A Regularization Framework for Large-scale Hierarchical Classification

Siddharth Gopal (CMU)
Yiming Yang (CMU)
Alexandru Niculescu-Mizil (NEC Labs, Princeton)

Talk Outline

- Introduction
  - Problem Notation
  - Challenges
- Proposed Solution
  - Formulation
  - Optimization
- Results

Classification

GIVEN

Class-label

Labeled Training Data $D = \{(x_i, t_i)\}_{i=1}^{i=N}, x_i \epsilon\, \Re^d, t_i \epsilon\, \mathcal{N}$

Webpage, Text Article etc

All class-labels

LEARN

Mapping Function $F : x \rightarrow t$

PREDICT

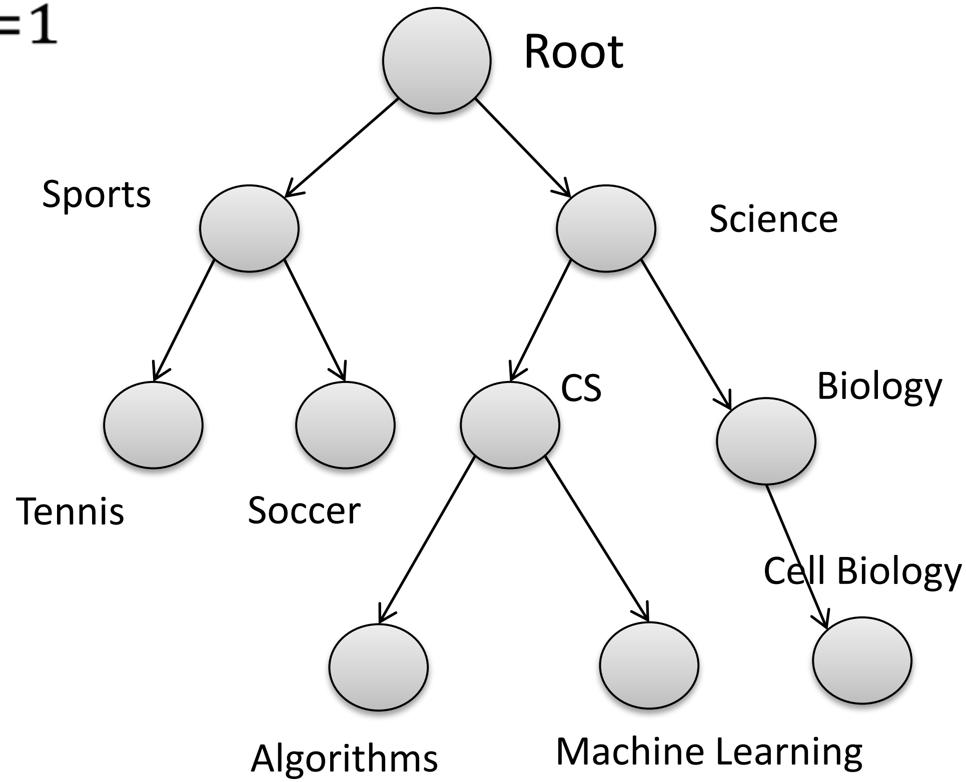
Predict on Unseen Instance $F(x), x \notin D$

Hierarchical Classification

Labeled Training Data

$$D = \{(x_i, t_i)\}_{i=1}^{i=N}$$

$$H \equiv \{\pi : \mathcal{N} \to \mathcal{N}\}$$



Also define,

$y_{in} = I(t_i = n)$
$C_n = \{y : \pi(y) = n\}$
$T \equiv$ {Leaf nodes}

Challenge 1- Scalability

- Typical Real-world Hierarchies have
  - Large number of training instances
  - High dimensionality
  - Large number of class labels

  => Large number of parameters

Dataset	# Examples	# Features	# Classes
LSHTC-2010	128,685	347,256	12,294
DMOZ-2012	486,843	575,556	11,947
DMOZ-2011	394,756	594,158	27,875
Yahoo! Directory	792,601	-	[illegible]
Wikipedia	2,817,603	1,617,899	325,056

1) 1.5GB of training examples
2) 2000GB of uncompressed parameters

Challenge 2 – Using the Hierarchy

- Hierarchy encodes similarity relationships between class-labels.
- How to encode them in the learning process ?
- How to share information between 2000GB of parameters ?

