Recursive Regularization for Large-scale Classification with Hierarchical and Graphical Dependencies

Siddharth Gopal    Yiming Yang

Carnegie Mellon University

12th Aug 2013
Outline of the Talk

- Motivation
- Related work
- Proposed model and Optimization
- Experiments
Motivation

- Big data era - easy access to lots of structured data.
- Hierarchies and graphs provide a natural way to organize data.
- For example
  1. **Open Directory Project** - A collection of Billions of webpages into a hierarchy with $\sim 300,000$ classes.
  2. **International Patent Taxonomy** - Millions of patents across the world follow this hierarchy.
  3. **Wikipedia pages** - Millions of wikipedia pages have associated categories which are linked to each other.
Assign an unseen webpage/patent/article to one or more nodes in the hierarchy or graph.
Assign an unseen webpage/patent/article to one or more nodes in the hierarchy or graph.

How to use the inter-class dependencies to improve classification?

A webpage that belongs to the class ‘medicine’ is unlikely to also belong to ‘mutual funds’.
Assign an unseen webpage/patent/article to one or more nodes in the hierarchy or graph.

How to use the inter-class dependencies to improve classification?

A webpage that belongs to the class ‘medicine’ is unlikely to also belong to ‘mutual funds’.

How to scale to large number of classes?
### Scalability

Some existing datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Instances</th>
<th>#Labels</th>
<th>#Features</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODP subset</td>
<td>394,756</td>
<td>27,875</td>
<td>594,158</td>
<td>16,562,154,250</td>
</tr>
<tr>
<td>Wikipedia subset</td>
<td>2,365,436</td>
<td>325,056</td>
<td>1,617,899</td>
<td>525,907,777,344</td>
</tr>
</tbody>
</table>

Some existing datasets

- ODP subset: ∼66 GB of parameters
- Wikipedia subsets: ∼2 TB of parameters
Scalability

Some existing datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Instances</th>
<th>#Labels</th>
<th>#Features</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODP subset</td>
<td>394,756</td>
<td>27,875</td>
<td>594,158</td>
<td>16,562,154,250</td>
</tr>
<tr>
<td>Wikipedia subset</td>
<td>2,365,436</td>
<td>325,056</td>
<td>1,617,899</td>
<td>525,907,777,344</td>
</tr>
</tbody>
</table>

- ODP subset $\sim$ 66 GB of parameters
- Wikipedia subsets $\sim$ 2 TB of parameters
Scalability

Some existing datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Instances</th>
<th>#Labels</th>
<th>#Features</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODP subset</td>
<td>394,756</td>
<td>27,875</td>
<td>594,158</td>
<td>16,562,154,250</td>
</tr>
<tr>
<td>Wikipedia subset</td>
<td>2,365,436</td>
<td>325,056</td>
<td>1,617,899</td>
<td>525,907,777,344</td>
</tr>
</tbody>
</table>

- ODP subset $\sim 66$ GB of parameters
- Wikipedia subsets $\sim 2$ TB of parameters

Focus

1. How to use interclass dependencies?
2. How to scale?
Related Work

- **Earlier works** Top-down *pachinko machine* style approaches
  - [Dumais and Chen, 2000], [Yang et al., 2003] [Liu et al., 2005],
  - [Koller and Sahami, 1997]

- **Large-margin methods**
  1. Maximize the margin between correct and incorrect labels based on a hierarchical loss.
  2. Discriminant functions takes contribution from all nodes along the path to root-node.
  - [Tsochantaridis et al., 2006], [Cai and Hofmann, 2004], [Rousu et al., 2006],
  - [Dekel et al., 2004], [Cesa-Bianchi et al., 2006]

- **Bayesian methods** Hierarchical Naive Bayes
  - [McCallum et al., 1998] , Correlated Multinomial Logit
  - [Shahbaba and Neal, 2007], Hierarchical Bayesian logistic regression [Gopal et al., 2012]
Notations

Given training examples and hierarchy

1. Hierarchy of nodes $\mathcal{N}$ defined by parent function $\pi(n)$.

2. $N$ training examples,
   - $x_i$ denote $i^{th}$ instance
   - $y_{in}$ denotes whether $x_i$ is labeled to node $n$.

3. $\mathcal{T}$ denotes set of leaf nodes.

