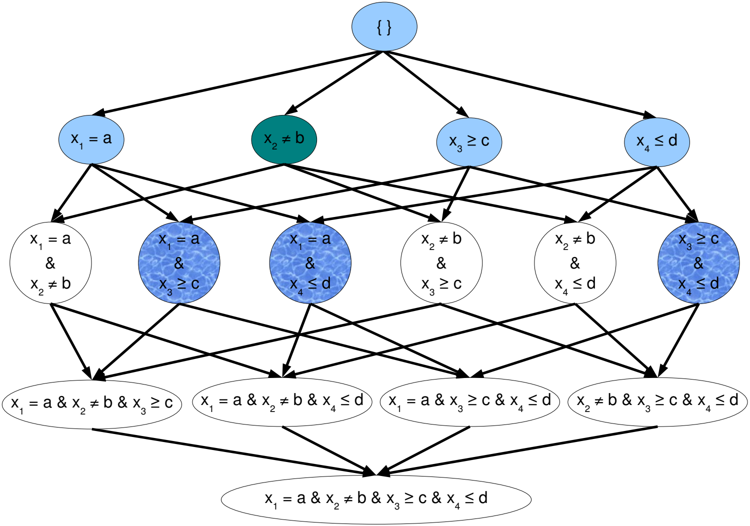
Active Set Method

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



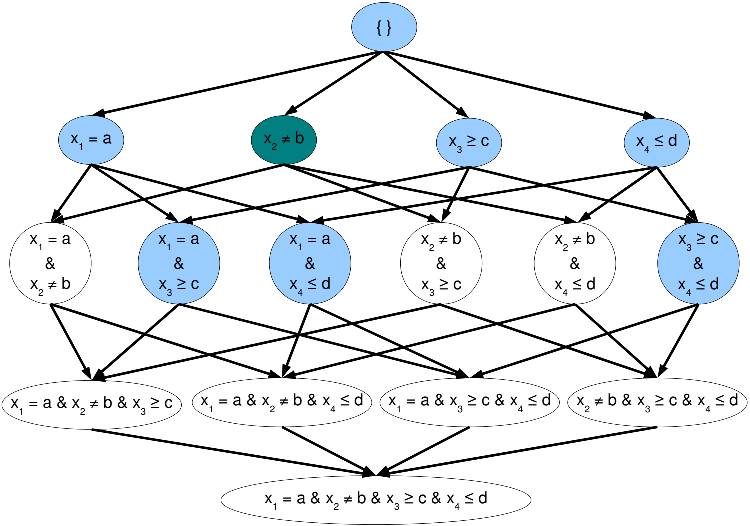
Active Set Method

Among them, optimality condition violators



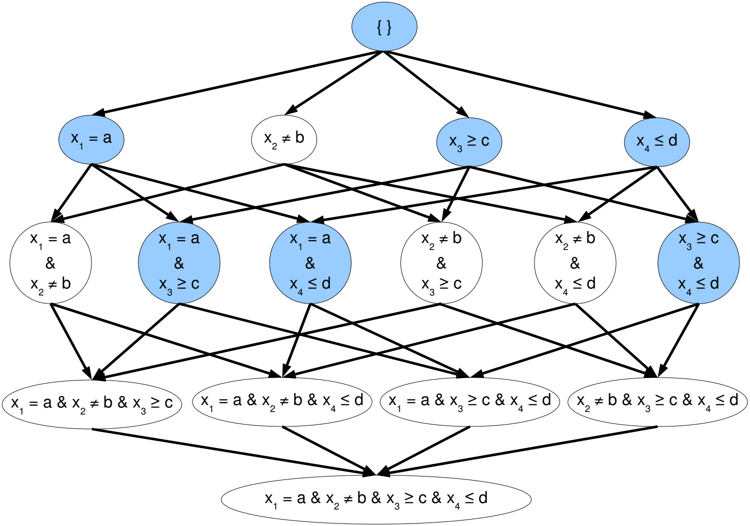
Active Set Method

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



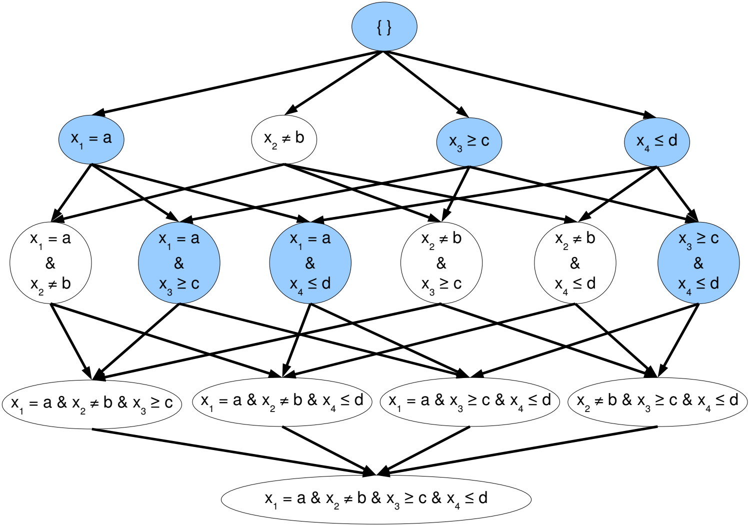
Active Set Method

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



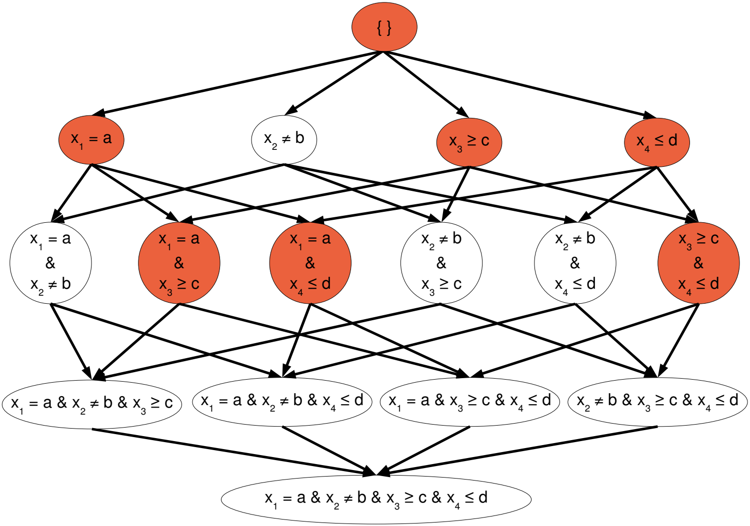
Active Set Method

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



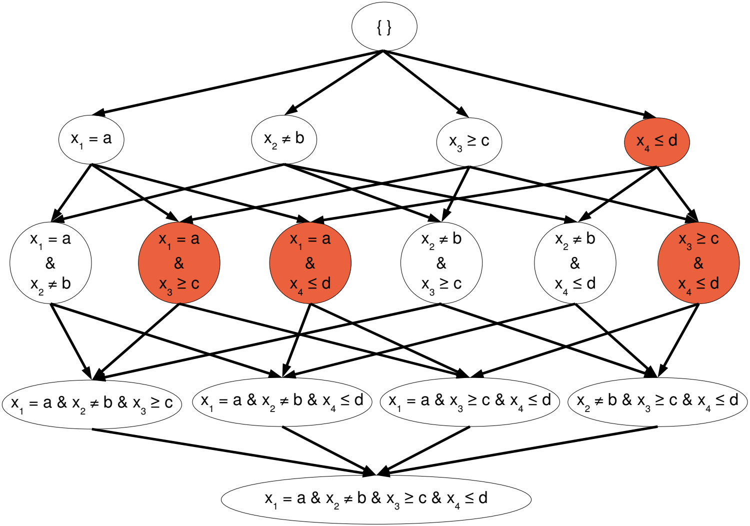
Active Set Method

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:

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

- $\rho \to 1$ ($\bar{\rho} \to \infty$), 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:

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

- $\rho \rightarrow 1$ ($\bar{\rho} \rightarrow \infty$), 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	HKL[2] $_{\rho=1.1}$
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**)

[2] Code at http://www.cse.iitb.ac.in/~pratik.j/REL-HKL.tar.gz

Performance Comparison

Dataset	RuleFit	SLI	ENDER	HKL	$\text{HKL}^2{}_{\rho=1.1}$
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**)

[2] Code at `http://www.cse.iitb.ac.in/~pratik.j/REL-HKL.tar.gz`

Summary

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

[3] 73% decrease in terms of classification error!

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[3]
  - Applicable elsewhere

[3] 73% decrease in terms of classification error!

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[3]
  - 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.

[3] 73% decrease in terms of classification error!

Questions?


Thank You