Related Work

- Decomposition methods (Koller and Sahami 1997, Dumais and Chen 2000, Liu et al 2003)
- Decompose and Share methods (Bennett and N DeCoro et al 2007, Xue et al 2008, ..)
- Smoothing methods (Mccallum et al 1998, Punera and 8)
- Global optimization methods
  - Bayesian methods (Shahbaba and Neal 2007, Gopa
  - Discriminative methods (Cai and Hoffman 2004, Tsoc aridis et al 2006, Zhou et al 2011, ..)

Scalable, but limited use of hierarchy

Better use of hierarchy, but cannot scalable

Hierarchically Regularized Methods

Regularization Constant

- Formulate using the Risk M work.
  Parame vect f weights w_n with node n

Regularization Term

Empirical Risk on the Training set

- The general form Structural Risk Minimiz

$$\lambda(\boldsymbol{w}) + C \times R_{emp}(\boldsymbol{D}, \boldsymbol{w})$$

- Define Empirical Risk as Loss at leaf nodes

Loss Function

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

- Incorporate the Hierarchy into Regularization Term,

$$\lambda(\boldsymbol{w}) = \sum_{n \in \mathcal{N}} \frac{1}{2} ||w_{\pi(n)} - w_n||^2$$

Formulation

- Optimization Objective

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

- Advantages over other approaches,
  - Hierarchy is not used inside the Empirical Risk term.
  - Can be split into multiple optimization problems.
  - Easily parallelizable.
  - Flexibility in choosing loss function
    - Hinge loss HR-SVM: $L(y_{in}, x_i, w_n) = \ max(0,1 - y_{in} {w_n}^T x_i)$
    - Logistic loss HR-LR: $L(y_{in}, x_i, w_n) = \ \log(1 + \exp(-y_{in} {w_n}^T x_i))$

Optimization

- A Sequential Optimization method,
  For each node n
    - Optimize OBJ_L w.r.t parameter w_n

Converges because function is Convex

- If n is a non-leaf node i.e. $n \notin T$

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

- If n is a leaf-node i.e. $n \,\epsilon\, T$

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

Optimization

- HR-SVM

$$\frac{1}{2}||w_{\pi(n)} - w_n||^2 + C\sum_{i=1}^{i=N} max(0,1 - y_{in}{w_n}^T x_i)$$

  - Non-differentiable objective; form a differentiable dual problem.
  - Solve dual using co-ordinate descent

- HR-LR

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

  - Differentiable objective. Gradient is given by,

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

  - Use Limited Memory BFGS for optimization

Parallelization

- Large-scale hierarchies have
  - High memory requirements (remember 2000GB parameters ..)
  - High computational requirements (optimization for 325,000 classes…)
- Solution: **Parallel Optimization**