4. $C_n$ denotes the set of child-nodes of node $n$. 
Proposed model

Learn a prediction function with parameters $\mathbf{W}$. Estimate $\mathbf{W}$ as

$$\arg\min_{\mathbf{W}} \lambda(\mathbf{W}) + C \times R_{emp}$$

Each node $n$ is associated with parameter vector $w_n$. 
Proposed model

Define $R_{emp}$ as the empirical loss using loss function $L$ at the leaf-nodes.

$$R_{emp} = \sum_{i=1}^{N} \sum_{n \in T} L(w_n^T x_i, y_{in})$$
Proposed model

Define $R_{emp}$ as the empirical loss using loss function $L$ at the leaf-nodes.

$$R_{emp} = \sum_{i=1}^{N} \sum_{n \in T} L(w_n^T x_i, y_{in})$$

Incorporate the hierarchy into regularization term $\lambda(W)$

$$\lambda(W) = \sum_{n \in N} \| w_n - w_{\pi(n)} \|^2$$
Proposed model

Define $R_{emp}$ as the empirical loss using loss function $L$ at the leaf-nodes.

$$R_{emp} = \sum_{i=1}^{N} \sum_{n \in T} L(w_n^T x_i, y_{in})$$

Incorporate the hierarchy into regularization term $\lambda(W)$

$$\lambda(W) = \sum_{n \in N} \|w_n - w_{\pi(n)}\|^2$$

With a graph with edges $E \subset \{(i, j) : i, j \in N\}$,

$$\lambda(W) = \sum_{(i, j) \in E} \|w_i - w_j\|^2$$
Advantages

Advantages over other works

2. Multiple independent problems that can be parallelized.
3. Flexibility in choosing a loss function.
Advantages

Advantages over other works

2. Multiple independent problems that can be parallelized.
3. Flexibility in choosing a loss function.

\[
\text{[HR-SVM]} \quad \min_{W} \sum_{n \in \mathcal{N}} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{n \in \mathcal{T}} \sum_{i=1}^{N} (1 - y_{in} w_n^T x_i)_+ \\
\text{[HR-LR]} \quad \min_{W} \sum_{n \in \mathcal{N}} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{n \in \mathcal{T}} \sum_{i=1}^{N} \log(1 + \exp(-y_{in} w_n^T x_i))
\]
Optimizing with Hinge-loss

**[HR-SVM]**  \[
\min_w \sum_{n \in N} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{n \in T} \sum_{i=1}^{N} (1 - y_{in} w_n^T x_i) +
\]

Problems
- Large-number of parameters (2 Terabytes)
- Non-differentiability of Hinge-loss
Optimizing with Hinge-loss

\[
[\text{HR-SVM}] \quad \min_W \sum_{n \in \mathcal{N}} \frac{1}{2} ||w_n - w_{\pi(n)}||^2 + C \sum_{n \in \mathcal{T}} \sum_{i=1}^{N} (1 - y_n w_n^T x_i) +
\]

Problems

- Large-number of parameters (2 Terabytes)
- Non-differentiability of Hinge-loss

Solution

- Block-coordinate descent to handle large number of parameters (update one \(w_n\) at a time).
- Solve dual problem within block for non-differentiability.
Optimizing HR-SVM

Update for non-leaf node $w_n$, 

\[ w_n = \frac{1}{|C_n| + 1} \left[ \pi(n) + \sum_{c \in C_n} w_c \right] \]

For leaf-node, the objective is 

\[ \min_{w_n} \frac{1}{2} \| w_n - \pi(n) \|_2^2 + C_N \sum_{i=1}^N (1 - y_{in} w_n^\top x_i) + \sum_{i=1}^N \alpha_i \left( 1 - y_{in} \pi(n)^\top x_i \right) \]

\[ \text{s.t. } 0 \leq \alpha_i \leq C \]

[Use co-ordinate descent again! Update one $\alpha_i$ at a time]
Optimizing HR-SVM

Update for non-leaf node $w_n$,

$$w_n = \frac{1}{|C_n| + 1} \left( w_{\pi(n)} + \sum_{c \in C_n} w_c \right)$$
Optimizing HR-SVM

Update for non-leaf node $w_n$,

$$w_n = \frac{1}{|C_n| + 1} \left( w_{\pi(n)} + \sum_{c \in C_n} w_c \right)$$

For leaf-node, the objective is

$$\min_{w_n} \frac{1}{2} \left\| w_n - w_{\pi(n)} \right\|^2 + C \sum_{i=1}^{N} (1 - y_{in} w_n^T x_i)_+$$
Optimizing HR-SVM

