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

Siddharth Gopal Yiming Yang

Carnegie Mellon Univeristy

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.

Challenges

Assign an unseen webpage/patent/article to one or more nodes in the hierarchy or graph.

Challenges

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*' in unlikely to also belong to '*mutual funds*'.

Challenges

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*' in unlikely to also belong to '*mutual funds*'.

How to scale to large number of classes ?

Scalability

Some existing datasets

Dataset	#Instances	#Labels	#Features	#Parameters
ODP subset	394,756	27,875	594,158	16,562,154,250
Wikipedia subset	2,365,436	325,056	1,617,899	525,907,777,344

Scalability

Some existing datasets

Dataset	#Instances	#Labels	#Features	#Parameters
ODP subset	394,756	27,875	594,158	16,562,154,250
Wikipedia subset	2,365,436	325,056	1,617,899	525,907,777,344

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

Scalability

Some existing datasets

Dataset	#Instances	#Labels	#Features	#Parameters
ODP subset	394,756	27,875	594,158	16,562,154,250
Wikipedia subset	2,365,436	325,056	1,617,899	525,907,777,344

- 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 **W**. Estimate **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.

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

Proposed model

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

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

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

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

Proposed model

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

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

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

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

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

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

Advantages

Advantages over other works

1. Structure not used in the Empirical Risk term.
2. Multiple independent problems that can be parallelized.
3. Flexibility in choosing a loss function.

Advantages

Advantages over other works

1. Structure not used in the Empirical Risk term.
2. Multiple independent problems that can be parallelized.
3. Flexibility in choosing a loss function.

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

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

Optimizing with Hinge-loss

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

Problems

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

Optimizing with Hinge-loss

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

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,

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

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

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

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

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

$$s.t. \quad 0 \leq \boldsymbol{\alpha} \leq C$$

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

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

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

$$s.t. \quad 0 \leq \boldsymbol{\alpha} \leq C$$

[Use co-ordinate descent again ! Update one α_i at a time]

Optimizing HR-SVM

It turns out the each α_i has closed form update.

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

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

Optimizing HR-SVM

It turns out the each α_i has closed form update.

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

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

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

Optimizing HR-SVM

It turns out the each α_i has closed form update.

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

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

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

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

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

Optimizing HR-SVM

It turns out the each α_i has closed form update.

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

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

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

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

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

1. Convex and Differentiable.
2. Block co-ordinate descent to handle parameter size.
3. LBFGS for optimization.

Recap

RECAP

Recap

RECAP

1. **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 $\lambda(\mathbf{W})$.

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

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

Recap

RECAP

1. **Assumption**: Nodes closer in the hierarchy/graph share similar model parameters.
2. **Model**: Incorporate the structure into $\lambda(\mathbf{W})$.

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

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

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 $\lambda(\mathbf{W})$.

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

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

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.

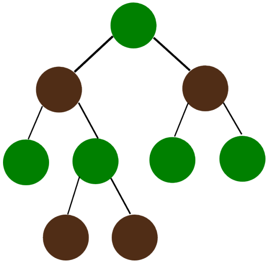
Can we update multiple blocks in parallel ?

Key point for parallelization: Parameters are only locally dependent.

1. In a hierarchy, the parameters of a node depend only parent and children.
2. In a graph, the parameters of a node depend on its neighbours.

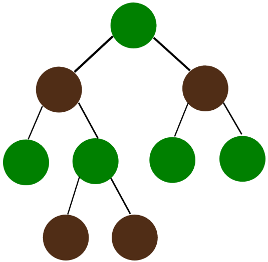
Parallelization (cont)

- 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.

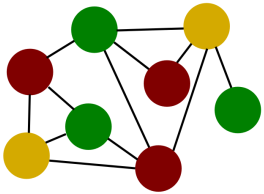


Parallelization (cont)

- 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.



Experiments

DATASETS

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

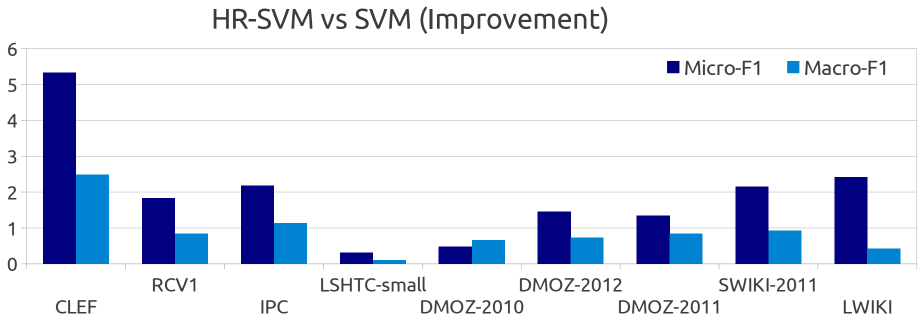
Comparison with published results

	LSHTC Published Results	HR-SVM	HR-LR
DMOZ-2010			
Macro-F_1	**34.12**	33.12	32.42
Micro-F_1	**46.76**	46.02	45.84
DMOZ-2012			
Macro-F_1	31.36	**33.05**	20.04
Micro-F_1	51.98	**57.17**	53.18
DMOZ-2011			
Macro-F_1	**26.48**	25.69	23.90
Micro-F_1	38.85	**43.73**	42.27
SWIKI-2011			
Macro-F_1	23.16	**28.72**	24.26
Micro-F_1	37.39	**41.79**	40.99
LWIKI			
Macro-F_1	18.68	**22.31**	20.22
Micro-F_1	34.67	**38.08**	37.67