Parameters depend on each other

~~**In parallel** for each node n~~

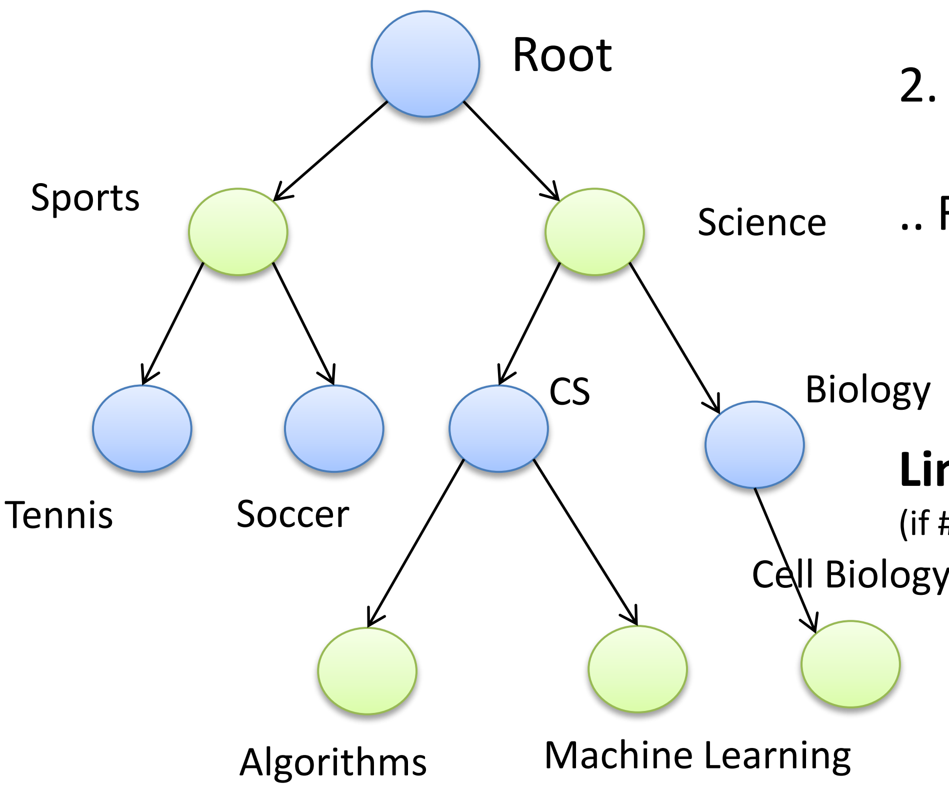
- ~~Optimize OBJ_L w.r.t parameter w_n~~

- Key Idea:
  - Interactions between the w_n's are only through the parent and child nodes.

$$OBJ_L = \sum_{n \in \mathcal{N}} \frac{1}{2} ||w_{\pi(n)} - w_n||^2 + ..$$

  - By Fixing the parent and the children of a node, it can optimized independently from the rest of the hierarchy.

Parallelization (2)



1. Fix Blue nodes, optimize the green nodes in parallel
2. Fix the green nodes, optimize blue nodes in parallel

.. Repeat

Linear speed up in processors

(if #class-labels > #processors)

