Large Scale Hierarchical Classification: Foundations, Algorithms and Applications

Huzefa Rangwala

Department of Computer Science
MLBio+ Laboratory
Fairfax, Virginia, USA

CS 584, Fall 2016

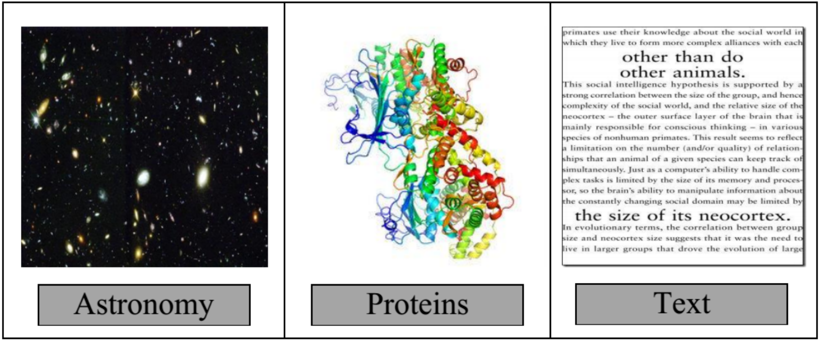
11/14/2016

Roadmap

1. Introduction and Background
   - Motivation
   - Hierarchical Classification (HC) problem description
   - Challenges
   - Methods for solving HC
2. State-of-the-Art HC Approaches
   - Parent-child regularization
   - Cost-sensitive learning
3. Learning from Multiple Hierarchies
4. Inconsistent Hierarchy
5. Conclusion

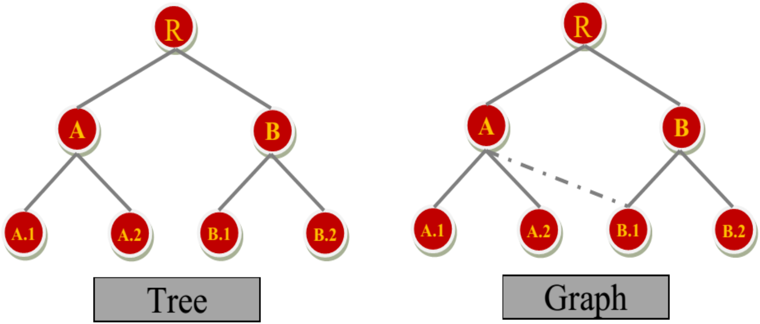
Motivation

- Exponential growth in data (image, text, video) over time
  - Big data era - megabytes & gigabytes to **terabytes** & **petabytes**
  - growth in almost all fields - astronomical, biological, web content



Data Organization

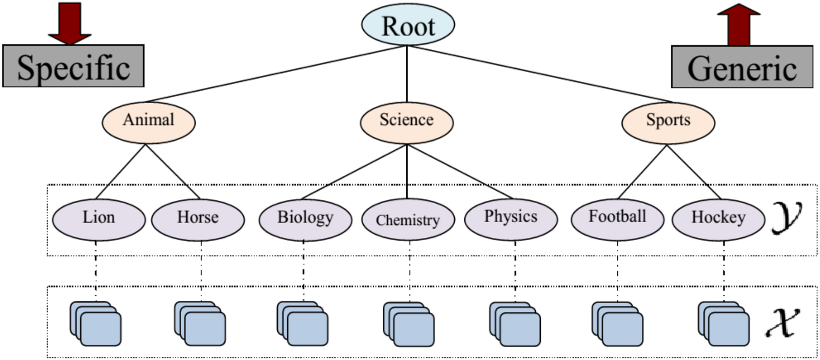
- Organize data into structure
  - tree, graph [LSHTC, BioASQ and ILSVRC challenge]



- Useful in various applications
  - query search, browsing and categorizing products

Hierarchical Structure

- Classes organized into the hierarchical structure
- **Generic (↑)** to **specific (↓)** categories in top-down order



Hierarchical Classification

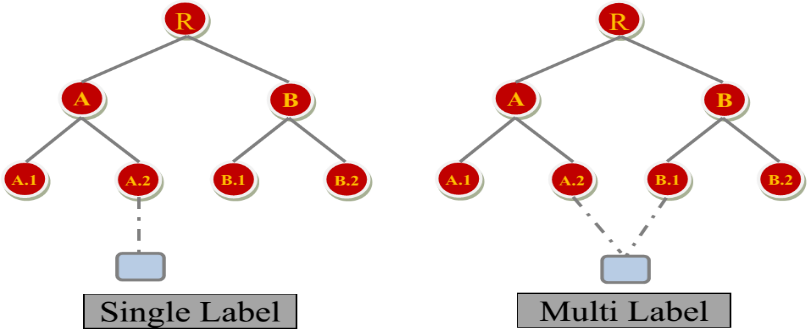
Goal

Given hierarchy of classes **exploit the hierarchical structure** to learn models and classify unlabeled test examples (instances) to one or more nodes in the hierarchy

Challenges - I

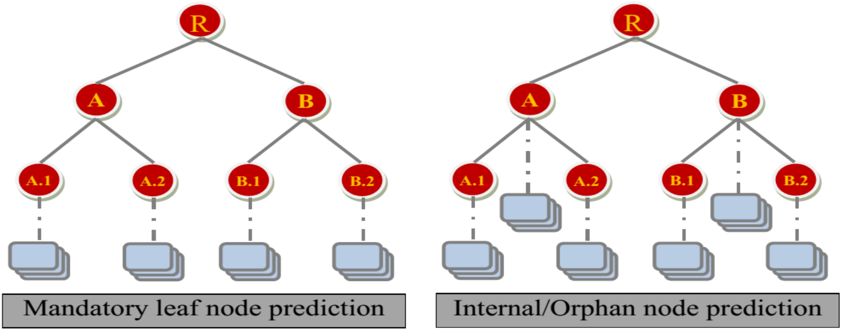
Single label vs. multi-label

- **Single label classification** - each example belongs exclusively to one class only
- **Multi-label classification** - example may belong to more than one class



Mandatory leaf node vs. internal node prediction

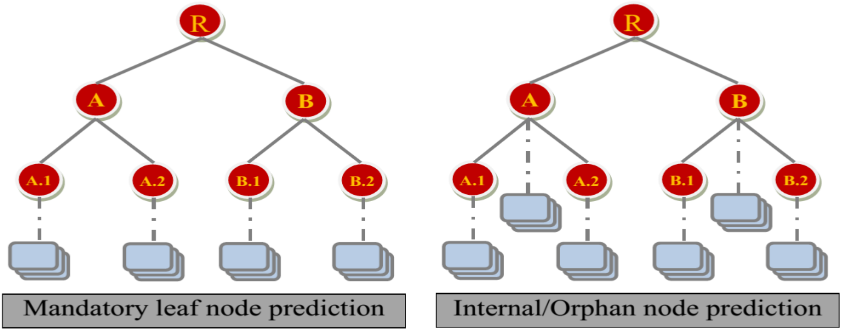
- Example may be assigned to internal nodes



Challenges - II

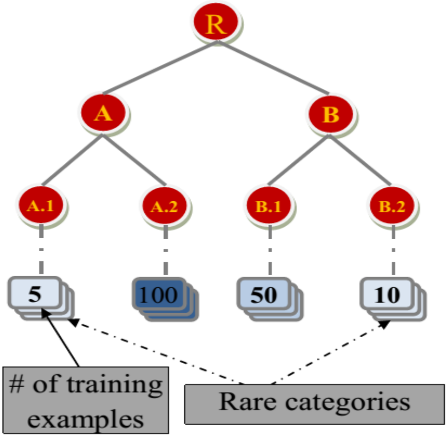
Mandatory leaf node vs. internal node prediction

