Classification in a large number of categories

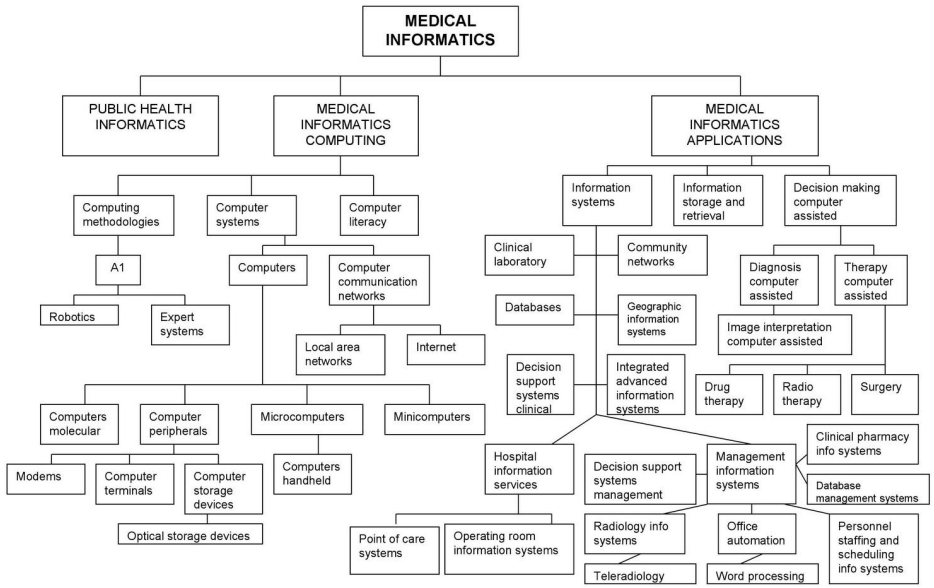
T. Artières[◇],
Joint work with the MLIA team[◇] at LIP6, the AMA team at LIG[†] and Demokritos[‡]

[◇] Lab. d'informatique de Paris 6, France
[†] Lab. d'Informatique de Grenoble & Grenoble University, France
[‡] N.C.S.R., Demokritos, Athens, Greece

September 4, 2017

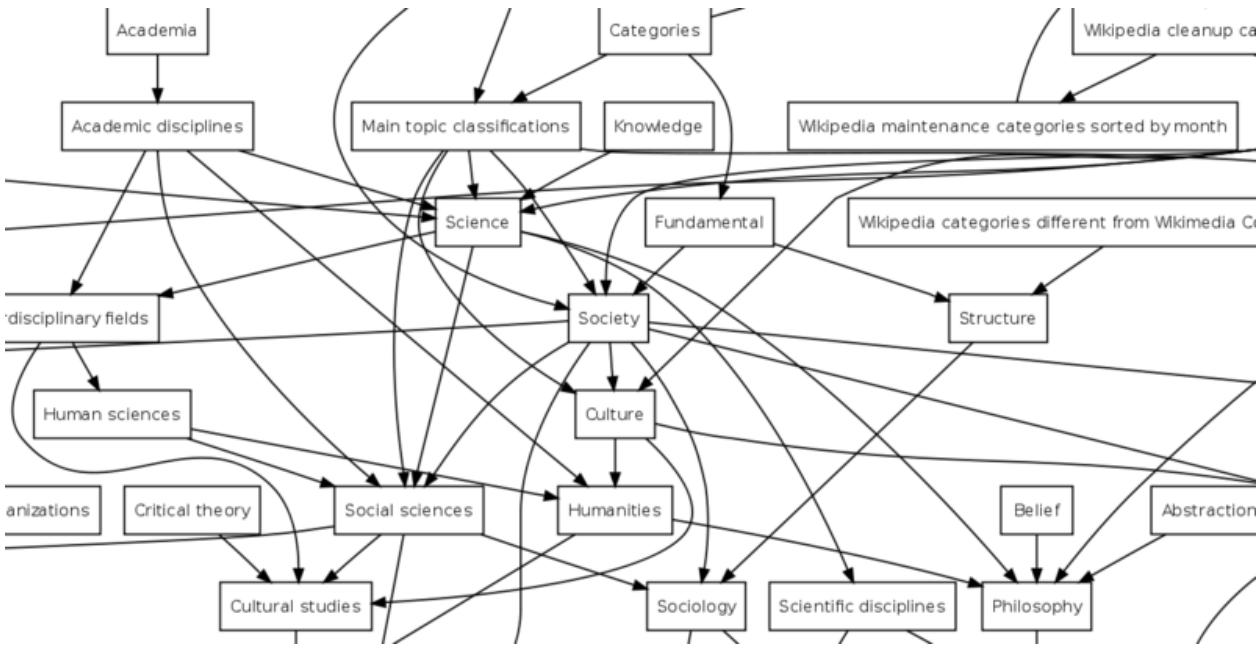
Classification in a Large Number of Categories (CLNC)

- Existing huge hierarchies
  - Yahoo! Directory (130k categories)
  - Wikipedia (325k categories)
  - Mesh (Medical), International patents (IPC), Open Directory Project (ODP), Reuters Hierarchy for news-stories, ...



Classification in a Large Number of Categories (CLNC)

- Existing huge hierarchies
  - Yahoo! Directory (130k categories)
  - Wikipedia (325k categories)
  - Mesh (Medical), International patents (IPC), Open Directory Project (ODP), Reuters Hierarchy for news-stories, ...



Classification in a Large Number of Categories (CLNC)

- Existing huge hierarchies
  - Yahoo! Directory (130k categories)
  - Wikipedia (325k categories)
  - Mesh (Medical), International patents (IPC), Open Directory Project (ODP), Reuters Hierarchy for news-stories, ...
- How many is a large Number of Classes ?
  - Qualitative change of the methods
  - May vary according to the data (e.g. Images vs. Text)
  - 38% accuracy on 325k classes (text docs) (Cf. LSHTC3) vs. 17% on 10k classes (Imagnet)

Classification in a Large Number of Categories (CLNC)

- Existing huge hierarchies
  - Yahoo! Directory (130k categories)
  - Wikipedia (325k categories)
  - Mesh (Medical), International patents (IPC), Open Directory Project (ODP), Reuters Hierarchy for news-stories, ...
- How many is a large Number of Classes ?
  - Qualitative change of the methods
  - May vary according to the data (e.g. Images vs. Text)
  - 38% accuracy on 325k classes (text docs) (Cf. LSHTC3) vs. 17% on 10k classes (Imagnet)
- Applications / context
  - Search engines
  - Image Annotation
  - Text classification

Additional ressource about labels (Prior Label Relation)

Prior Label Relation structure

- Tree hierarchy of generalization / specification (usually disjoint)
- Some times it is a DAG, less often it is a graph

Trees

- More intuitive (generalization/specification relation)
- Low complexity algorithms (training and inference)

In practice

- Most often lots of cleaning before use
- Selection of a part of a hierarchy, removal of unpredictable nodes

Standard settings of CLNC

Main challenge

Maintaining accuracy while being scalable (training and inference)

Side effects of CLNC problems

- Large number of samples
- Large number of features

Standard settings of CLNC

Main challenge

Maintaining accuracy while being scalable (training and inference)

Side effects of CLNC problems

- Large number of samples
- Large number of features
- Unbalanced classification problems and almost not learnable classifiers

Standard settings of CLNC

Main challenge

Maintaining accuracy while being scalable (training and inference)

Side effects of CLNC problems

- Large number of samples
- Large number of features
- Unbalanced classification problems and almost not learnable classifiers
- Multilabel data

Standard settings of CLNC

Main challenge

Maintaining accuracy while being scalable (training and inference)

Side effects of CLNC problems

- Large number of samples
- Large number of features
- Unbalanced classification problems and almost not learnable classifiers
- Multilabel data
- Optimize wrt an appropriate evaluation criterion