$$OBJ_L = \sum_{n \in \mathcal{N}} \frac{1}{2} ||w_{\pi(n)} - w_n||^2 \ + ..$$

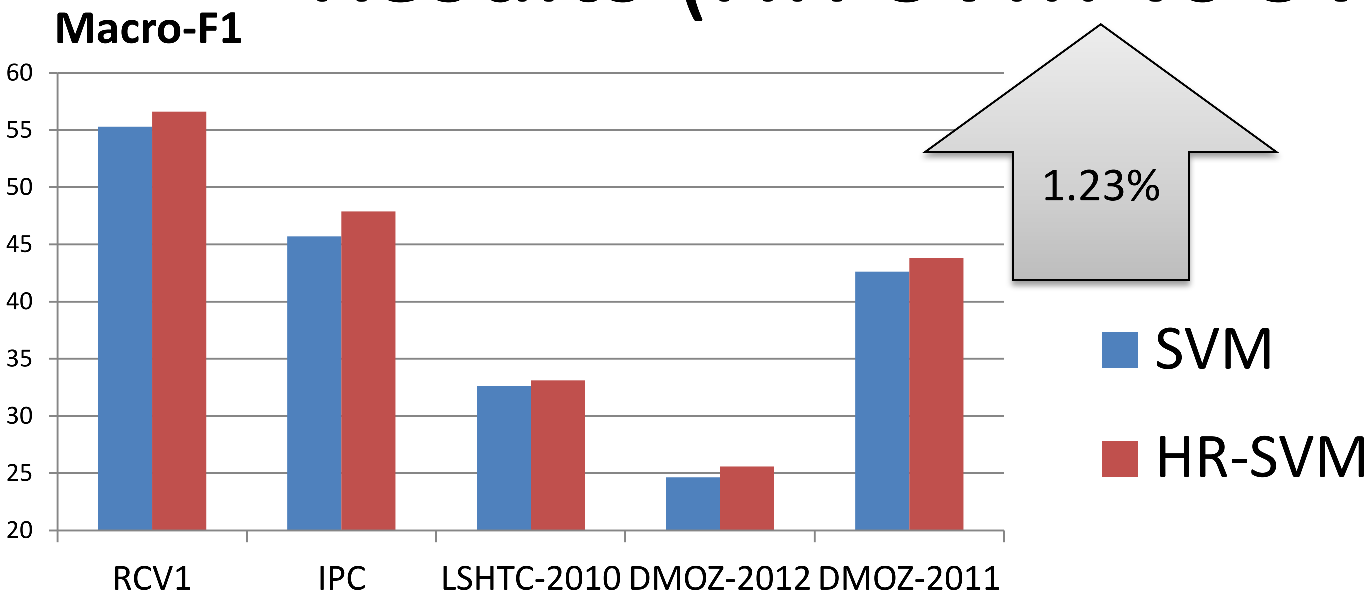
Setup

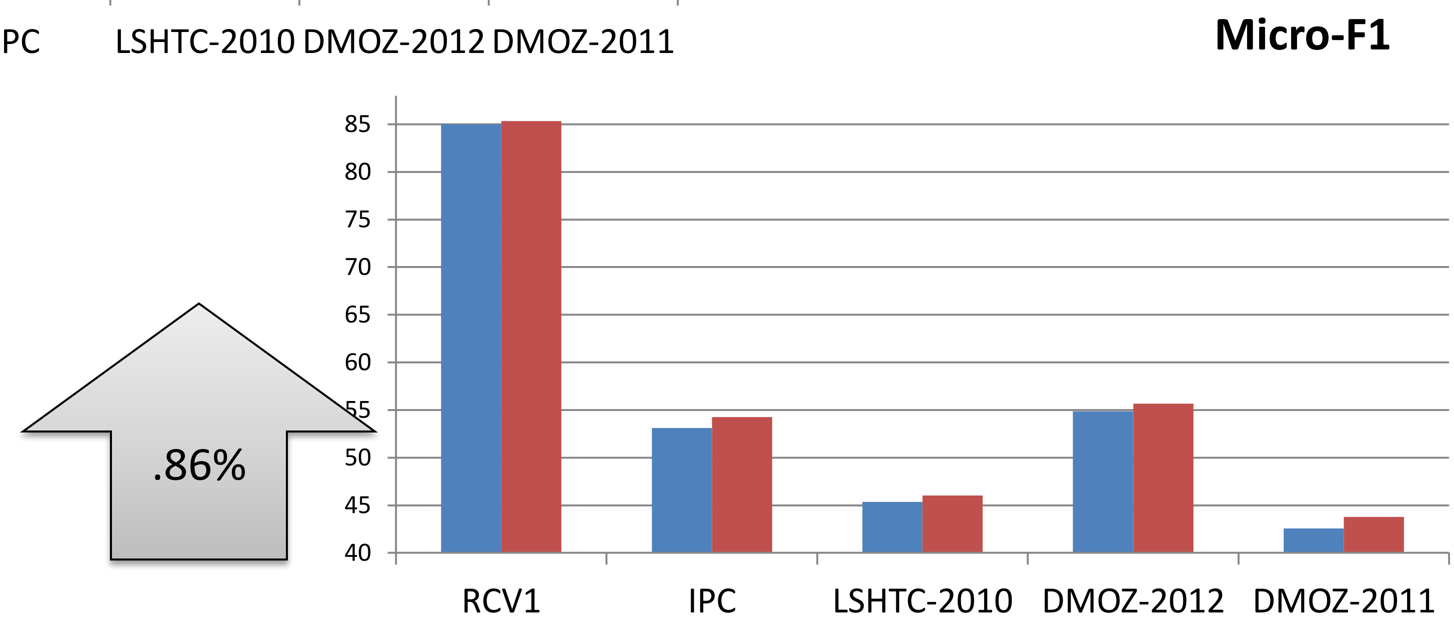
- Datasets

Dataset	*#Training*	*#Testing*	*#Class-Labels*	*#Leaf-labels*	*Depth*	*#Features*
RCV1	13732	468355	97	75	5	48734
IPC	46324	28926	552	451	4	541869
LSHTC-2010	128710	34880	15358	12294	6	381580
DMOZ-2012	177580	177581	13137	11737	6	348548
DMOZ-2011	384161	104263	33591	27840	6	594158