- Example may be assigned to internal nodes
- **Orphan node** detection problem



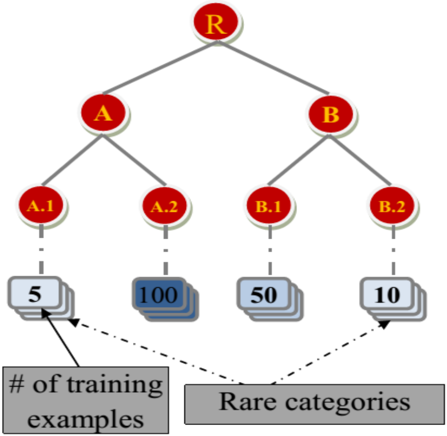
Rare categories

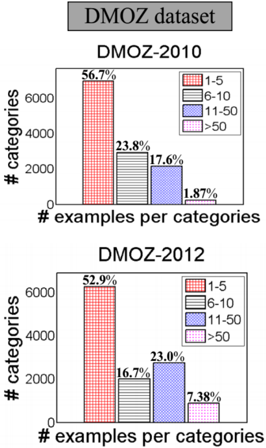
- Many classes with very few labeled examples



Rare categories

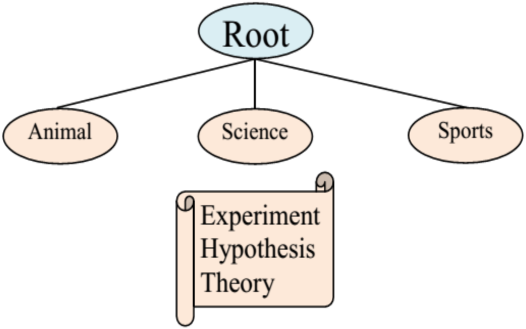
- Many classes with very few labeled examples
- More prevalent in large scale datasets - $\geq$**70%** have $\leq$**10** examples





Feature selection

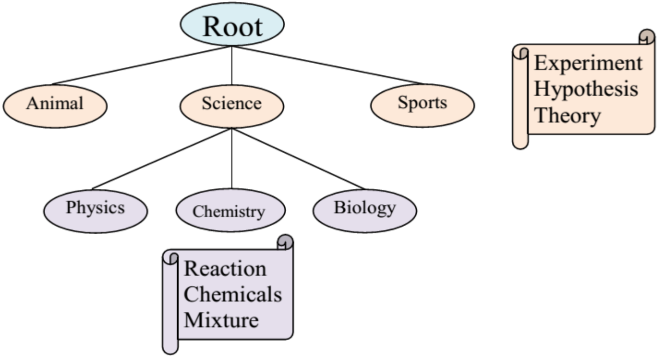
- All features are **not essential** to **discriminate** between classes
- Identify features to improve classification performance



Challenges - IV

Feature selection

- All features are **not essential** to **discriminate** between classes
- Identify features to improve classification performance



Other Challenges

- **Parameter optimization**
  - incorporate relationships (**parent-child, silings**) information

Other Challenges

- **Parameter optimization**
  - incorporate relationships (**parent-child, silings**) information

- **Scalability**
  - large # of classes, features and examples require **distributed computation**

Dataset	#Training examples	#Leaf node (classes)	#Features	#Parameters	Parameter size (approx)
DMOZ-2010	128,710	12,294	381,580	4,652,986,520	18.5 GB
DMOZ-2012	383,408	11,947	348,548	4,164,102,956	16.5 GB

Other Challenges

- **Parameter optimization**
  - incorporate relationships (**parent-child, silings**) information
- **Scalability**
  - large # of classes, features and examples require **distributed computation**

Dataset	#Training examples	#Leaf node (classes)	#Features	#Parameters	Parameter size (approx)
DMOZ-2010	128,710	12,294	381,580	4,652,986,520	18.5 GB
DMOZ-2012	383,408	11,947	348,548	4,164,102,956	16.5 GB

- **Inconsistent hierarchy**
  - not suitable for classification (more details later)

Notation

n = # of training examples (instances)	D = dimension of each instance
$\mathcal{N}$ = set of nodes in the hierarchy	L = set of leaf node (classes)
$\mathcal{C}(t)$ = children of node t	$\pi(t)$ = parent of node t

$$\mathbf{X} = \begin{array}{cc} \text{FEATURES} & \\ \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^D \\ x_2^1 & x_2^2 & \cdots & x_2^D \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^D \end{bmatrix} & \text{INSTANCES} \end{array}$$

INSTANCE MATRIX

$$\mathbf{Y} = \begin{array}{cc} \text{CLASSES} & \\ \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} & \text{INSTANCES} \end{array}$$

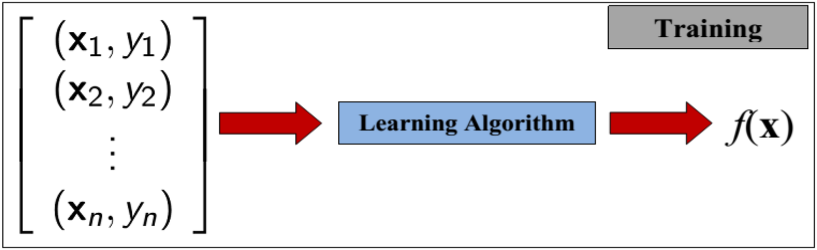
LABEL MATRIX

$$\mathbf{W} = \begin{array}{cc} \text{FEATURES} & \\ \begin{bmatrix} w_1^1 & w_1^2 & \cdots & w_1^D \\ w_2^1 & w_2^2 & \cdots & w_2^D \\ \vdots & \vdots & \ddots & \vdots \\ w_L^1 & w_L^2 & \cdots & w_L^D \end{bmatrix} & \text{CLASSES} \end{array}$$

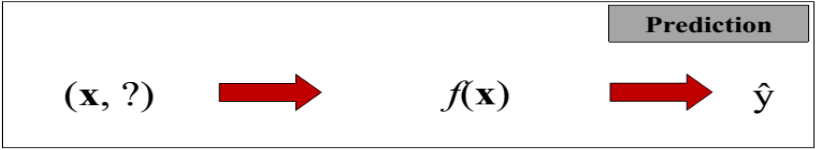
WEIGHT MATRIX

Classification

Training - Learn mapping function using training data



Testing - Predict the label of test example



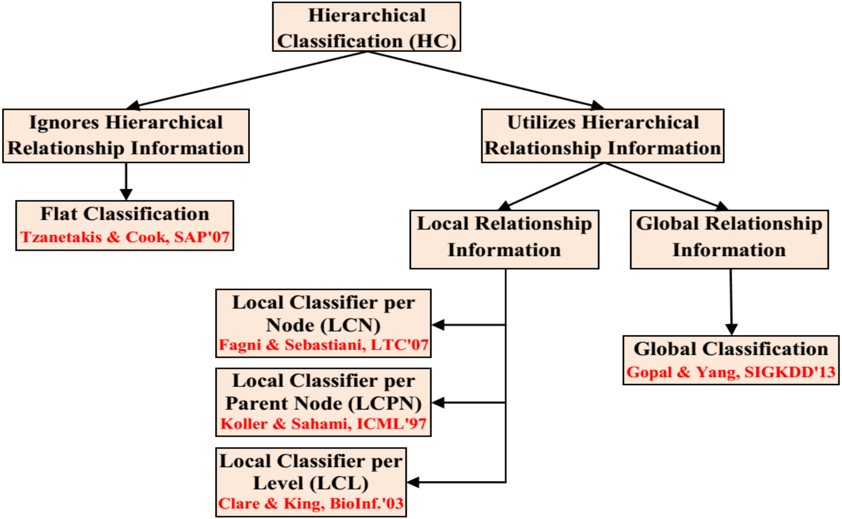
Learning Algorithm: General Formulation

Combination of two terms:

1. **Empirical loss** - controls how well the learnt models fits the training data
2. **Regularization** - prevent models from over-fitting and encodes additional information such as hierarchical relationships

$$\min_{\mathbf{W}} \underbrace{\mathcal{L}(f(\mathbf{X}, \mathbf{W}), \mathbf{Y})}_{\textbf{Empirical Loss}} + \underbrace{\lambda\Omega(\mathbf{W})}_{\textbf{Regularization}}$$

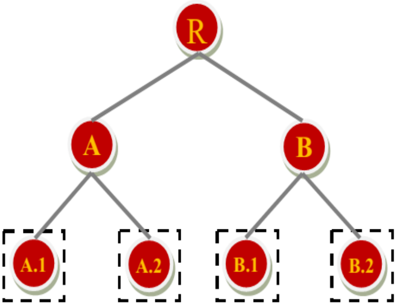
Different Approaches for Solving HC Problem



Flat Classification Approach

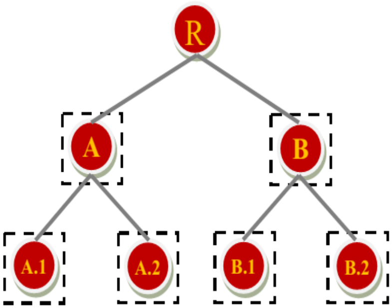
- Simplest method (ignores hierarchy)
- Learn discriminant classifiers for each leaf node in the hierarchy
- Unlabeled test example classified using the rule:

$$\boxed{\hat{y} = \underset{y \in \mathcal{Y}}{\textbf{arg max}}\ f(\mathbf{x}, y|\mathbf{w})}$$



Local Classifier per Node (LCN)

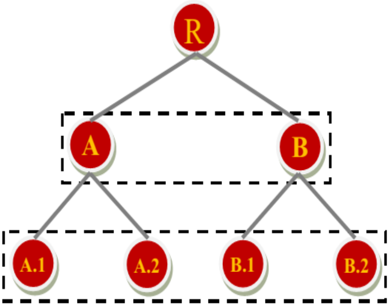
- Learn binary classifiers for all non-root nodes
- Goal is to effectively discriminate between the siblings
- Top-down approach is followed for classifying unlabeled test examples



Local Classification Approach - II

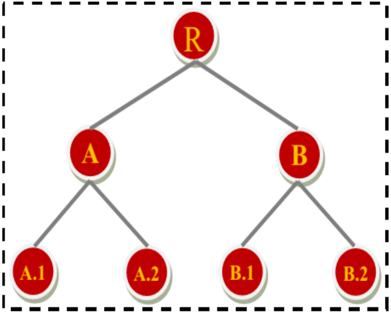
Local Classifier per Level (LCL)

- Learn multi-class classifiers for all levels in the hierarchy
- Least popular among local approaches
- Prediction inconsistency may occur and hence post-processing step is required



Global Classification Approach

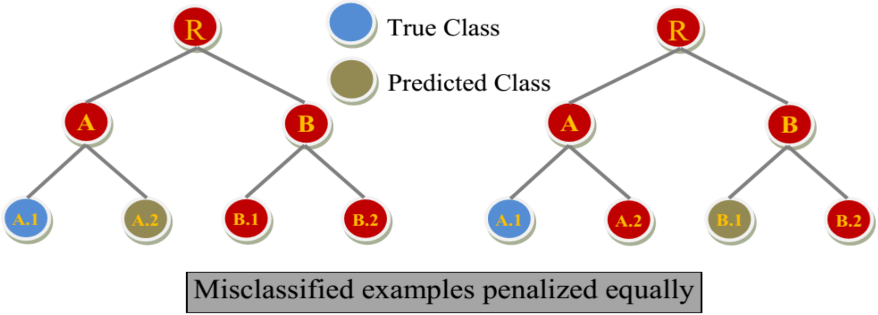
- Learn global function considering all hierarchical relationships
- Often referred as Big-Bang approach
- Unlabeled test instance is classified using an approach similar to flat or local methods



Evaluation Metrics - I

Flat evaluation measures

- Misclassifications treated equally

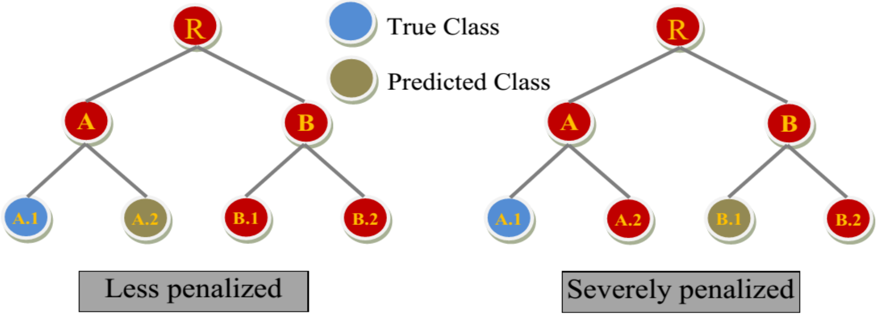


- Common evaluation metrics:
  - **Micro-F1** - gives equal weightage to all examples, dominated by common class
  - **Macro-F1** - gives equal weightage to each class

Evaluation Metrics - II

Hierarchical evaluation measures

- Hierarchical distance between the true and predicted class taken into consideration for performance evaluation

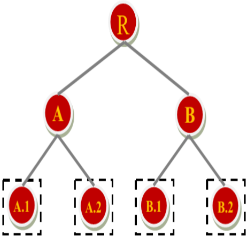


- Common evaluation metrics:
  - **Hierarchical-F1** - common ancestors between true and predicted class
  - **Tree Error** - average hierarchical distance b/w true and predicted class

Multi-Task Learning (MTL)

- Involves **joint training** of multiple related tasks to improve generalization performance
- Independent learning problems can utilize the **shared knowledge**
- Exploits **inductive biases** that are helpful to all the related tasks
  - similar set of parameters
  - common feature space

Motivation



- Traditional approach learn classifiers for each leaf node (task) to discriminate one class from other

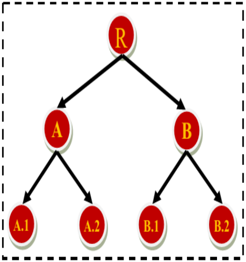
$$\min_{\mathbf{w}_t} \ \frac{1}{2}||\mathbf{w}_t||_2^2 + C\sum_{i=1}^{n}\left[1 - \mathbf{Y}_{it}\mathbf{w}_t^T\mathbf{x}_i\right]_+$$

- Works well if:
    - Dataset is small
    - Balanced
    - Sufficient positive examples per class to learn generalized discriminant function

Drawbacks

- Real world datasets suffers from rare categories issue
  Remember: **70% classes have less than 10 examples per class**
- Large number of classes (**scalability issue**)