Standard settings of CLNC

Main challenge

Maintaining accuracy while being scalable (training and inference)

Side effects of CLNC problems

- Large number of samples
- Large number of features
- Unbalanced classification problems and almost not learnable classifiers
- Multilabel data
- Optimize wrt an appropriate evaluation criterion
- Relevance of an existing hierarchy / ontology (if any)
- Time varying hierarchy

LNC datasets

Name	#cat	#features	#docs	cat/doc	Tree hier	max path
news20	20	19,996	139,217	1	✓	2
rcv1	101	47,236	806,791	3,1	✓	3
Yahoo! Directory (2004)	132,199	4,194,304	792,601	2.2	✓	16
Ohshumed	14,321	72,076	233,445	12	✓	10
DMOZ (LSHTC1)	27,875	497,992	594,158	1.02	✓	5
Wiki Small (LSHTC2)	36,504	346,299	538,148	1.86	-	10
Wiki Large (LSHTC3)	325,056	1,617,899	2,817,603	3.26	-	14
BioAsq	26,563	-	10,876,004	12.55	✓	5.24

LNC datasets main features

Size of the categories

- Power law distribution
  - 76% of the Yahoo! 246k categories have less than 5 docs

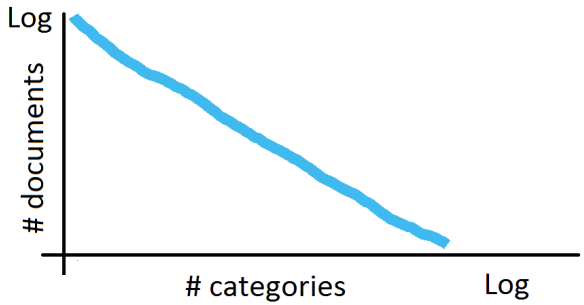
Rare categories

- Proportion of rare categories increases at deeper hierarchy levels
- It is a often good idea to remove rare classes ((DS10))

LNC datasets main features

Size of the categories

- Power law distribution
  - 76% of the Yahoo! 246k categories have less than 5 docs



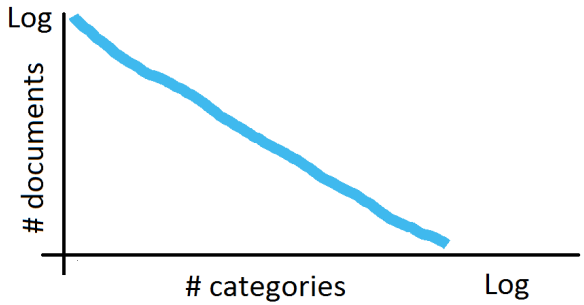
Rare categories

- Proportion of rare categories increases at deeper hierarchy levels
- It is a often good idea to remove rare classes ((DS10))

LNC datasets main features

Size of the categories

- Power law distribution
  - 76% of the Yahoo! 246k categories have less than 5 docs



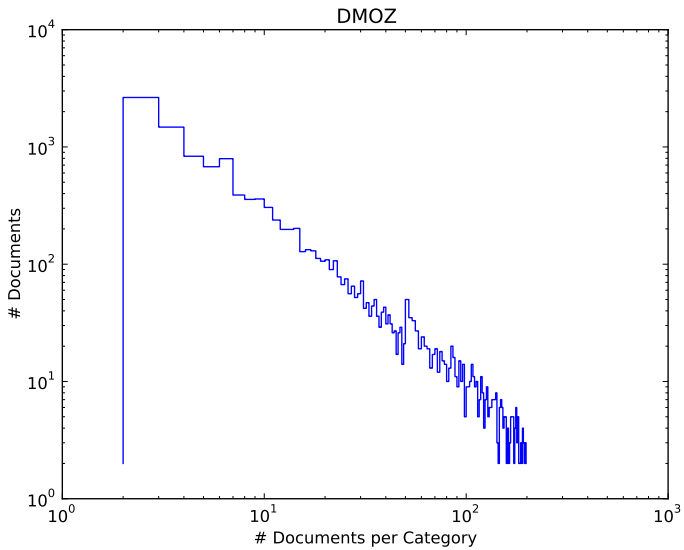
Rare categories

- Proportion of rare categories increases at deeper hierarchy levels
- It is a often good idea to remove rare classes ((DS10))

LNC datasets main features

Size of the categories

- Power law distribution
  - 76% of the Yahoo! 246k categories have less than 5 docs



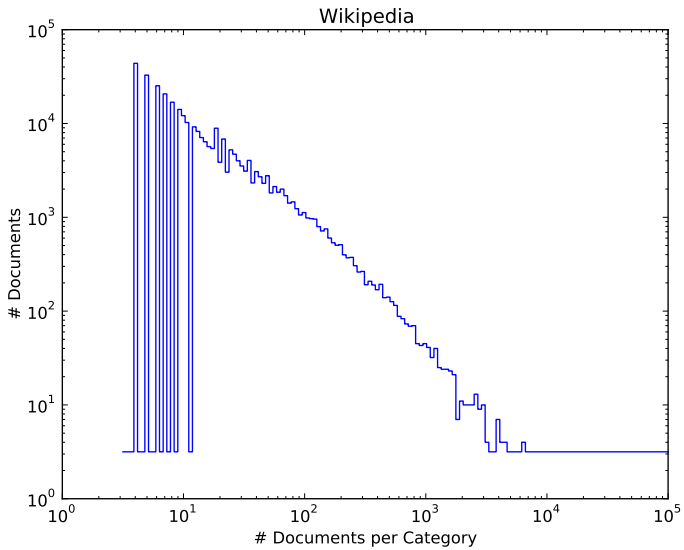
Rare categories

- Proportion of rare categories increases at deeper hierarchy levels
- It is a often good idea to remove rare classes ((DS10))

LNC datasets main features

Size of the categories

- Power law distribution
  - 76% of the Yahoo! 246k categories have less than 5 docs



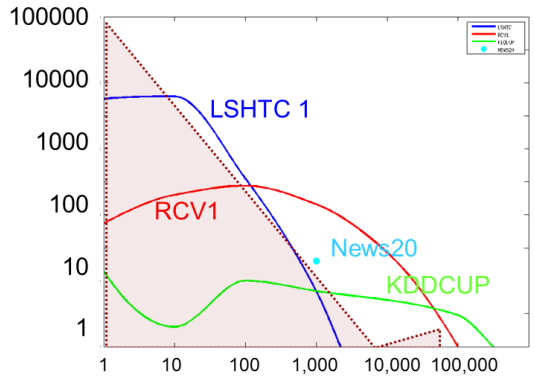
Rare categories

- Proportion of rare categories increases at deeper hierarchy levels
- It is a often good idea to remove rare classes ((DS10))

LNC datasets main features

Size of the categories

- Power law distribution
  - 76% of the Yahoo! 246k categories have less than 5 docs



[Y. Yang, Tutorial ECIR 2010]

Rare categories

- Proportion of rare categories increases at deeper hierarchy levels
- It is a often good idea to remove rare classes ((DS10))

LNC datasets main features

Distribution vs. depth

- Number of Category vs hierarchy depth (level)
- Number of documents vs hierarchy depth (level)

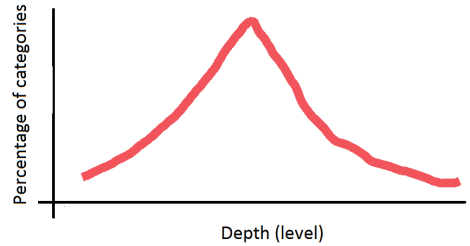
Multilabel

- Usually multilabel but not so much
  - 2.23 label/doc in Yahoo! Dataset
  - 3.26 label/doc in Wikipedia Large
  - 1.11 label/doc for subset of 1k largest Wikipedia categories

LNC datasets main features

Distribution vs. depth

- Number of Category vs hierarchy depth (level)
- Number of documents vs hierarchy depth (level)