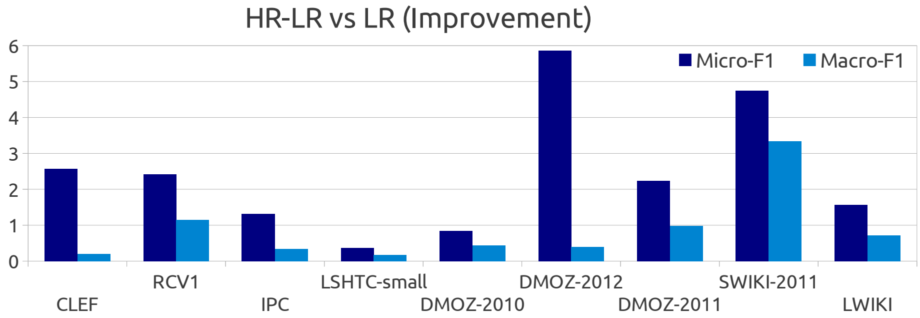
Methods for comparison

- **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) [Tsochantaridis 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

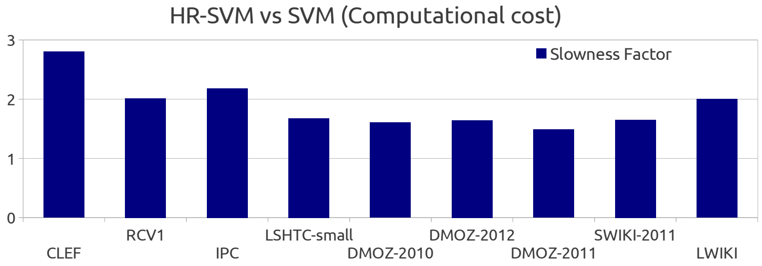
Against flat baselines

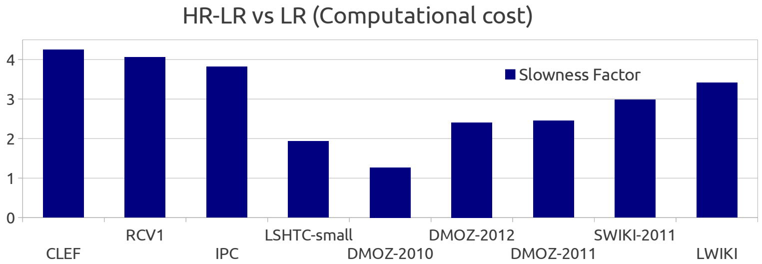


Against flat baselines



Time complexity





Conclusion

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 !

Against Hierarchical Baselines

Micro-F1 comparison

Datasets	**HR-SVM**	**TD**	**HSVM**	**OT**	**HBLR**
CLEF	80.02	70.11	79.72	73.84	**81.41**
RCV1	**81.66**	71.34	NA	NS	NA
IPC	54.26	50.34	NS	NS	**56.02**
LSHTC-small	45.31	38.48	39.66	37.12	**46.03**
DMOZ-2010	**46.02**	38.64	NS	NS	NA
DMOZ-2012	**57.17**	55.14	NS	NS	NA
DMOZ-2011	**43.73**	35.91	NA	NS	NA
SWIKI-2011	**41.79**	36.65	NA	NA	NA
LWIKI	**38.08**	NA	NA	NA	NA

[NA - Not applicable, NS - Not scalable]

Time complexity

Time (in mins)

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

Conclusion

1. A **scalable** framework that can leverage class-label dependencies.
2. and that works in practice !

Cai, L. and Hofmann, T. (2004).
Hierarchical document categorization with support vector machines.
In *CIKM*, pages 78–87. ACM.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. (2006).
Incremental algorithms for hierarchical classification.
JMLR, 7:31–54.

Dekel, O., Keshet, J., and Singer, Y. (2004).
Large margin hierarchical classification.
In *ICML*, page 27. ACM.

Dumais, S. and Chen, H. (2000).
Hierarchical classification of web content.
In *ACM SIGIR*.

Gopal, S., Yang, Y., Bai, B., and Niculescu-Mizil, A. (2012).
Bayesian models for large-scale hierarchical classification.

In *Advances in Neural Information Processing Systems 25*, pages 2420–2428.

Koller, D. and Sahami, M. (1997).
Hierarchically classifying documents using very few words.

Liu, T., Yang, Y., Wan, H., Zeng, H., Chen, Z., and Ma, W. (2005).
Support vector machines classification with a very large-scale taxonomy.
ACM SIGKDD, pages 36–43.

McCallum, A., Rosenfeld, R., Mitchell, T., and Ng, A. (1998).
Improving text classification by shrinkage in a hierarchy of classes.
In *ICML*, pages 359–367.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006).

Kernel-based learning of hierarchical multilabel classification models.
The Journal of Machine Learning Research, 7:1601–1626.

Shahbaba, B. and Neal, R. (2007).
Improving classification when a class hierarchy is available using a hierarchy-based prior.
Bayesian Analysis, 2(1):221–238.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2006).
Large margin methods for structured and interdependent output variables.
JMLR, 6(2):1453.

Yang, Y., Zhang, J., and Kisiel, B. (2003).
A scalability analysis of classifiers in text categorization.
In *SIGIR*, pages 96–103. ACM.

Zhou, D., Xiao, L., and Wu, M. (2011).

Hierarchical classification via orthogonal transfer.
Technical report, MSR-TR-2011-54.