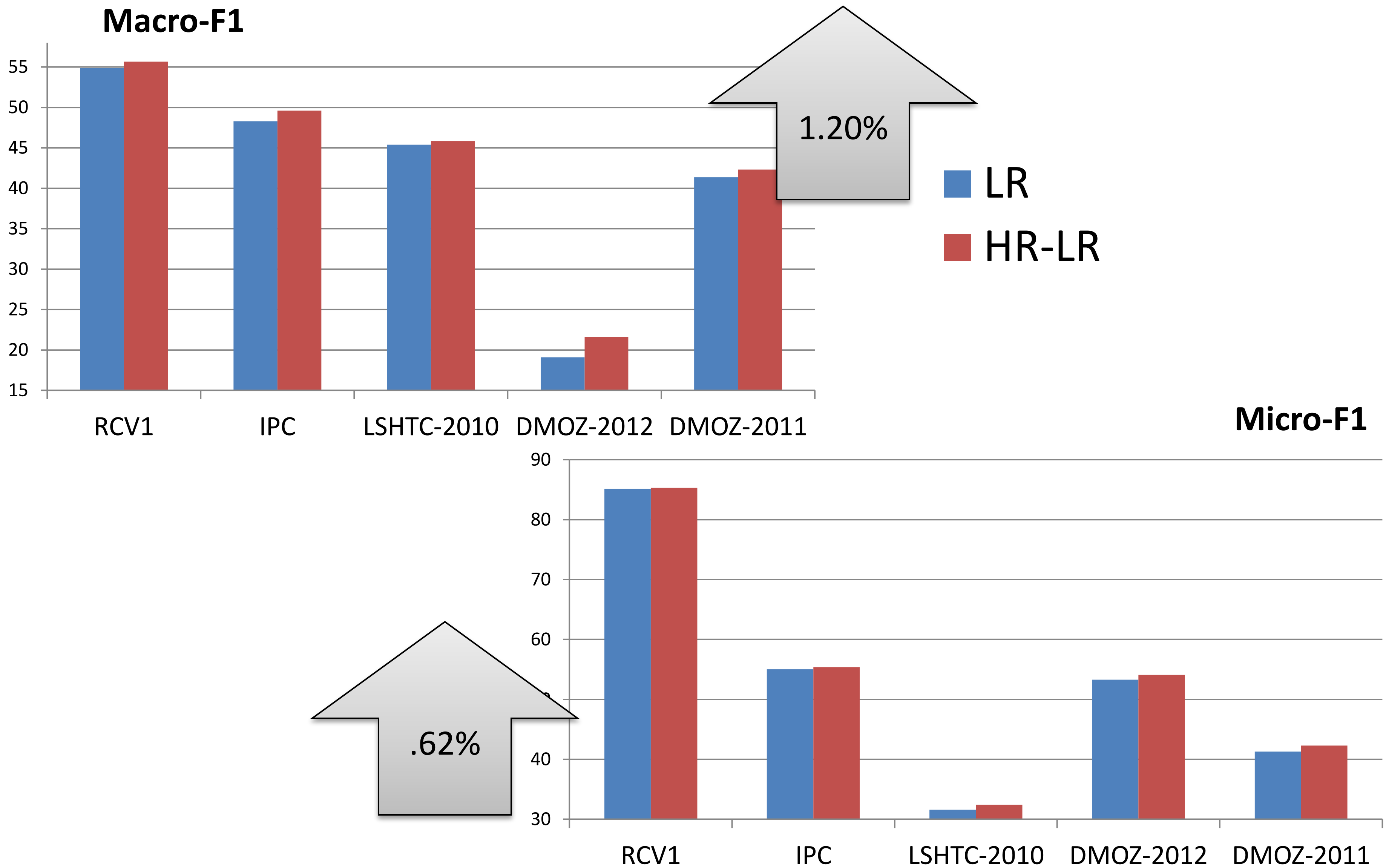
- Cluster configuration
  - map-reduce based Hadoop 20.2
  - 64 worker nodes having 8 cores and 16GB RAM
  - 300 cores were used as Mappers and 220 cores were used as Reducers
  - Accumulo 1.4 key-value store for fast lookup of weight vectors

Results (HR-SVM vs SVM)





HR-LR vs LR


Macro-F1
55
50
45
40
35
30
25
20
15
RCV1
IPC
LSHTC-2010
DMOZ-2012
DMOZ-2011
1.20%
LR
HR-LR
Micro-F1
90
80
70
60
40
30
.62%
RCV1
IPC
LSHTC-2010
DMOZ-2012
DMOZ-2011

Results (cont..)