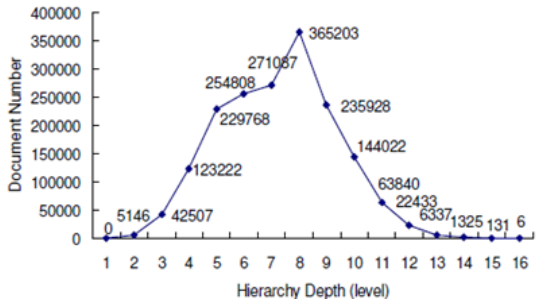
Multilabel

- Usually multilabel but not so much
  - 2.23 label/doc in Yahoo! Dataset
  - 3.26 label/doc in Wikipedia Large
  - 1.11 label/doc for subset of 1k largest Wikipedia categories

LNC datasets main features

Distribution vs. depth

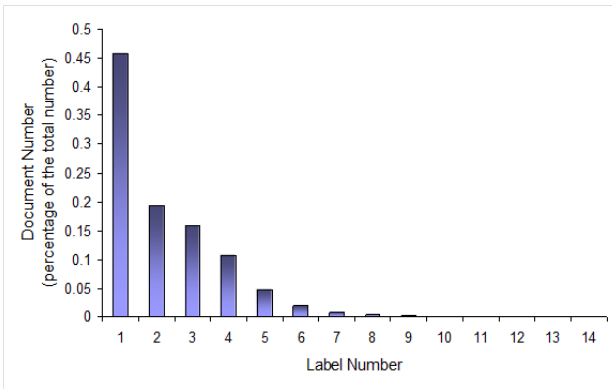
- Number of Category vs hierarchy depth (level)
- Number of documents vs hierarchy depth (level)



(LYW+05)

Multilabel

- Usually multilabel but not so much
  - 2.23 label/doc in Yahoo! Dataset
  - 3.26 label/doc in Wikipedia Large
  - 1.11 label/doc for subset of 1k largest Wikipedia categories

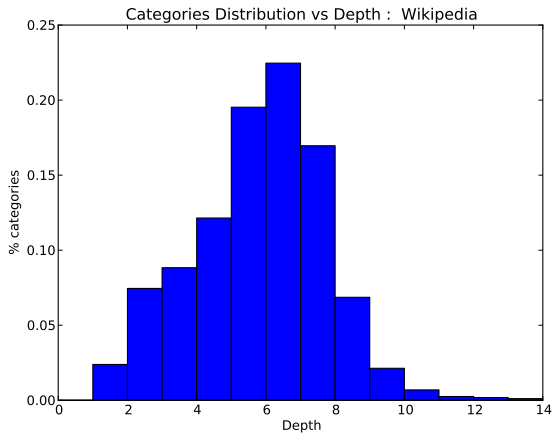


[Y. Yang, Tutorial ECIR 2010]

LNC datasets main features

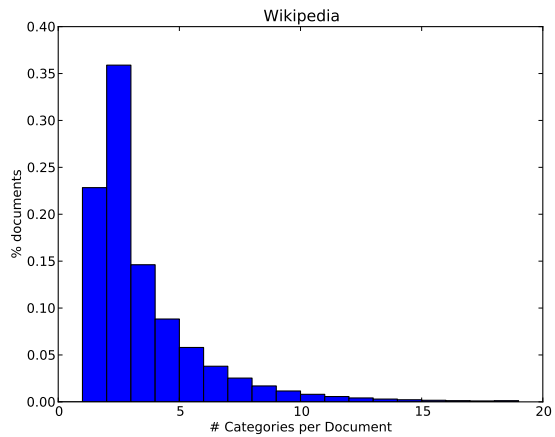
Distribution vs. depth

- Number of Category vs hierarchy depth (level)
- Number of documents vs hierarchy depth (level)

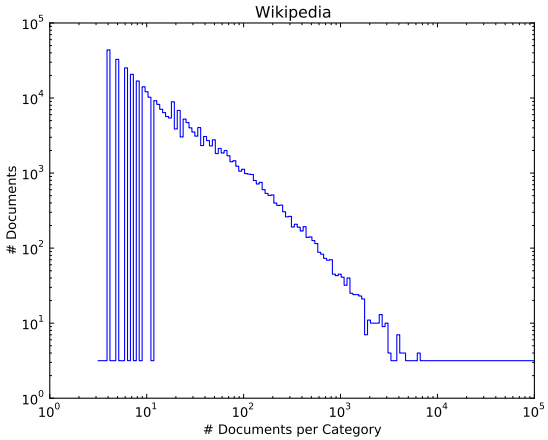


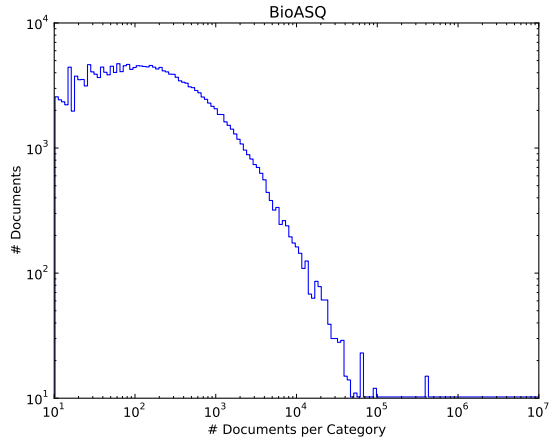
Multilabel

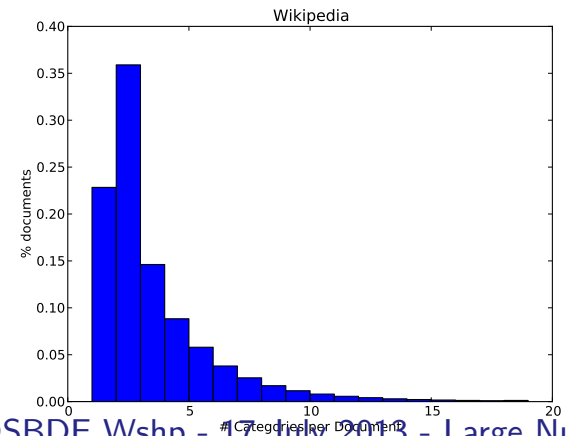
- Usually multilabel but not so much
  - 2.23 label/doc in Yahoo! Dataset
  - 3.26 label/doc in Wikipedia Large
  - 1.11 label/doc for subset of 1k largest Wikipedia categories

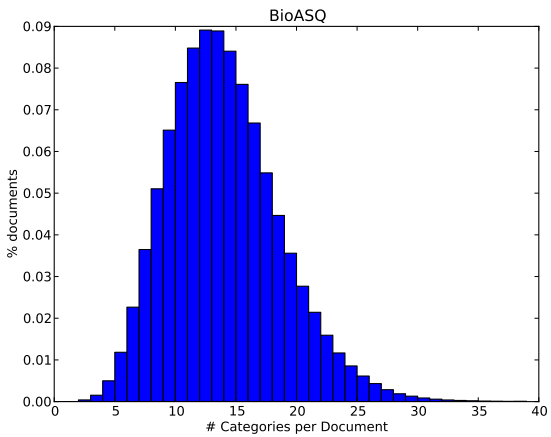


Organization hierarchies vs. tag hierarchies









LSHTC Challenges

Large Scale Hierarchical Text Classification
(`http://lshtc.iit.demokritos.gr/`)

LSHTC1 - 2010 - Whsp ECIR

- Tree hierarchy/ Monolabel / Number of categories up to 12,000

LSHTC2 - 2011 - Wshp ECML

- Tree or DAG hierarchy / Multi-label / Number of categories up to 325,000

LSHTC3 - 2012 - Wshp ECML

- Tree or DAG hierarchy / Multi-label / Number of categories up to 325,000
- Additional tracks: Multi-task/Transfer Learning and Refinement learning (semi-supervised and unsupervised)