Motivation - II



- Can we improve the performance of **data sparse** leaf nodes by taking advantage of **data rich** nodes at higher levels?
- Incorporate **inter-class dependencies** to improve classification
  - examples belonging to *Soccer* category is less likely to belong to *Software* category

$$\min_{\mathbf{w}_t} \ \frac{1}{2}||\mathbf{w}_t - \mathbf{w}_{\pi(t)}||_2^2 + C \sum_{k \in \mathcal{C}(t)} \sum_{i=1}^{n} \left[1 - \mathbf{Y}_{ik} \mathbf{w}_t^T \mathbf{x}_i\right]_+$$

Objective

- How to effectively incorporate the hierarchical relationships into the objective function to improve generalization performance
- Make it scalable for larger datasets

Proposed Formulation

- Enforces model parameters (weights) to be similar to the parent in regularization
- Proposed state-of-the-art: HR-SVM and HR-LR global formulation

HR-SVM

$$\min_{\mathbf{W}} \sum_{t \in \mathcal{N}} \frac{1}{2} ||\mathbf{w}_t - \mathbf{w}_{\pi(t)}||_2^2 + C \sum_{k \in L} \sum_{i=1}^{n} \left[1 - \mathbf{Y}_{ik} \mathbf{w}_k^T \mathbf{x}_i\right]_+$$

Internal Node

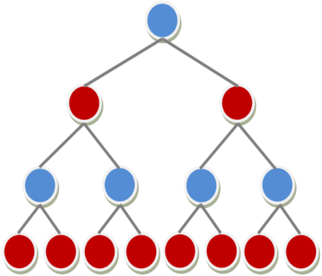
$$\min_{\mathbf{w}_t} \frac{1}{2} ||\mathbf{w}_t - \mathbf{w}_{\pi(t)}||_2^2 + \frac{1}{2} \sum_{c \in \mathcal{C}(t)} ||\mathbf{w}_c - \mathbf{w}_t||_2^2$$

Leaf Node

$$\min_{\mathbf{w}_t} \frac{1}{2} ||\mathbf{w}_t - \mathbf{w}_{\pi(t)}||_2^2 + \frac{1}{2} \sum_{i=1}^{n} \left[1 - \mathbf{Y}_{it} \mathbf{w}_t^T \mathbf{x}_i\right]_+$$

Proposed Parallel Implementation

- Each node is independent of all other nodes except its neighbours
- Objective function is block separable. Therefore, Parallel Block Coordinate Descent (CD) can be used for optimization



Alternate level optimized in each iteration

1. Fix odd-levels parameters, optimize even-levels in parallel
2. Fix even-levels parameters, optimize odd-levels in parallel
3. Repeat untill convergence

- Extended to graph by first finding the minimum graph coloring [Np-hard] and repeatedly optimizing nodes with the same color in parallel during each iteration

Experiments

Dataset description

- Wide range of single and multi-label dataset with varying number of features and categories were used for model evaluation

Datasets	# Features	# Categories	Type	Avg # labels (per instance)
CLEF	89	87	Single-label	1
RCV1	48,734	137	Multi-label	3.18
IPC	541,869	552	Single-label	1
DMOZ-SMALL	51,033	1,563	Single-label	1
DMOZ-2010	381,580	15,358	Single-label	1
DMOZ-2012	348,548	13,347	Single-label	1
DMOZ-2011	594,158	27,875	Multi-label	1.03
SWIKI-2011	346,299	50,312	Multi-label	1.85
LWIKI	1,617,899	614,428	Multi-label	3.26

Table: **Dataset statistics**

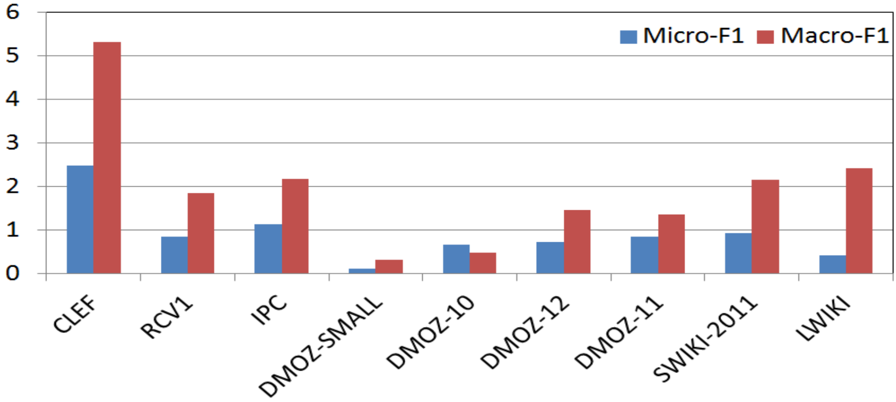


Figure: **Performance improvement: HR-SVM vs. SVM**

Hierarchical Baselines Comparison

Datasets	HR-SVM	HR-LR	TD	HSVM	OT	HBLR
CLEF	80.02	80.12	70.11	79.72	73.84	**81.41**
RCV1	**81.66**	81.23	71.34	NA	NS	NA
IPC	54.26	55.37	50.34	NS	NS	**56.02**
DMOZ-SMALL	45.31	45.11	38.48	39.66	37.12	**46.03**
DMOZ-2010	**46.02**	45.84	38.64	NS	NS	NS
DMOZ-2012	**57.17**	53.18	55.14	NS	NS	NS
DMOZ-2011	**43.73**	42.27	35.91	NA	NS	NA
SWIKI-2011	**41.79**	40.99	36.65	NA	NA	NA
LWIKI	**38.08**	37.67	NA	NA	NA	NA

[**NA - Not Applicable; NS - Not Scalable**]

Table: **Micro-F1 performance comparison**

Runtime Comparison - hierarchical baselines

Datasets	HR-SVM	HR-LR	TD	HSVM	OT	HBLR
CLEF	0.42	1.02	0.13	3.19	1.31	3.05
RCV1	0.55	11.74	0.21	NA	NS	NA
IPC	6.81	15.91	2.21	NS	NS	31.20
DMOZ-SMALL	0.52	3.73	0.11	289.60	132.34	5.22
DMOZ-2010	8.23	123.22	3.97	NS	NS	NS
DMOZ-2012	36.66	229.73	12.49	NS	NS	NS
DMOZ-2011	58.31	248.07	16.39	NA	NS	NA
SWIKI-2011	89.23	296.87	21.34	NA	NA	NA
LWIKI	2230.54	7282.09	NA	NA	NA	NA

[**NA - Not Applicable; NS - Not Scalable**]

Table: **Training runtime comparison (in mins) but on several nodes.**

Cost-sensitive Learning, Charuvaka & Rangwala, ECML'15

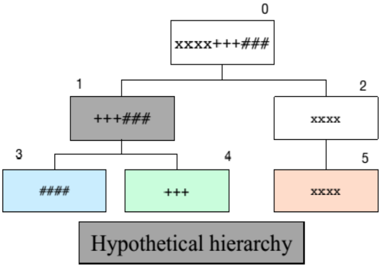
Motivation

- Drawbacks of Recursive Regularization
  - scalable, but more expensive to train than flat classification
  - requires specialized implementation and communication between processing node
  - Does not deal with class imbalance directly

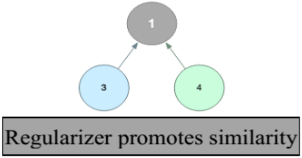
Objective

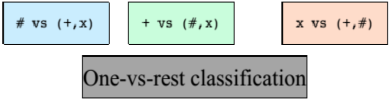
- Decouple models so that they can be trained in parallel without dependencies between models
- Account for class imbalance in the optimization framework