Update for non-leaf node $w_n$,

$$w_n = \frac{1}{|C_n| + 1} \left( w_{\pi(n)} + \sum_{c \in C_n} w_c \right)$$

For leaf-node, the objective is

$$\min_{w_n} \frac{1}{2} \left| w_n - w_{\pi(n)} \right|^2 + C \sum_{i=1}^{N} (1 - y_i w_n^T x_i)_+$$

**Dual**

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{N} \alpha_i (1 - y_i w_{\pi(n)}^T x_i)$$

s.t. $0 \leq \alpha \leq C$
Optimizing HR-SVM

Update for non-leaf node $w_n$,

$$
    w_n = \frac{1}{|C_n| + 1} \left( w_{\pi(n)} + \sum_{c \in C_n} w_c \right)
$$

For leaf-node, the objective is

$$
    \min_{w_n} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{i=1}^{N} (1 - y_{in} w_n^T x_i)_+
$$

**Dual**

$$
    \min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_{in} y_{jn} x_i^T x_j - \sum_{i=1}^{N} \alpha_i (1 - y_{in} w_{\pi(n)}^T x_i)
$$

s.t. $0 \leq \alpha \leq C$

[Use co-ordinate descent again! Update one $\alpha_i$ at a time.]
Optimizing HR-SVM

It turns out each $\alpha_i$ has closed form update.

\[
G = \left( \sum_{j=1}^{N} \alpha_j y_j x_j \right)^T x_i - 1 + y_i w_{\pi(n)}^T x_i
\]

\[
\alpha_i^{new} = \min \left( \max \left( \alpha_i^{old} - \frac{G}{x_i^T x_i}, 0 \right), C \right)
\]
Optimizing HR-SVM

It turns out the each $\alpha_i$ has closed form update.

$$G = (\sum_{j=1}^{N} \alpha_j y_j n_j x_j)^\top x_i - 1 + y_i n_i w_{\pi(n)}^\top x_i$$

$$\alpha_i^{new} = \min \left( \max \left( \alpha_i^{old} - \frac{G}{x_i^\top x_i}, 0 \right), C \right)$$

For each $\alpha_i$ update, naive time complexity : $O(Training data)$.
Optimizing HR-SVM

It turns out the each $\alpha_i$ has closed form update.

$$G = \left( \sum_{j=1}^{N} \alpha_j y_j x_j \right)^T x_i - 1 + y_i w_{\pi(n)}^T x_i$$

$$\alpha_i^{new} = \min \left( \max \left( \alpha_i^{old} - \frac{G}{x_i^T x_i}, 0 \right), C \right)$$

For each $\alpha_i$ update, naive time complexity: $O(Trainingdata)$.

Trick: precompute $\sum_{j=1}^{N} \alpha_j y_j x_j$ and keep maintaining the sum.

New time complexity: $O(nnz(x_i))$
It turns out each $\alpha_i$ has a closed form update.

$$G = \left( \sum_{j=1}^{N} \alpha_j y_{jn} x_j \right)^T x_i - 1 + y_{in} w_{\pi(n)}^T x_i$$

$$\alpha_i^{\text{new}} = \min \left( \max \left( \alpha_i^{\text{old}} - \frac{G}{x_i^T x_i}, 0 \right), C \right)$$

For each $\alpha_i$ update, the naive time complexity is $O(Trainingdata)$.

Trick: precompute $\sum_{j=1}^{N} \alpha_j y_{jn} x_j$ and keep maintaining the sum.

New time complexity: $O(nnz(x_i))$

Recover original primal solution, $w_n = w_{\pi(n)} + \sum_{i=1}^{N} \alpha_i y_{in} x_i$. 
Optimizing HR-LR

\[
[\text{HR-LR}] \quad \min_w \sum_{n \in \mathcal{N}} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{n \in \mathcal{T}} \sum_{i=1}^{N} \log(1 + \exp(-y_{in} w_n^T x_i))
\]

1. Convex and Differentiable.
2. Block co-ordinate descent to handle parameter size.
3. LBFGS for optimization.
Assumption: Nodes closer in the hierarchy/graph share similar model parameters.