BioASQ Challenge 2013, 2014

- Challenge on biomedical semantic indexing and Question-Answering
- Motivating example: *Q1: What is the role of thyroid hormones administration in the treatment of heart failure?*

Objectives

1. Large-scale classification of biomedical documents onto ontology concepts, in order to automate semantic indexing

Task 1a

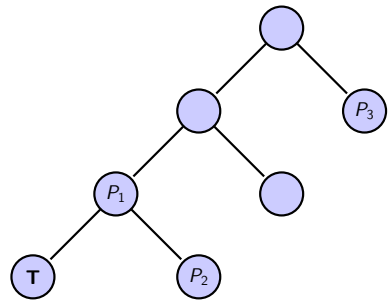
- BioASQ distributes new unclassified PubMeed abstracts
- BioMedAnswers attaches MeSH terms (limited resp. time)
- Evaluation when abstracts get classified by PubMed curators

2. Delivery of all retrieved information in a concise and

Evaluation measures

Flat vs. hierarchical

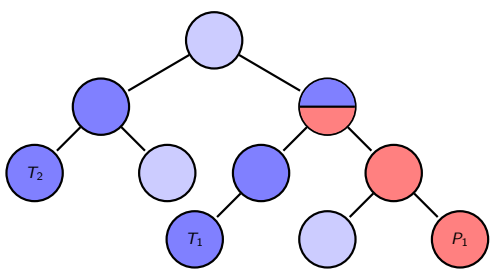
- T = target, P_1, P_2, P_3 are prediction
- Flat measure counts uniformly 1 error
- Hierarchical measure: Tree induced loss



Multilabel measures (TKV10)

- Accuracy, F_1, ...

Hierarchical versions of P and R, F_1 (CLCF07)

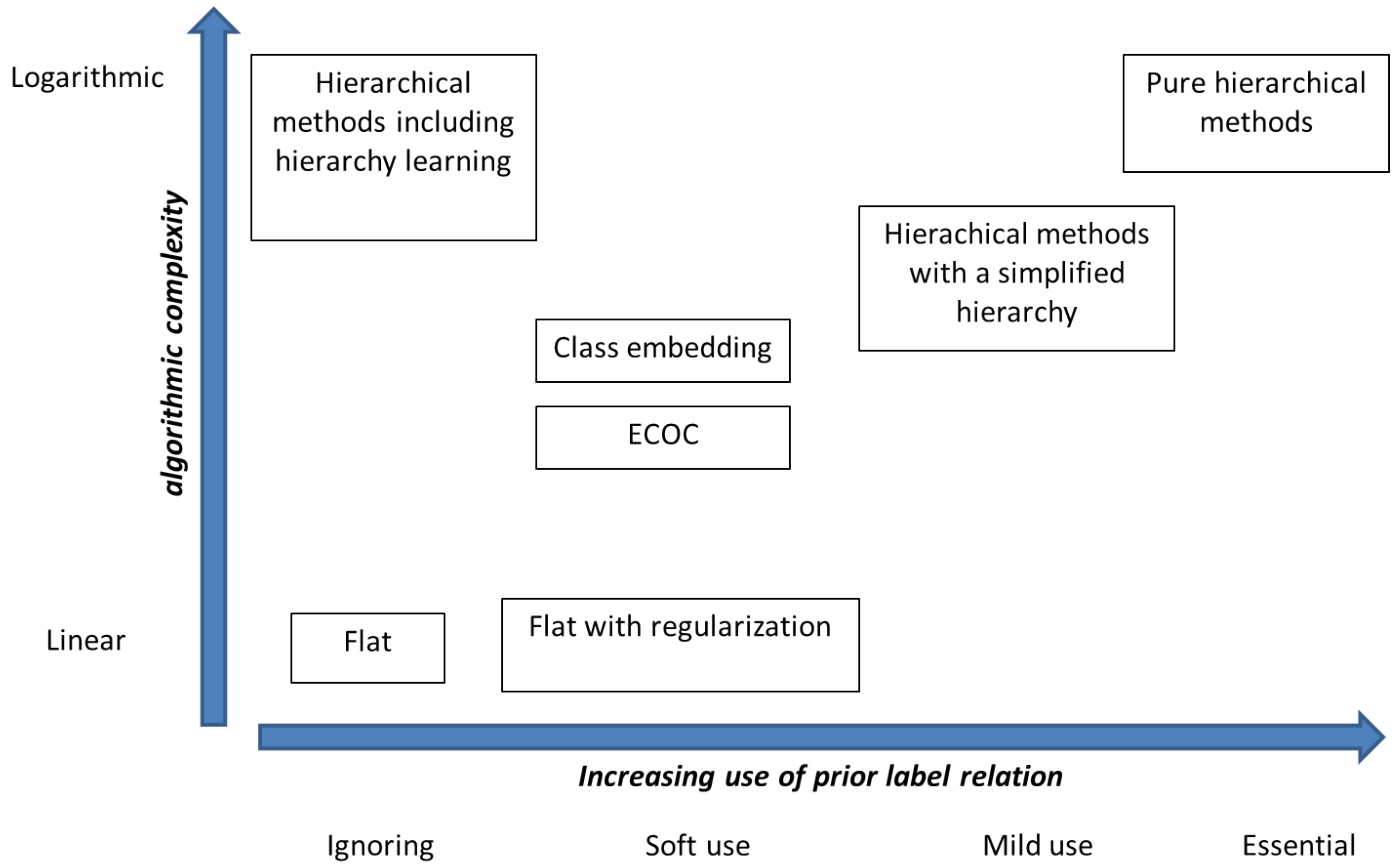


HP = 1/3, HR=1/5

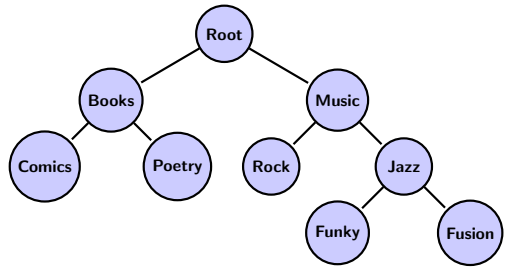
Main categories of methods

Many dichotomies

Flat vs. hierarchical / Exploiting hierarchy vs. ignoring / ...



Flat methods



Standard methods

KNN, One-vs-rest classifiers, (ECOC) based classifiers

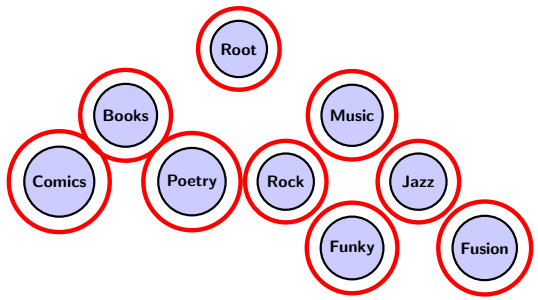
Advantages

- Conceptually simple, naturally multilabel
- Optimal wrt. accuracy / Hamming loss in ML case
- Do not require a (or rely on a possibly irrelevant) label information

Disadvantages

- Unbalanced classification problems
- Slow (training and inference)