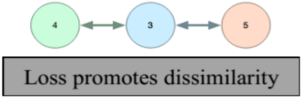
Hierarchical Regularization Re-examination - I



$$\min_{\mathbf{W}} \mathcal{L}(f(\mathbf{X}, \mathbf{W}), \mathbf{Y}) + \lambda\Omega(\mathbf{W})$$







Hierarchical Regularization Re-examination - II

- Opposing learning influences:
  - **loss term** - model for a node is forced to be dissimilar to all other nodes
  - **regularization term** - model is forced to be similar to its neighbors; greater similarity to nearer neighbors
- Resultant effect:
  - Mistakes on negative examples that come from near nodes is less severe than those coming from far nodes while still taking advantage of the hierarchy

Cost-sensitive Loss

- Consider the loss term for class "t" which is separable over examples

$$\sum_i loss(y_i, \mathbf{w}_i^T \mathbf{x}_i)$$

- Each loss value is multiplied by importance of the example for this class

$$\sum_i loss(y_i, \mathbf{w}_i^T \mathbf{x}_i) \times \phi(t, y_i)$$

- This is an example of "instance-based" cost sensitive learning

$$c_i^t = \phi(t, y_1)$$

Hierarchical Costs

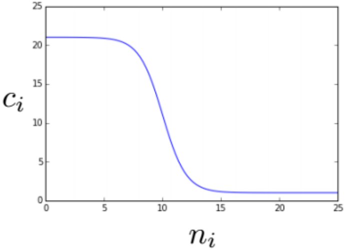
How to define costs based on hierarchy?

- **Tree Distance (TrD)** - undirected graph distance between between nodes
- **Number Common Ancestors (NCA)** - the number of ancestors in common to target class and class label
- **Exponentiated Tree Distance (ExTrD)** - squash tree distance into a suitable range using validation

Imbalance Costs

- Using the same formulation of cost-sensitive learning, data imbalance can also be addressed
- Due to very large skew, inverse class size can result in extremely large weights. Fix using squashing function shown in Fig.
- Multiply to combine with Hierarchical costs

$$c_i = 1 + L/[1 + exp|n - n_0|]$$



n_i = num examples
n_0, L = user defined constants

Experiments

Dataset

- For comparison purpose same dataset has been used as proposed in the paper [**Gopal and Yang, SIGKDD'13**]

Comparison Methods

Flat baseline

- **LR** - one-vs-rest binary logistic regression is used in the conventional flat classification setting

Hierarchical baselines

- **Top-down Logistic Regression (TD-LR)** - one-vs-rest multi-class classifier trained at each internal node
- **HR-LR** [**Gopal and Yang, SIGKDD'13**] - a recursive regularization approach based on hierarchical relationships

Results (Hierarchical Costs)

Datasets		Micro-F1 (↑)	Macro-F1 (↑)	hF1 (↑)	TE (↓)
CLEF	LR	79.82	53.45	85.24	0.994
	TrD	80.02	55.51	**85.39**	0.984
	NCA	80.02	57.48	85.34	0.986
	ExTrD	**80.22**	**57.55**†	85.34	**0.982**
DMOZ-SMALL	LR	46.39	30.20	67.00	3.569
	TrD	**47.52**‡	**31.37**‡	**68.26**	**3.449**
	NCA	47.36‡	31.20‡	68.12	3.460
	ExTrD	47.36‡	31.19‡	68.20	3.456
IPC	LR	55.04	48.99	72.82	1.974
	TrD	55.24‡	50.20‡	73.21	1.954
	NCA	**55.33**‡	**50.29**‡	**73.28**	**1.949**
	ExTrD	55.31‡	**50.29**‡	73.26	1.951
RCV1	LR	78.43	60.37	80.16	0.534
	TrD	79.46‡	60.61	82.83	0.451
	NCA	**79.74**‡	60.76	**83.11**	**0.442**
	ExTrD	79.33‡	**61.74**†	82.91	0.466

Table: **Performance comparison of hierarchical costs**

Results (Imbalance Costs)

Datasets		Micro-F1 (↑)	Macro-F1 (↑)	hF1 (↑)	TE (↓)
CLEF	IMB + LR	79.52	53.11	85.19	1.002
	IMB + TrD	79.92	52.84	85.59	0.978
	IMB + NCA	79.62	51.89	85.34	0.994
	IMB + ExTrD	**80.32**	**58.45**	**85.69**	**0.966**
DMOZ-SMALL	IMB + LR	48.55‡	32.72‡	68.62	3.406
	IMB + TrD	**49.03**‡	33.21‡	69.41	3.334
	IMB + NCA	48.87‡	33.27‡	69.37	3.335
	IMB + ExTrD	**49.03**‡	**33.34**‡	**69.54**	**3.322**
IPC	IMB + LR	55.04	49.00	72.82	1.974
	IMB + TrD	55.60‡	**50.45**†	73.56	1.933
	IMB + NCA	55.33	50.29	73.28	1.949
	IMB + ExTrD	**55.67**‡	50.42	**73.58**	**1.931**
RCV1	IMB + LR	78.59‡	60.77	81.27	0.511
	IMB + TrD	**79.63**‡	61.04	**83.13**	**0.435**
	IMB + NCA	79.61	61.04	82.65	0.458
	IMB + ExTrD	79.22	**61.33**	82.89	0.469

Table: **Peformance comparison with imbalance cost included**

Results (our best with other methods)

Datasets		Micro-F1 (↑)	Macro-F1 (↑)	hF1 (↑)	TE (↓)
CLEF	TD-LR	73.06	34.47	79.32	1.366
	LR	79.82	53.45	85.24	0.994
	HR-LR	80.12	55.83	NA	NA
	HierCost	**80.32**	**58.45**†	**85.69**	**0.966**
DMOZ-SMALL	TD-LR	40.90	24.15	**69.99**	**3.147**
	LR	46.39	30.20	67.00	3.569
	HR-LR	45.11	28.48	NA	NA
	HierCost	**49.03**‡	**33.34**‡	69.54	3.322
IPC	TD-LR	50.22	43.87	69.33	2.210
	LR	55.04	48.99	72.82	1.974
	HR-LR	55.37	49.60	NA	NA
	HierCost	**55.67**‡	**50.42**†	**73.58**	**1.931**
RCV1	TD-LR	77.85	57.80	**88.78**	0.524
	LR	78.43	60.37	80.16	0.534
	HR-LR	**81.23**	55.81	NA	NA
	HierCost	79.22‡	**61.33**	82.89	**0.469**

Table: **Performance comparison of HierCost with other baseline methods**

Runtime comparison

Datasets	TD-LR	LR	HierCost
CLEF	<1	<1	<1
DMOZ-SMALL	4	41	40
IPC	27	643	453
RCV1	20	29	48
DMOZ-2010	196	15191	20174
DMOZ-2012	384	46044	50253

Table: **Total training runtimes (in mins)**

Software

- Freely available for research and education purpose at:

  https://cs.gmu.edu/~mlbio/HierCost/

- Software: implemented in python using **scikit-learn** machine learning and **svmlight-loader** package
- Other prerequisite package:
  - numpy
  - scipy
  - networkx
  - pandas