Model: Incorporate the structure into $\lambda(W)$.

$[HR-LR]$ \[
\min W \sum_{n \in N} \frac{1}{2} ||w_n - w_{\pi(n)}||_2 + C \sum_{n \in T} N \sum_{i=1}^N \log(1 + \exp(-y_{in} W_n^\top x_i))
\]

$[HR-SVM]$ \[
\min W \sum_{n \in N} \frac{1}{2} ||w_n - w_{\pi(n)}||_2 + C \sum_{n \in T} N \sum_{i=1}^N (1 - y_{in} W_n^\top x_i) + 
\]

Block co-ordinate descent to avoid memory issues.

Handle non differentiability using dual space.

RECAP
Recap

RECAP

Assumption: Nodes closer in the hierarchy/graph share similar model parameters.
Recap

RECAP

1. **Assumption**: Nodes closer in the hierarchy/graph share similar model parameters.

2. **Model**: Incorporate the structure into $\lambda(W)$.

\[
\begin{align*}
\text{[HR-LR]} & \quad \min_W \sum_{n \in N} \frac{1}{2} \|w_n - w_{\pi(n)}\|^2 + C \sum_{n \in T} \sum_{i=1}^{N} \log(1 + \exp(-y_{in}w_n^T x_i)) \\
\text{[HR-SVM]} & \quad \min_W \sum_{n \in N} \frac{1}{2} \|w_n - w_{\pi(n)}\|^2 + C \sum_{n \in T} \sum_{i=1}^{N} (1 - y_{in}w_n^T x_i) +
\end{align*}
\]
RECAP

1. **Assumption**: Nodes closer in the hierarchy/graph share similar model parameters.

2. **Model**: Incorporate the structure into $\lambda(W)$.

   \[
   \text{[HR-LR]} \quad \min_W \sum_{n \in N} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{n \in T} \sum_{i=1}^{N} \log(1 + \exp(-y_in w_n^T x_i))
   \]

   \[
   \text{[HR-SVM]} \quad \min_W \sum_{n \in N} \frac{1}{2} \| w_n - w_{\pi(n)} \|^2 + C \sum_{n \in T} \sum_{i=1}^{N} (1 - y_in w_n^T x_i)_+
   \]

3. **Block co-ordinate descent to avoid memory issues.**
Recap

RECAP

1. **Assumption**: Nodes closer in the hierarchy/graph share similar model parameters.

2. **Model**: Incorporate the structure into $\lambda(W)$.

$$\text{[HR-LR]} \quad \min_W \sum_{n \in \mathcal{N}} \frac{1}{2} \|w_n - w_{\pi(n)}\|^2 + C \sum_{n \in \mathcal{T}} \sum_{i=1}^{N} \log(1 + \exp(-y_{in}w_n^T x_i))$$

$$\text{[HR-SVM]} \quad \min_W \sum_{n \in \mathcal{N}} \frac{1}{2} \|w_n - w_{\pi(n)}\|^2 + C \sum_{n \in \mathcal{T}} \sum_{i=1}^{N} (1 - y_{in}w_n^T x_i)_+$$

3. Block co-ordinate descent to avoid memory issues.

4. Handle non differentiability using dual space.
Parallelization

Updating only one block of parameters at a time is suboptimal.
Parallelization

Updating only one block of parameters at a time is suboptimal.

Can we update multiple blocks in parallel?
Parallelization

Updating only one block of parameters at a time is suboptimal.

Can we update multiple blocks in parallel?

Key point for parallelization: Parameters are only locally dependent.

1. In a hierarchy, the parameters of a node depend only on parent and children.
2. In a graph, the parameters of a node depend on its neighbours.
Hierarchies:
1. Fix parameters at odd-levels, optimize even levels in parallel.
2. Fix parameters at even-level, optimize odd levels in parallel.
3. Repeat until convergence.
Hierarchies:
1. Fix parameters at odd-levels, optimize even levels in parallel.
2. Fix parameters at even-level, optimize odd levels in parallel.
3. Repeat until convergence.