Flat methods



Standard methods

KNN, One-vs-rest classifiers, (ECOC) based classifiers

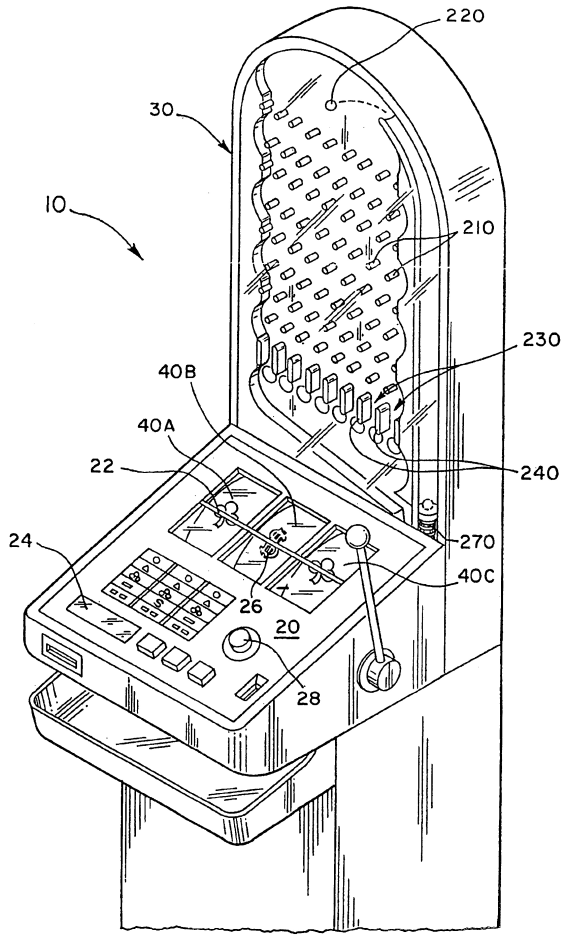
Advantages

- Conceptually simple, naturally multilabel
- Optimal wrt. accuracy / Hamming loss in ML case
- Do not require a (or rely on a possibly irrelevant) label information

Disadvantages

- Unbalanced classification problems
- Slow (training and inference)

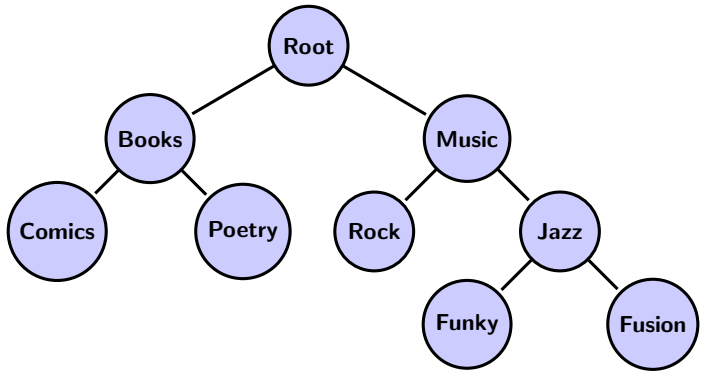
Hierarchical methods (Pachinko)



Hierarchical methods (Pachinko)

Inference *a la Pachinko*

1: $n = root$
2: **repeat**
3: $\quad n = Choose(Childs(n), x)$
4: **until** n is final for x



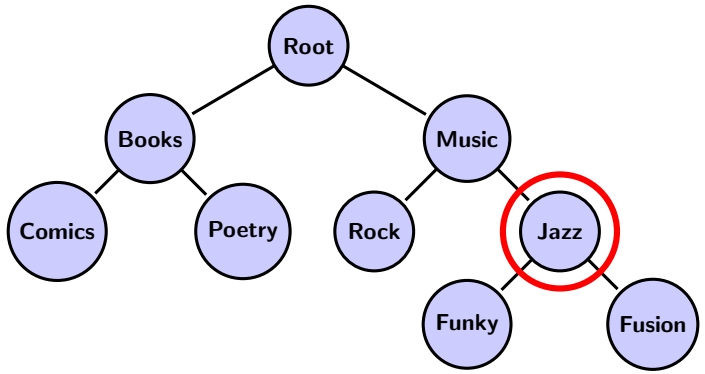
Big bang and Local Classifiers methods

- Big bang approaches: One global classifier
  - Margin based classifier for the tree induced loss (CH04) and the 0/1 tree loss (BWG10)
- Local classifiers methods: any kind of classifier may be used (SVM, ...)
  - One classifier per node / One classifier per father
  - With various ways to define positive and negative samples from the sub-hierarchy

Hierarchical methods (Pachinko)

Inference *a la Pachinko*

1: $n = root$
2: **repeat**
3: $\quad n = Choose(Childs(n), x)$
4: **until** n is final for x



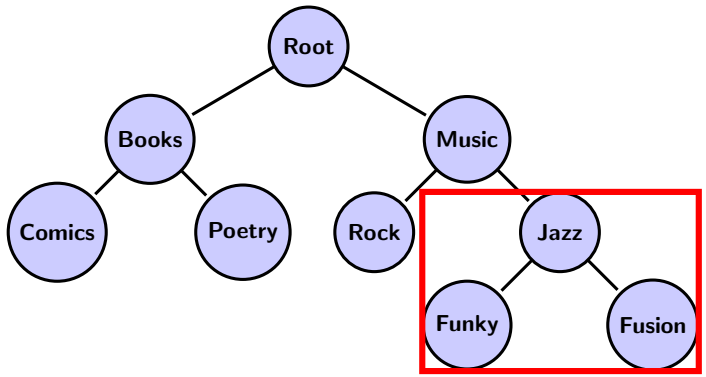
Big bang and Local Classifiers methods

- Big bang approaches: One global classifier
  - Margin based classifier for the tree induced loss (CH04) and the 0/1 tree loss (BWG10)
- Local classifiers methods: any kind of classifier may be used (SVM, ...)
  - One classifier per node / One classifier per father
  - With various ways to define positive and negative samples from the sub-hierarchy

Hierarchical methods (Pachinko)

Inference *a la Pachinko*

1: $n = root$
2: **repeat**
3: $\quad n = Choose(Childs(n), x)$
4: **until** n is final for x



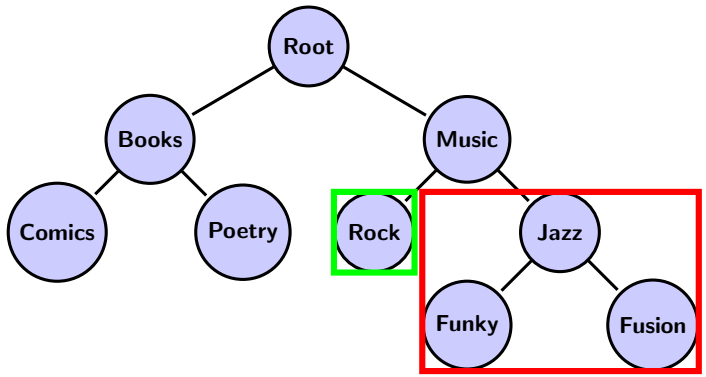
Big bang and Local Classifiers methods

- Big bang approaches: One global classifier
  - Margin based classifier for the tree induced loss (CH04) and the 0/1 tree loss (BWG10)
- Local classifiers methods: any kind of classifier may be used (SVM, ...)
  - One classifier per node / One classifier per father
  - With various ways to define positive and negative samples from the sub-hierarchy

Hierarchical methods (Pachinko)

Inference *a la Pachinko*

1: $n = root$
2: **repeat**
3: $\quad n = Choose(Childs(n), x)$
4: **until** n is final for x



Big bang and Local Classifiers methods

- Big bang approaches: One global classifier
  - Margin based classifier for the tree induced loss (CH04) and the 0/1 tree loss (BWG10)
- Local classifiers methods: any kind of classifier may be used (SVM, ...)
  - One classifier per node / One classifier per father
  - With various ways to define positive and negative samples from the sub-hierarchy

Hierarchical methods

Advantages

- More balanced classification problems
- Much lower Inference complexity

Disadvantages

- Error propagation (Local classifier methods)
- Non linear decision surfaces
- Data sparsity (leaf nodes)
- Hierarchy maybe irrelevant or wrong and needs to be simplified in practical cases (Cf. LSHTC3)

Main lessons from experiments

- Flat methods
  - Perform well up to few thousands of classes
  - Does not scale beyond tens of thousands
- Pure Hierarchical method
  - Much lower Inference AND training complexity

    2,1h Hierarchical vs 1,8 months (flat) for training - 0.0016s vs 0,69s for testing (LYW+05) (Yahoo! Dataset)
  - Ususally found more accurate than flat methods (SF11)
  - Data sparisty problem