- Training Time (in minutes)

Dataset	SVM	HR-SVM
RCV1	2.73	1[illegible]
IPC	3.12	1[illegible]
LSHTC-2010	5.12	2[illegible]
DMOZ-2012	21.34	126.83
DMOZ-2011	38.47	221.23

4x
slower

Dataset	LR	HR-LR
RCV1	2.89	11.7[illegible]
IPC	4.17	15.9[illegible]
LSHTC-2010	97.24	123.22
DMOZ-2012	98.38	223.6
DMOZ-2011	102.9	242.45

2x
slower

- HR-{SVM,LR} are 4x,2x slower than SVM, LR respectively.
- LR methods are generally slower than SVM methods.

Software !

- Coming soon – ‘HiClass’
  A comprehensive Large-scale parallelizable Hierarchical classification package.
  - Readily deployable on Hadoop Clusters
  - Support for Amazon EC2 and S3 data storage
  - Support for Accumulo key-value storage
- A preliminary version can be downloaded from

  http://gcdart.blogspot.com/

Some Results with Hiclass

Track 1: Medium-size Wikipedia

Results Ordered by Accuracy:

Name	Acc	EBF	EBP	EBR	LBMaF	LBMaP	LBMaR	LBMiF	LBMiP
arthur	0.438162	0.493725	0.551626	0.496298	0.267413	0.573306	0.287564	0.493908	0.565777
coolvegpuff	0.429097	0.482416	0.55004	0.476256	0.250652	0.526077	0.256693	0.47792	0.552359
TTI	0.41998	0.477056	0.504828	0.508361	0.283516	0.506366	0.306949	0.47252	0.476908
chrishan	0.411662	0.476801	0.517573	0.511599	0.245418	0.42618	0.299562	0.418699	0.393709

42 mins

Track 1: Large-size Wikipedia

Results Ordered by Accuracy:

Name	Acc	EBF	EBP	EBR	LBMaF	LBMaP	LBMaR	LBMiF	LBMiP	LBMiR
coolvegpuff	0.380602	0.459109	0.571037	0.462536	0.217395	0.324894	0.243229	0.321584	0.293466	0.2[illegible]56
chrishan	0.345788	0.419836	0.606974	0.366956	0.170133	0.537858	0.175952	0.344795	0.551397	0.2508
dhlee	0.340175	0.414932	0.478364	0.406007	0.256355	0.408649	0.30777	0.34523	0.415535	0.2952
daq	0.332619	0.400666	0.493979	0.400826	0.173825	0.32088	0.203059	0.309634	0.32858	0.2927

~ 23 hours

Track 2: DMOZ

Results Ordered by Accuracy:

Name	Acc	F	P	R	HF	HP	HR
coolvegpuff	0.564441	0.355264	0.346797	0.338724	0.753465	0.754661	0.753945
chrishan	0.519756	0.313586	0.30928	0.30509	0.733515	0.734806	0.733953
fancycheung	0.346798	0.147958	0.142383	0.137212	0.60598	0.609595	
Top-Down NB - baseline	0.334519	0.0834439	0.0927612	0.104421	0.614807	0.614362	

Baseline 0.249137 0.317596 0.282953

24 mins

NOTE: Different threshold strategy was used though !

Acknowledgements !

- Thank you –
  - Eric Gaussier.

    For arranging to support my travel to the workshop.
  - Elsa Hollard, Pulfer Corinne

    For flight bookings, and all administrative stuff.
  - LSHTC Workshop

    For giving a chance to present our work.
  - Audience

    For reading until this very last line ☺