Graphs: First find the minimum graph coloring [Np-hard]
1. Pick a color.
2. In parallel, optimize all nodes with that color.
3. Repeat with a different color.
### DATASETS

<table>
<thead>
<tr>
<th>Name</th>
<th>#Training</th>
<th>#Classes</th>
<th>#dims</th>
<th>Avg #labels per instance</th>
<th>Parameter size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEF</td>
<td>10,000</td>
<td>87</td>
<td>89</td>
<td>1</td>
<td>30 KB</td>
</tr>
<tr>
<td>RCV1</td>
<td>23,149</td>
<td>137</td>
<td>48,734</td>
<td>3.18</td>
<td>26 MB</td>
</tr>
<tr>
<td>IPC</td>
<td>46,324</td>
<td>552</td>
<td>541,869</td>
<td>1</td>
<td>1.1 GB</td>
</tr>
<tr>
<td>LSHTC-small</td>
<td>4,463</td>
<td>1,563</td>
<td>51,033</td>
<td>1</td>
<td>320 MB</td>
</tr>
<tr>
<td>DMOZ-2010</td>
<td>128,710</td>
<td>15,358</td>
<td>381,580</td>
<td>1</td>
<td>23 GB</td>
</tr>
<tr>
<td>DMOZ-2012</td>
<td>383,408</td>
<td>13,347</td>
<td>348,548</td>
<td>1</td>
<td>18 GB</td>
</tr>
<tr>
<td>DMOZ-2011</td>
<td>394,756</td>
<td>27,875</td>
<td>594,158</td>
<td>1.03</td>
<td>66 GB</td>
</tr>
<tr>
<td>SWIKI-2011</td>
<td>456,886</td>
<td>50,312</td>
<td>346,299</td>
<td>1.85</td>
<td>70 GB</td>
</tr>
<tr>
<td>LWIKI</td>
<td>2,365,436</td>
<td>614,428</td>
<td>1,617,899</td>
<td>3.26</td>
<td>2 TB</td>
</tr>
</tbody>
</table>
## Comparison with published results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LSHTC Published Results</th>
<th>HR-SVM</th>
<th>HR-LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMOZ-2010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-$F_1$</td>
<td>34.12</td>
<td>33.12</td>
<td>32.42</td>
</tr>
<tr>
<td>Micro-$F_1$</td>
<td>46.76</td>
<td>46.02</td>
<td>45.84</td>
</tr>
<tr>
<td>DMOZ-2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-$F_1$</td>
<td>31.36</td>
<td>33.05</td>
<td>20.04</td>
</tr>
<tr>
<td>Micro-$F_1$</td>
<td>51.98</td>
<td>57.17</td>
<td>53.18</td>
</tr>
<tr>
<td>DMOZ-2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-$F_1$</td>
<td>26.48</td>
<td>25.69</td>
<td>23.90</td>
</tr>
<tr>
<td>Micro-$F_1$</td>
<td>38.85</td>
<td>43.73</td>
<td>42.27</td>
</tr>
<tr>
<td>SWIKI-2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-$F_1$</td>
<td>23.16</td>
<td>28.72</td>
<td>24.26</td>
</tr>
<tr>
<td>Micro-$F_1$</td>
<td>37.39</td>
<td>41.79</td>
<td>40.99</td>
</tr>
<tr>
<td>LWIKI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-$F_1$</td>
<td>18.68</td>
<td>22.31</td>
<td>20.22</td>
</tr>
<tr>
<td>Micro-$F_1$</td>
<td>34.67</td>
<td>38.08</td>
<td>37.67</td>
</tr>
</tbody>
</table>
Methods for comparison

- **Flat baselines:**
  1. One-versus-rest binary Support Vector Machines (SVM)
  2. One-versus-rest regularized logistic regression (LR).

- **Hierarchical baselines:**
  1. **Hierarchical SVM** (HSVM) [Tschantaridis et al., 2006] a large-margin discriminative method with path dependent discriminant function.
  2. **Hierarchical Orthogonal Transfer** (OT) [Zhou et al., 2011], a large-margin method enforcing orthogonality between the parent and the children.
  3. **Top-down SVM** (TD) a Pachinko-machine style SVM.
  4. **Hierarchical Bayesian Logistic Regression** (HBLR), [Gopal et al., 2012], our previous work using a fully Bayesian hierarchical model.
    1. Computationally more costly than HR-LR
    2. Not applicable for graph-based dependencies
Against flat baselines

HR-SVM vs SVM (Improvement)