    Performance strongly corelated with the number of positive examples per category (LYW+05)
  - Partially irrelevant existing hierarchies
- In practice
  - Mix of ideas (e.g. Hierarchical classifier + label embedding in (SBG10))
  - Best performers in LSHTC series from different families

Ongoing work at LIP6

Improving flat monolabel classifiers

- Build on ECOC ideas but with low dimensional codes ($p < L$)
- Exploit Prior Label Relations (e.g. hierarchy) to design codes

Improving flat multilabel classifiers

- Design scalable multilabel classifiers that scale with the number of labels
- Use a low dimensional binary reduction again

Joint learning of the hierarchy and the hierarchical classifier

- No useful hierarchy for non text data (e.g. images)
- Many recent works in this direction / Preliminary works on that part

ECOC

Multiclass classification with ECOCs

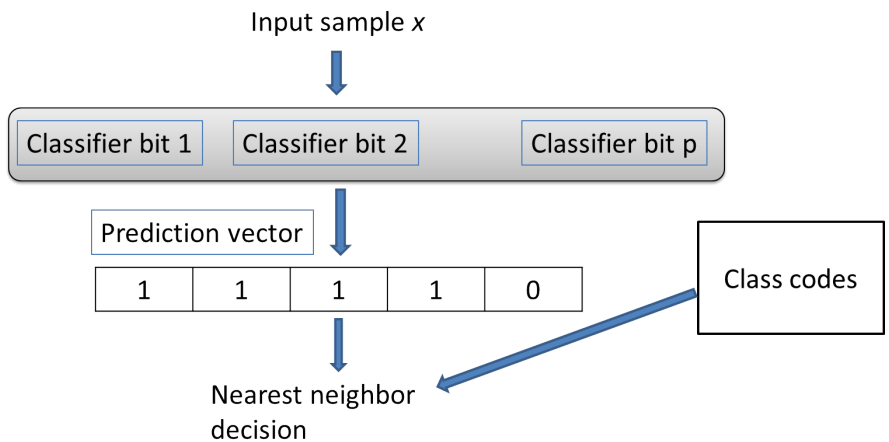
- Binary reduction of a L multiclass classification
  - Replace with $p > L$ binary classification problems
- Principle
  - Random generation of binary codes for every classes
  - Learn of a dicotomizer per bit
  - Inference through minimum Hamming distance between predicted binary vector and class codes

Class	bit 1	bit 2	bit 3	bit 4	bit 5
C_1	1	0	1	1	0
C_2	0	1	1	0	1
C_3	1	1	0	0	0

ECOC

Multiclass classification with ECOCs

- Binary reduction of a L multiclass classification
  - Replace with $p > L$ binary classification problems
- Principle
  - Random generation of binary codes for every classes
  - Learn of a dicotomizer per bit
  - Inference through minimum Hamming distance between predicted binary vector and class codes



Learning compact class codes using a prior on label relations (CAG12)

Using prior label relation to design codes

Learn binary codes that preserves the similarity between labels (class representations)

- Learn real valued codes
- Binarize the codes by thresholding
- Then training and inference as in ECOC classifiers

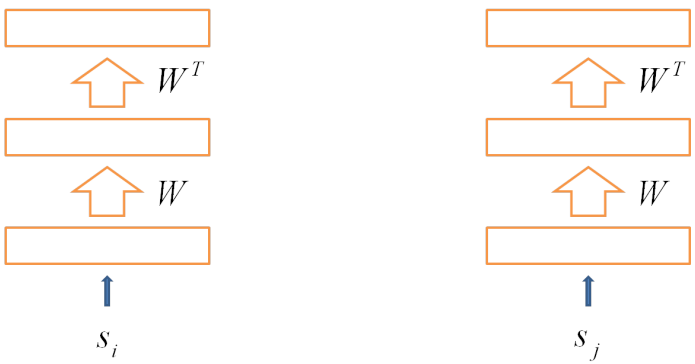
Learning continuous class codes

Start with a similarity matrix computed from the hierarchy

$$\begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,L} \\ \dots & \dots & \dots & \dots \\ s_{i,1} & s_{i,2} & \dots & s_{i,L} \\ \dots & \dots & \dots & \dots \\ s_{L,1} & s_{L,2} & \dots & s_{L,L} \end{bmatrix}$$

$$s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,L})$$

$$s_{i,j} = e^{-d_{Tree}(i,j)}$$



Algorithm

Repeat :

- pick randomly two vectors (s_i, s_j)
- make a gradient to optimize the corresponding loss

Until convergence

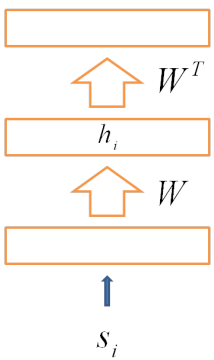
Learning continuous class codes

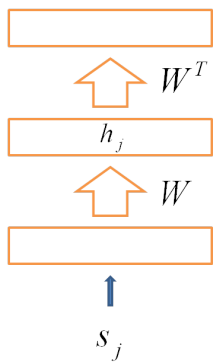
Start with a similarity matrix computed from the hierarchy

$$\begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,L} \\ \dots & \dots & \dots & \dots \\ s_{i,1} & s_{i,2} & \dots & s_{i,L} \\ \dots & \dots & \dots & \dots \\ s_{L,1} & s_{L,2} & \dots & s_{L,L} \end{bmatrix}$$

$$s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,L})$$

$$s_{i,j} = e^{-d_{Tree}(i,j)}$$





Algorithm

Repeat :

- pick randomly two vectors (s_i, s_j)
- make a gradient to optimize the corresponding loss

Until convergence

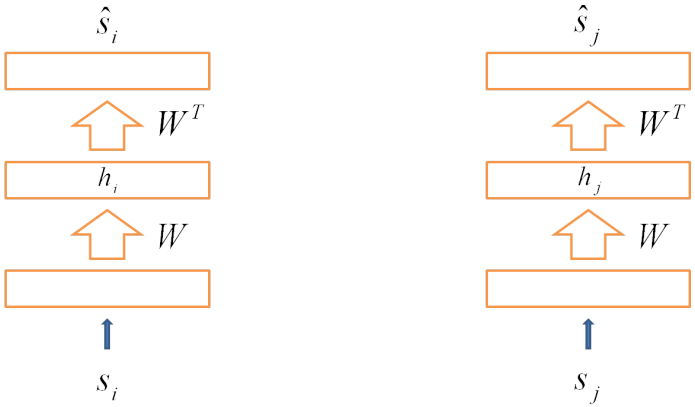
Learning continuous class codes

Start with a similarity matrix computed from the hierarchy

$$\begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,L} \\ \dots & \dots & \dots & \dots \\ s_{i,1} & s_{i,2} & \dots & s_{i,L} \\ \dots & \dots & \dots & \dots \\ s_{L,1} & s_{L,2} & \dots & s_{L,L} \end{bmatrix}$$

$$s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,L})$$

$$s_{i,j} = e^{-d_{Tree}(i,j)}$$



Algorithm

Repeat :

- pick randomly two vectors (s_i, s_j)
- make a gradient to optimize the corresponding loss