Learning using Multiple Hierarchies (MTL), Charuvaka and Rangwala, ICDM'12

Motivation

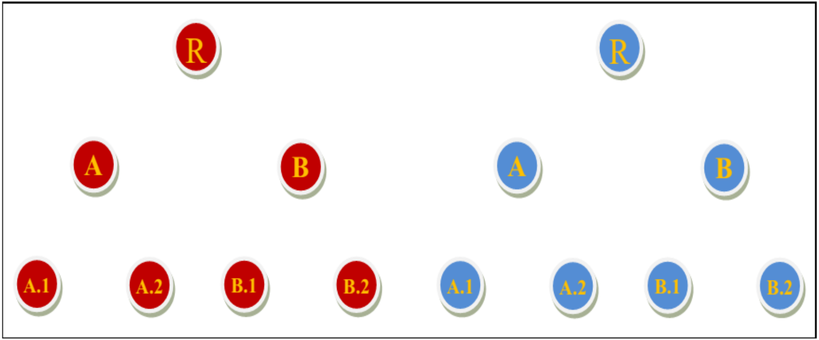
- Hierarchies are so common that sometimes multiple hierarchies classify similar data
- Heterogenous label view provide additional knowledge which should be exploited by learners
- Examples
  - **protein structure classification** - several hierarchical schemes for organizing proteins based on curation process or 3D structure
  - **web-page classification** - several hierarchy exist for categorizing such as DMOZ and wikipedia datasets

Objective

- Utilize multiple hierarchical label views in multi-task learning context to improve classification performance

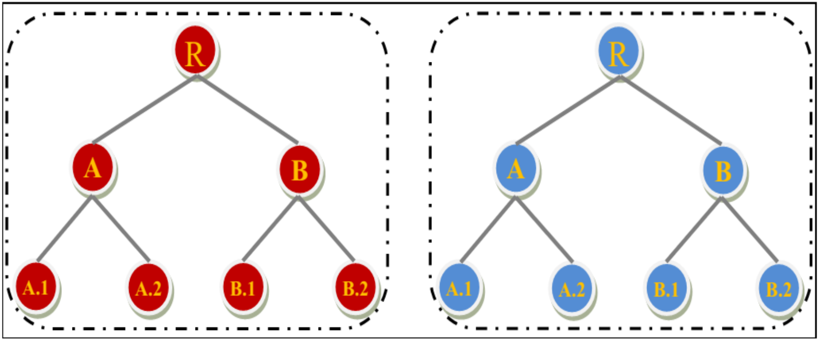
Three Different Learning Settings - I

(i) **Single Task Learning (STL)** - each task model parameters learned independently



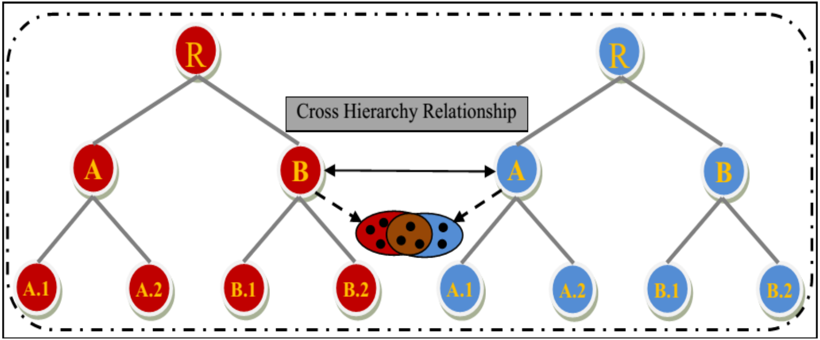
Three Different Learning Settings - II

(ii) **Single Hierarchy Multi-Task Learning (SHMTL)** - relationship between tasks within a hierarchy are combined individually



Three Different Learning Settings - III

(iii) **Multiple Hierarchy Multi-Task Learning (MHMTL)** - relationship between tasks from different hierarchies are extracted using common examples



MTL Formulations

- General MTL formulation:

$$\underbrace{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\mathcal{L}(f(\mathbf{x}_i,\mathbf{W}_t),y_i)}_{\textbf{Empirical Loss}} + \underbrace{\lambda\Omega\left(\{\mathbf{W}_t\}_{t=1}^{T}\right)}_{\textbf{Regularization}}$$

- Different MTL formulation based on regularization:
  - **Sparse** - All tasks share a single set of useful features

    $$\Omega(\mathbf{W}) = ||\mathbf{W}||_{2,1}$$

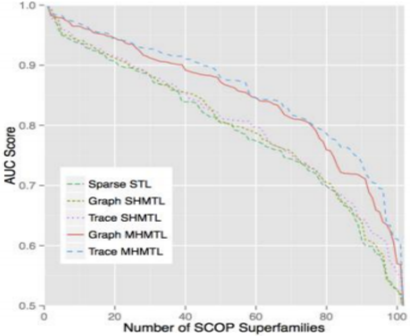
  - **Graph Regularization** - Related tasks have similar parameters

    $$\Omega(\mathbf{W}) = \sum_{(a,b)\in\mathcal{E}} ||\mathbf{W}_a - \mathbf{W}_b||_2^2$$

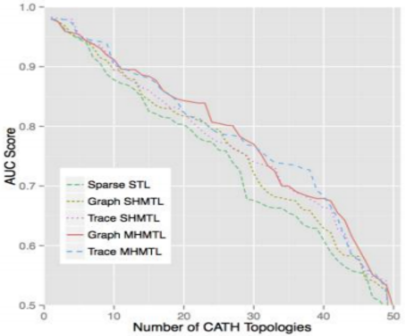
  - **Trace** - Task parameters are drawn from a low dimensional sub-space

    $$\Omega(\mathbf{W}) = ||\mathbf{W}||_* = \mathit{TraceNorm}(\mathbf{W})$$

Performance: AUC Comparison

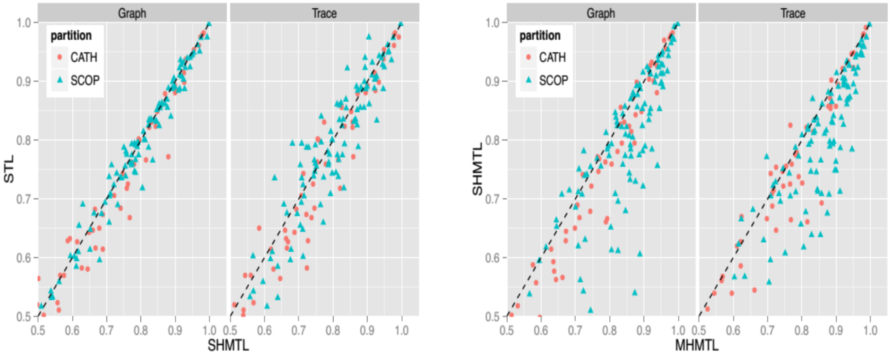


SCOP



CATH

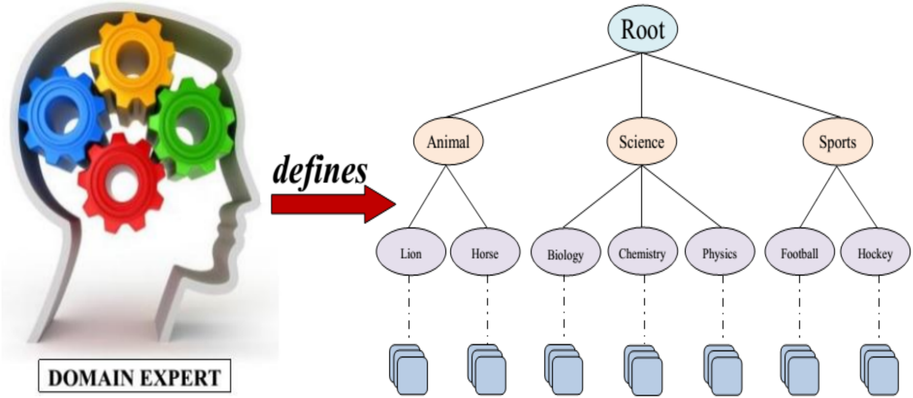
STL, SHMTL and MHMTL Comparison



Motivation

Predefined Hierarchy

- Hierachy defined by the domain experts
- Reflects human-view of the domain - may not be optimal for machine learning classification algorithms