CLEF
RCV1
IPC
LSHTC-small
DMOZ-2010
DMOZ-2012
DMOZ-2011
SWIKI-2011
LWIKI

0
1
2
3
4
5
6

Micro-F1
Macro-F1

Siddharth Gopal, Yiming Yang
Against flat baselines

HR-LR vs LR (Improvement)

CLEF     RCV1     IPC     LSHTC-small     DMOZ-2010     DMOZ-2012     DMOZ-2011     SWIKI-2011     LWIKI

Macro-F1

Siddharth Gopal, Yiming Yang
Recursive Regularization for Large-scale Classification with Hierarchical and Graphical Dependencies
Time complexity

**HR-SVM vs SVM (Computational cost)**

- CLEF
- RCV1
- IPC
- LSHTC-small
- DMOZ-2010
- DMOZ-2012
- SWIKI-2011
- LWIKI

**HR-LR vs LR (Computational cost)**

- CLEF
- RCV1
- IPC
- LSHTC-small
- DMOZ-2010
- DMOZ-2012
- SWIKI-2011
- LWIKI

Siddharth Gopal, Yiming Yang

Recursive Regularization for Large-scale Classification with Hierarchical and Graphical Dependencies
A Model that can

1. Use both hierarchial and graphical dependencies between classes to improve classification.
2. And can be scaled to real-world data.

Thanks!
### Micro-F1 comparison

<table>
<thead>
<tr>
<th>Datasets</th>
<th>HR-SVM</th>
<th>TD</th>
<th>HSVN</th>
<th>OT</th>
<th>HBLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEF</td>
<td>80.02</td>
<td>70.11</td>
<td>79.72</td>
<td>73.84</td>
<td>81.41</td>
</tr>
<tr>
<td>RCV1</td>
<td>81.66</td>
<td>71.34</td>
<td>NA</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>IPC</td>
<td>54.26</td>
<td>50.34</td>
<td>NS</td>
<td>NS</td>
<td>56.02</td>
</tr>
<tr>
<td>LSHTC-small</td>
<td>45.31</td>
<td>38.48</td>
<td>39.66</td>
<td>37.12</td>
<td>46.03</td>
</tr>
<tr>
<td>DMOZ-2010</td>
<td>46.02</td>
<td>38.64</td>
<td>NS</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>DMOZ-2012</td>
<td>57.17</td>
<td>55.14</td>
<td>NS</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>DMOZ-2011</td>
<td>43.73</td>
<td>35.91</td>
<td>NA</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>SWIKI-2011</td>
<td>41.79</td>
<td>36.65</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>LWIKI</td>
<td>38.08</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

[NA - Not applicable, NS - Not scalable]
## Time complexity

### Time (in mins)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>HR-SVM</th>
<th>TD</th>
<th>HSVM</th>
<th>OT</th>
<th>HBLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEF</td>
<td>.42</td>
<td>.13</td>
<td>3.19</td>
<td>1.31</td>
<td>3.05</td>
</tr>
<tr>
<td>RCV1</td>
<td>.55</td>
<td>.213</td>
<td>NA</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>IPC</td>
<td>6.81</td>
<td>2.21</td>
<td>NS</td>
<td>NS</td>
<td>31.2</td>
</tr>
<tr>
<td>LSHTC-small</td>
<td>.52</td>
<td>.11</td>
<td>289.60</td>
<td>132.34</td>
<td>5.22</td>
</tr>
<tr>
<td>DMOZ-2010</td>
<td>8.23</td>
<td>3.97</td>
<td>NS</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>DMOZ-2012</td>
<td>36.66</td>
<td>12.49</td>
<td>NS</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>DMOZ-2011</td>
<td>58.31</td>
<td>16.39</td>
<td>NA</td>
<td>NS</td>
<td>NA</td>
</tr>
<tr>
<td>SWIKI-2011</td>
<td>89.23</td>
<td>21.34</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>LWIKI</td>
<td>2230.54</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Conclusion

1. A **scalable** framework that can leverage class-label dependencies.

2. and that works in practice!


Kernel-based learning of hierarchical multilabel classification models.  

Improving classification when a class hierarchy is available using a hierarchy-based prior.  
*Bayesian Analysis, 2(1):221–238.*

Large margin methods for structured and interdependent output variables.  
*JMLR, 6(2):1453.*

A scalability analysis of classifiers in text categorization.  
In *SIGIR*, pages 96–103. ACM.

Hierarchical classification via orthogonal transfer.