Until convergence

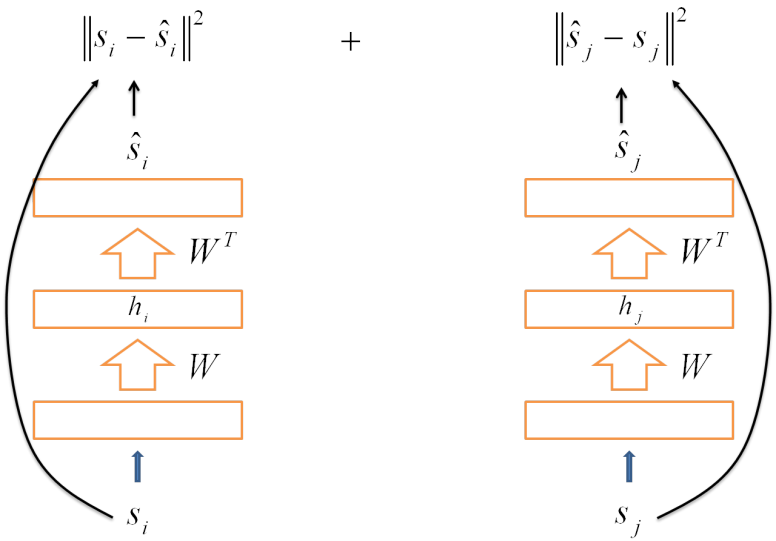
Learning continuous class codes

Start with a similarity matrix computed from the hierarchy

$$\begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,L} \\ \dots & \dots & \dots & \dots \\ s_{i,1} & s_{i,2} & \dots & s_{i,L} \\ \dots & \dots & \dots & \dots \\ s_{L,1} & s_{L,2} & \dots & s_{L,L} \end{bmatrix}$$

$$s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,L})$$

$$s_{i,j} = e^{-d_{Tree}(i,j)}$$



Algorithm

Repeat :

- pick randomly two vectors (s_i, s_j)
- make a gradient to optimize the corresponding loss

Until convergence

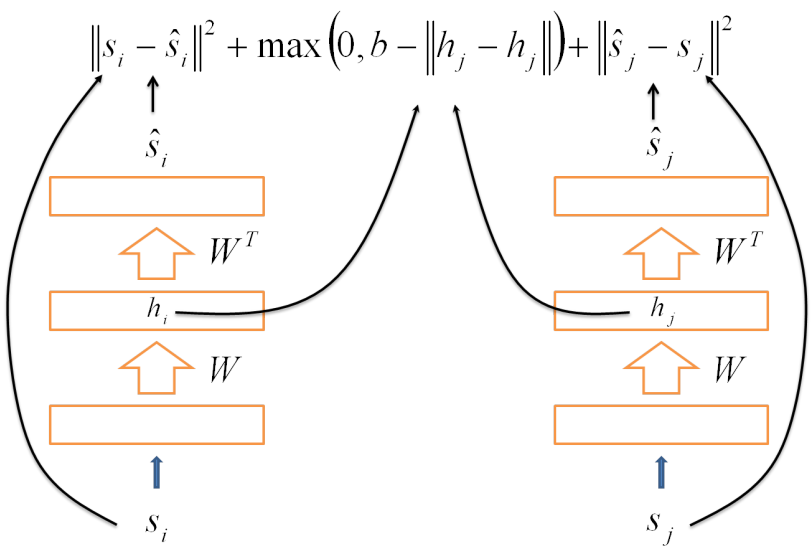
Learning continuous class codes

Start with a similarity matrix computed from the hierarchy

$$\begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,L} \\ \dots & \dots & \dots & \dots \\ s_{i,1} & s_{i,2} & \dots & s_{i,L} \\ \dots & \dots & \dots & \dots \\ s_{L,1} & s_{L,2} & \dots & s_{L,L} \end{bmatrix}$$

$$s_i = (s_{i,1}, s_{i,2}, \dots, s_{i,L})$$

$$s_{i,j} = e^{-d_{Tree}(i,j)}$$



Algorithm

Repeat :

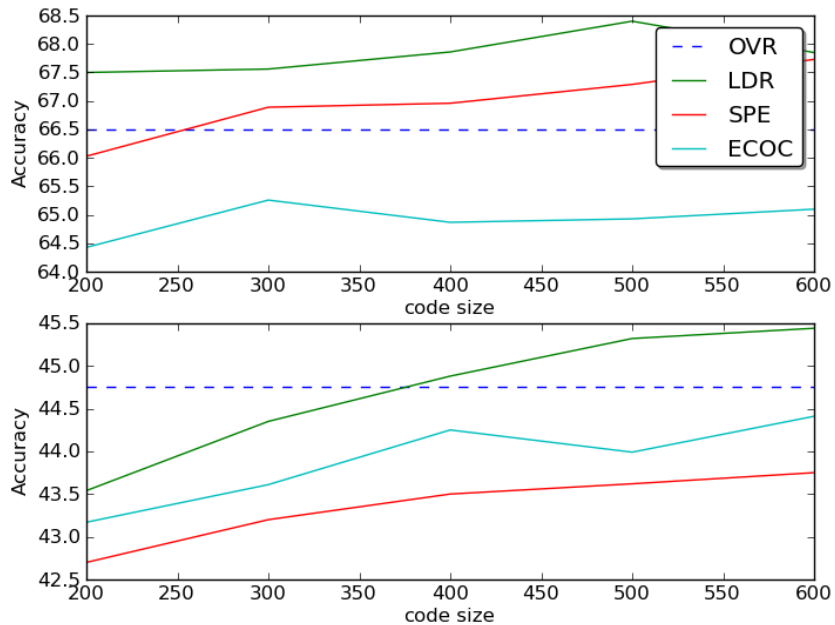
- pick randomly two vectors (s_i, s_j)
- make a gradient to optimize the corresponding loss

Until convergence

Designing binary class codes

1: **repeat**
2: Pick randomly two samples (s_i, s_j)
Make a gradient step to optimize $L_{\lambda,\alpha}(w)$
3: **until** convergence criteria is met
4: Compute the new representation of classes with the learned function $h_i = f(W \times s_i)$
5: **for all** j such that $1 \leq j \leq p$ **do**
6: Compute the median m_j of $(h_i(j))_{i=1..L}$
7: Redefine $\forall i = 1..L, u_i(j) = 1$ if $u_i(j) > m_j$, 0 otherwise
8: **end for**
9: **return** Class code matrix U

Results on LSHTC1 datasets



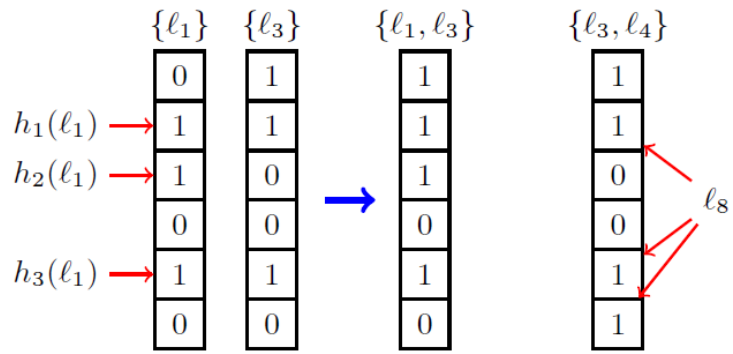
Zero shot learning is also possible

# labels removed	10	20	30	40	50
Accuracy (%)	25.64	24.45	16.76	14.31	12.76
std	12.20	6.34	4.24	3.18	2.48

Multilabel classification with Bloom filters

- Aim: Extend the previous idea to multilabel classification
- Main tool: Bloom Filters (BF)
  - Space efficient, random data structure for encoding small subsets of a large set of objects
  - The code of an element is defined by K hash functions
  - The code of a subset is the union of the codes of its elements
  - A code may be queried with a (small) false positive rate

label	h_1	h_2	h_3	label	h_1	h_2	h_3
ℓ_1	2	3	5	ℓ_5	1	4	6
ℓ_2	2	4	5	ℓ_6	3	5	6
ℓ_3	1	2	5	ℓ_7	3	4	5
ℓ_4	1	5	6	ℓ_8	2	5	6