Case Study

Question

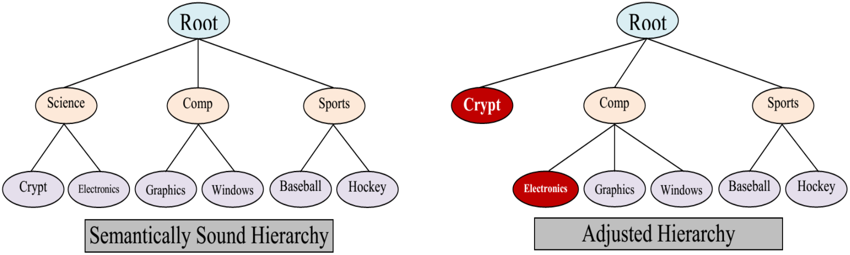
- Can we trust the predefined expert's hierarchy for achieving the good classification performance?
- Can we tweak (adjust) the hierarchy to improve the performance?

Question

- Can we trust the predefined expert's hierarchy for achieving the good classification performance?
- Can we tweak (adjust) the hierarchy to improve the performance?

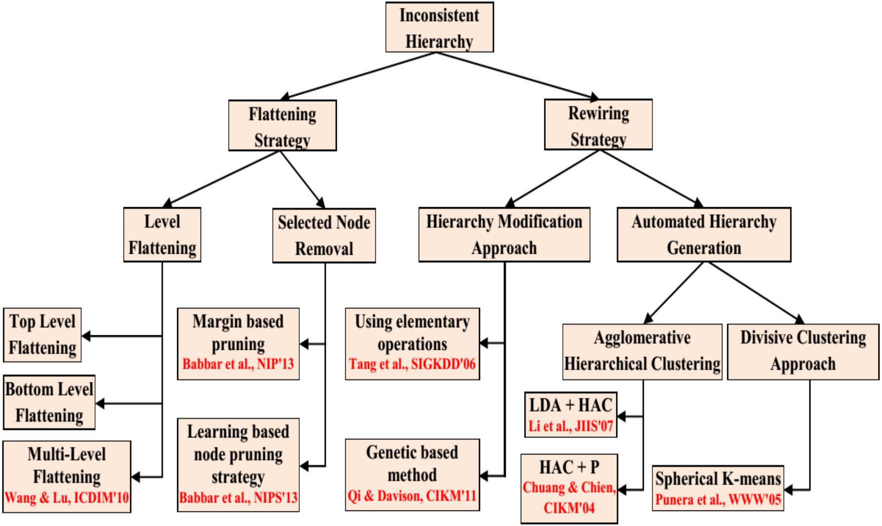
Answer

- Case study on subset of newsgroup dataset



"Adjusted hierarchy classification performance **comparatively better** than semantically sound hierarchy"

Literature Overview



Hierarchy Modification Strategy

Motivation

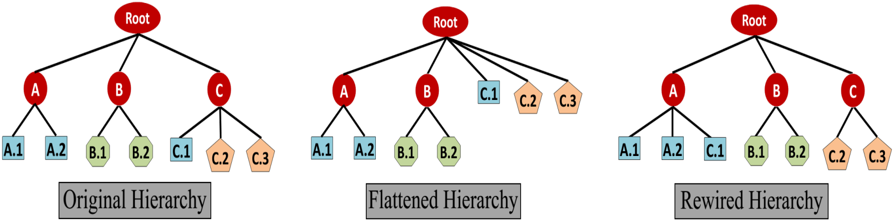
- For large scale datasets top-down (TD) hierarchical models are preferred over flat models due to computational benefit (training and prediction time)
- TD models performance suffers due to **error propagation** *i.e.* compounding of errors from misclassifications at higher levels which cannot be rectified at the lower levels

Objective

- Modify predefined hierarchy by removing (flattening) or rewiring inconsistent nodes to improve the classification performance of TD models
- Reduces top-down error propagation due to less number of decisions for classifying unlabeled examples

Flattening and Rewiring Strategy

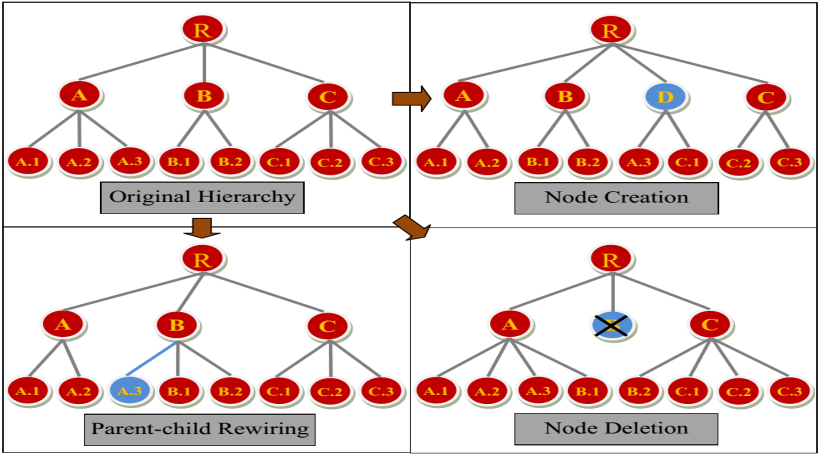
- Flattening strategy although useful upto certain extent has few limitations
    - Inability to deal with inconsistencies in different branches of the hierarchy



- Rewiring strategy can be used to resolve inconsistencies that occurs in different branch

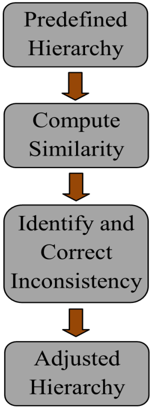
Proposed Rewiring Strategy

- Elementary operation: **node creation, parent-child rewiring, node deletion**



Proposed Rewiring Strategy Algorithm

- Filter based approach for hierarchy modification



Input: Predefined hierarchy (H_0), Train data ($\mathcal{D}_t$)

1. Compute pairwise similarity between classes defined in H_0 on $\mathcal{D}_t$
2. Group together most similar classes
3. Identify inconsistencies within the hierarchy
4. Apply elementary operations: node creation or parent-child rewiring to correct inconsistencies and obtain new hierarchy H_1
5. Perform post-processing step (node deletion) on H_1 to obtain new hierarchy H_2
6. Train and evaluate hierarchical classification models on H_2

Performance Results

Datasets		Flattening	Rewiring	
		Best TD Model (Flattening)	Tang et al.	Proposed Filter Model
CLEF	$\mu F_1(\uparrow)$	77.14 (0.01)	**78.12 (0.16)**	78.00 (0.22)
	$MF_1(\uparrow)$	46.54 (0.06)	**48.83**‡ **(0.08)**	47.10 (0.03)
	$hF_1(\uparrow)$	79.06 (0.01)	**81.43 (0.03)**	80.14 (0.02)
DIATOMS	$\mu F_1(\uparrow)$	61.31 (0.53)	**62.34**‡ **(0.28)**	62.05‡ (0.10)
	$MF_1(\uparrow)$	51.85 (0.23)	**53.81**‡ **(0.11)**	52.14‡ (0.14)
	$hF_1(\uparrow)$	62.80 (0.04)	**64.28 (0.22)**	63.24 (0.13)
IPC	$\mu F_1(\uparrow)$	52.30 (0.12)	53.94† (0.24)	**54.28**† **(0.18)**
	$MF_1(\uparrow)$	45.65 (0.11)	**46.10**† **(0.21)**	46.04† (0.22)
	$hF_1(\uparrow)$	64.73 (0.12)	67.23 (0.24)	**68.34 (0.18)**
DMOZ-SMALL	$\mu F_1(\uparrow)$	46.61 (0.28)	NS	**48.25**‡ **(0.13)**
	$MF_1(\uparrow)$	31.26 (0.64)		**33.92**‡ **(0.22)**
	$hF_1(\uparrow)$	63.37 (0.44)		**66.18 (0.15)**
DMOZ-2010	$\mu F_1(\uparrow)$	42.37 (0.27)	NS	**43.10 (0.28)**
	$MF_1(\uparrow)$	30.11 (0.64)		**31.21 (0.34)**
DMOZ-2012	$\mu F_1(\uparrow)$	50.64 (0.22)	NS	**51.82 (0.02)**
	$MF_1(\uparrow)$	30.58 (0.28)		**31.24 (0.12)**
	$hF_1(\uparrow)$	73.19 (0.02)		**74.21 (0.03)**

Runtime Comparison

Datasets	Flattening	Rewiring	
	Best TD Model (Flattening)	Tang et al.	Proposed Filter Model
CLEF	3.5	59	7.5
DIATOMS	10	268	24
IPC	830	26432	1284
DMOZ-SMALL	65	NS	168
DMOZ-2010	25600	NS	42000
DMOZ-2012	63000	NS	94800

Table: **Total training runtimes (in mins)**

Conclusion

- Large scale hierarchical classification is an important research problem in machine learning community due to its wide applicability across several domains
- Discussed various challenges associated with the hierarchical classification
- Discussed various state-of-the-art existing approaches;
- Discussed Multiple Hierarchy MTL and Approaches for Resolving Inconsistencies
- Emerging topics:
    - Large-scale classification with deep hierarchies
    - Orphan node prediction

References - I

- Gopal, Siddharth, and Yiming Yang. "Recursive regularization for large-scale classification with hierarchical and graphical dependencies." SIGKDD, 2013.
- Charuvaka, Anveshi, and Huzefa Rangwala. "HierCost: Improving Large Scale Hierarchical Classification with Cost Sensitive Learning." ECML, 2015.
- Tang, Lei, Jianping Zhang, and Huan Liu. "Acclimatizing taxonomic semantics for hierarchical content classification." SIGKDD, 2006.
- Li, Tao, Shenghuo Zhu, and Mitsunori Ogihara. "Hierarchical document classification using automatically generated hierarchy." Journal of Intelligent Information Systems 29.2 (2007): 211-230.
- Charuvaka, Anveshi, and Huzefa Rangwala. "Multi-task learning for classifying proteins using dual hierarchies." ICDM, 2012.
- Punera, Kunal, Suju Rajan, and Joydeep Ghosh. "Automatically learning document taxonomies for hierarchical classification." WWW, 2005.
- Qi, Xiaoguang, and Brian D. Davison. "Hierarchy evolution for improved classification." CIKM, 2011.
- Bennett, Paul N., and Nam Nguyen. "Refined experts: improving classification in large taxonomies." SIGIR, 2009.

References - II

- Silla Jr, Carlos N., and Alex A. Freitas. "A survey of hierarchical classification across different application domains." DMKD, 2011.
- Naik, Azad, A. Charuvaka, and H. Rangwala. "Classifying documents within multiple hierarchical datasets using multi-task learning." ICTAI, 2013.
- Babbar, Rohit, et al. "On flat versus hierarchical classification in large-scale taxonomies." NIPS, 2013.
- Wang, Xiao-Lin, and Bao-Liang Lu. "Flatten hierarchies for large-scale hierarchical text categorization." ICDIM, 2010.
- Chuang, Shui-Lung, and Lee-Feng Chien. "A practical web-based approach to generating topic hierarchy for text segments." CIKM, 2004.
- Fagni, Tiziano, and Fabrizio Sebastiani. "On the selection of negative examples for hierarchical text categorization." LTC, 2007.
- Clare, Amanda, and Ross D. King. "Predicting gene function in Saccharomyces cerevisiae." Bioinformatics, 2003.
- Koller, Daphne, and Mehran Sahami. "Hierarchically Classifying Documents Using Very Few Words." ICML, 1997.

References - III

- Xue et al. "Deep classification in large-scale text hierarchies." SIGIR, 2008.
- Tzanetakis, G., and P. Cook. "Musical genre classification of audio signals." IEEE transactions on Speech and Audio Processing, 2007.
- Gopal, Siddharth, et al. "Bayesian models for large-scale hierarchical classification." NIPS, 2012.
- Xiao, Lin, Dengyong Zhou, and Mingrui Wu. "Hierarchical classification via orthogonal transfer." ICML, 2011.
- Naik, Azad, and Huzefa Rangwala. "A ranking-based approach for hierarchical classification." DSAA, 2015.
- Tsochantaridis, Ioannis, et al. "Large margin methods for structured and interdependent output variables." JMLR, 2005.
- Liu, Tie-Yan, et al. "Support vector machines classification with a very large-scale taxonomy." SIGKDD, 2005.
- Caruana, Rich. "Multitask learning." Machine learning, 1997.
- Anveshi Charuvaka and Huzefa Rangwala. "Approximate block coordinate descent for large scale hierarchical classification." SAC, 2015.

References - IV

- Dumais, Susan, and Hao Chen. "Hierarchical classification of Web content." SIGIR, 2000.
- Mineiro, Paul, and Karampatziakis, Nikos. "A Hierarchical Spectral Method for Extreme Classification." eprint arXiv:1511.03260 (NIPS workshop), 2015.
- Choromanska, Anna, et al. "Extreme Multi Class Classification." NIPS Workshop: eXtreme Classification, 2013.
- McCallum, Andrew, et al. "Improving Text Classification by Shrinkage in a Hierarchy of Classes." ICML, 1998.
- Babbar, Rohit, et al. "Maximum-margin framework for training data synchronization in large-scale hierarchical classification." NIP, 2013.
- Choromanska, Anna E., and John Langford. "Logarithmic time online multiclass prediction." NIPS, 2015.
- Prabhu, Yashoteja, and Manik Varma. "FastXML: a fast, accurate and stable tree-classifier for extreme multi-label learning." SIGKDD, 2014.
- Bhatia, Kush, et al. "Sparse Local Embeddings for Extreme Multi-label Classification." NIPS, 2015.

References - V

- Naik, A., and Rangwala, H. "Filter based taxonomy modification for improving hierarchical classification." http://arxiv.org/abs/1603.00772, 2016.
- Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." PAMI, 2005.
- Ding, Chris, and Hanchuan Peng. "Minimum redundancy feature selection from microarray gene expression data." Journal of bioinformatics and computational biology, 2005.

Thanks

Ph.D. Students:

Anveshi Charuvaka

Azad Naik

Slides available for download at:

https://cs.gmu.edu/~mlbio/sdm2016tutorial.html