Multilabel classification with RBF

Multilabel Classification with BF

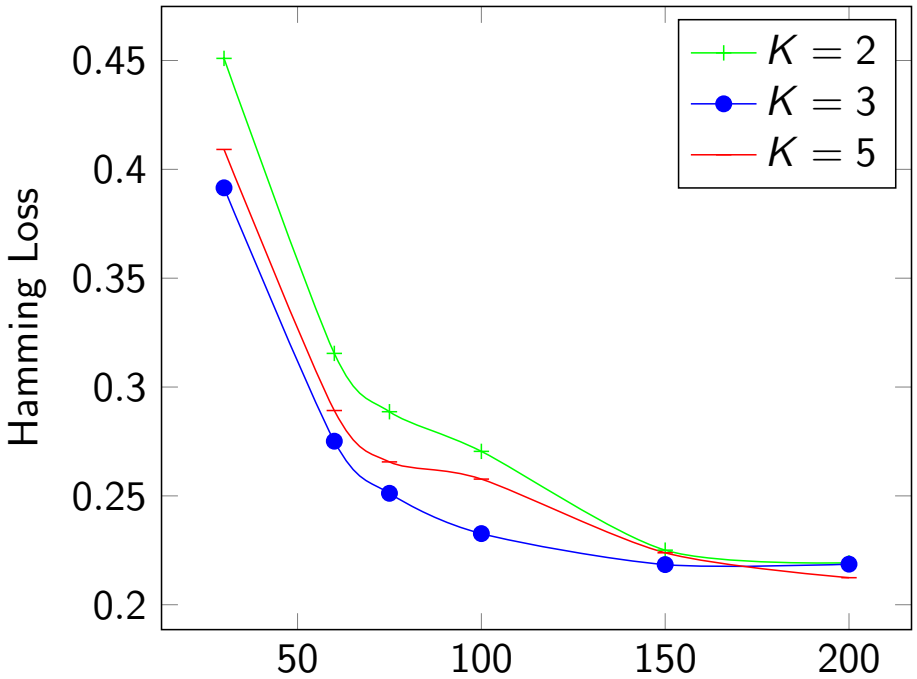
- Training
  - Define a BF (of length p) for encoding labels
  - Learn p binary classifiers
- Test
  - Compute the vectors of p predictions
  - Query every label
- May be fast provided p is small (p/L factor wrt. BR).

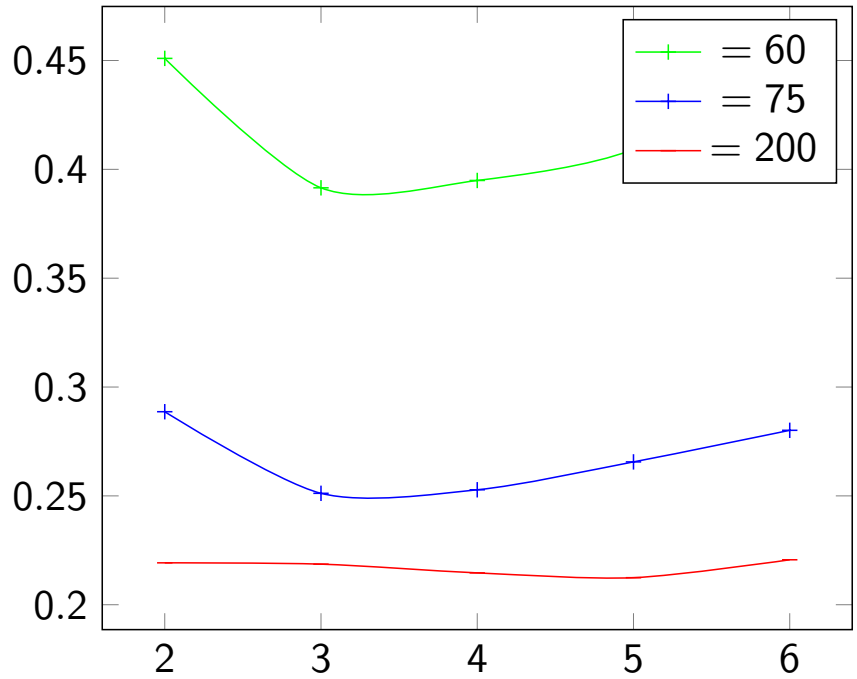
False positive rate

Unrecoverable Hamming Loss

- Due to the false positive rate $fp(p, P, K)$ which depends on:
  - p the size of the BF
  - P the marginal distribution on the label sets
  - K ..
- Should be made negligible wrt overall prediction error
- It is possible to derive good estimation of *reasonnable* B from the dataset distribution on label sets

Performance on rcv1 (100 labels)





Left: Hamming loss as a function of the BF's size p for the *Industries* dataset. The curves corrspond to various values of the number of hash function K.

Right: Hamming loss as a function of the number of hash function K for the *Industries* dataset. The curves corrspond to various values of BF's size p.

Performances with comparative methods

Comparison of Binary Relevance (BR), Pruned Binary Relevance (cf. Dekel), Bloom Filter with standard decoding (BF-SD) and improved decoding (BF-CD)

Classifier	NC	HL	m-F1	M-F1	NC	HL	m-F1	M-F1
	RCV-Industries				Wikipedia1K			
BR	303	0.200	72.43	47.82	1000	0.0711	55.96	34.7
BR-Dekel	75	0.360	30.00	15.72	100	0.1070	8.11	4.99
	150	0.308	46.98	30.14	250	0.0984	22.18	12.16
	200	0.233	65.78	40.09	500	0.0868	38.33	24.52
BF-SD	75	0.246	63.43	34.76	100	0.0801	46.03	22.35
	100	0.223	67.45	40.29	250	0.0742	53.02	31.41
	200	0.217	68.32	40.95	500	0.0734	53.90	32.57
BF-CD	75	0.251	63.74	37.37	100	0.0778	49.97	24.80
	100	0.218	68.42	42.20	250	0.0726	54.79	32.35
	200	0.212	70.07	43.37	500	0.0713	55.79	34.23

Current work: Design class codes using some information about labels.

Conclusion

- No clear winner among flat, hierarchical, IR, label embedding...
- Usually best peforming methods exploit many ideas
- Various settings (organization vs. tags) require various solution
- Medium scale problems (thousands to tens of thousands)
  - Flat methods are probably difficult to beat... but no really scalable such method for multilabelclassification
- Large scale problems (more than tens of thousands)
  - Hierarchical are mandatory
  - For texts: Hierarchies may be useful provided some cleaning / simplification
  - For images: Still much work to be done for automatc learning of hierarchies from data

[BWG10] Samy Bengio, Jason Weston, and David Grangier, *Label embedding trees for large multi-class tasks*, Advances in Neural Information Processing Systems 23 (J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, eds.), 2010, pp. 163–171.

[CAG12] M. Cissé, Thierry Artières, and Patrick Gallinari, *Learning compact class codes for fast inference in large multi class classification*, ECML/PKDD (1), 2012, pp. 506–520.

[CH04] Lijuan Cai and Thomas Hofmann, *Hierarchical document categorization with support vector machines*, Proceedings of the thirteenth ACM international conference on Information and knowledge management, 2004, pp. 78–87.

[CLCF07] E.P. Costa, A.C. Lorena, A.C.P.L.F. Carvalho, and A.A. Freitas, *A review of performance evaluation measures for hierarchical classifiers.*, Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop, AAAI Technical Report WS-07-05, July 2007, pp. 1–6.

[DS10] Ofer Dekel and Ohad Shamir, *Multiclass-multilabel classification with more classes than examples*, vol. 9, 2010, pp. 137–144.

[LYW+05] Tie-Yan Liu, Yiming Yang, Hao Wan, Qian Zhou, Bin Gao, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma, *An experimental study on large-scale web categorization*, WWW (Special interest tracks and posters), 2005, pp. 1106–1107.

[SBG10] J. Weston S. Bengio and D. Grangier, *Label embedding trees for large multi-class tasks*, NIPS, 2010.

[SF11] Carlos N. Silla, Jr. and Alex A. Freitas, *A survey of hierarchical classification across different application domains*, Data Min. Knowl. Discov. **22** (2011), no. 1-2, 31–72.

[TKV10] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Random k-labelsets for multi-label classification*, IEEE Transactions on Knowledge Discovery and Data Engineering, 2010.