On-line Multi-label Classification

A Problem Transformation Approach

Jesse Read

Supervisors: Bernhard Pfahringer, Geoff Holmes

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Hamilton, New Zealand

Outline

- Multi-label Classification
- Problem Transformation
  - Binary Method
  - Combination Method
- Pruned Sets Method (PS)
- Results
- On-line Applications
- Summary

Multi-label Classification

- Single-label Classification
  - Set of instances, set of labels
  - Assign one label to each instance
  - e.g. ”`Shares plunge on financial fears`”, `Economy`

Multi-label Classification

- Single-label Classification
  - Set of instances, set of labels
  - Assign one label to each instance
  - e.g. ”`Shares plunge on financial fears`”, `Economy`
- Multi-label Classification
  - Set of instances, set of labels
  - Assign a subset of labels to each instance
  - e.g. ”`Germany agrees bank rescue`”, `{Economy,Germany}`

Applications

- Text Classification:
  - News articles; Encyclopedia articles; Academic papers; Web directories; E-mail; Newsgroups
- Images, Video, Music:
  - Scene classification; Genre classification
- Other:
  - Medical classification; Bioinformatics

N.B. Not the same as *tagging / keywords*.

Multi-label Issues

- Relationships between labels
  - e.g. consider: `{US, Iraq}` vs `{Iraq, Antarctica}`
- Extra dimension
  - Imbalances exaggerated
  - Extra complexity
- Evaluation methods
  - Evaluate by label? by example?
- How to do Multi-label Classification?

Problem Transformation

1. Transform multi-label data into single-label data
2. Use one or more single-label classifiers
3. Transform classifications back into multi-label representation

- Can employ *any* single-label classifier
  - *Naive Bayes, SVMs, Decision Trees, etc, ...*
- *e.g. Binary Method, Combination Method, .. (overview by (Tsoumakas & Katakis, 2005))*

Algorithm Transformation

1. Adapts a single-label algorithm to make multi-label classifications
2. Runs directly on multi-label data

- Specific to a particular type of classifier
- Does some form of *Problem Transformation* internally
- *e.g. To AdaBoost* *(Schapire & Singer, 2000)*, *Decision Trees* *(Blockheel et al. 2008)*, *kNN* *(Zhang & Zhou. 2005)*, *NB* *(McCallum. 1999)*, ...

Outline

- Multi-label Classification
- Problem Transformation
  - Binary Method
  - Combination Method
- Pruned Sets Method (PS)
- Results
- On-line Applications
- Summary

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Multi-label Train

L = {A,B,C,D}

d0,{A,D}
d1,{C,D}
d2,{A}
d3,{B,C}

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Multi-label Train $L = \{A,B,C,D\}$		*SL Train* $L' = \{A,!A\}$	*SL Train* $L' = \{B,!B\}$	*SL Train* $L' = \{C,!C\}$	*SL Train* $L' = \{D,!D\}$
d0,{A,D}		d0,A	d0,!B	d0,!C	d0,D
d1,{C,D}	→	d1,!A	d1,!B	d1,C	d1,D
d2,{A}		d2,A	d2,!B	d2,!C	d2,!D
d3,{B,C}		d3,!A	d3,B	d3,C	d3,!D

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Multi-label Train $L = \{A,B,C,D\}$	*SL Train* $L' = \{A,!A\}$	*SL Train* $L' = \{B,!B\}$	*SL Train* $L' = \{C,!C\}$	*SL Train* $L' = \{D,!D\}$
d0,{A,D}	d0,A	d0,!B	d0,!C	d0,D
d1,{C,D}	d1,!A	d1,!B	d1,C	d1,D
d2,{A}	d2,A	d2,!B	d2,!C	d2,!D
d3,{B,C}	d3,!A	d3,B	d3,C	d3,!D
Single-label Test:	dx,*?*	dx,*?*	dx,*?*	dx,*?*

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Multi-label Train *L = {A,B,C,D}*	*SL Train* *L' = {A,!A}*	*SL Train* *L' = {B,!B}*	*SL Train* *L' = {C,!C}*	*SL Train* *L' = {D,!D}*
d0,{A,D}	d0,A	d0,!B	d0,!C	d0,D
d1,{C,D}	d1,!A	d1,!B	d1,C	d1,D
d2,{A}	d2,A	d2,!B	d2,!C	d2,!D
d3,{B,C}	d3,!A	d3,B	d3,C	d3,!D
Single-label Test:	dx,!A	dx,!B	dx,C	dx,D

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Multi-label Train $L = \{A,B,C,D\}$	*SL Train* $L' = \{A,!A\}$	*SL Train* $L' = \{B,!B\}$	*SL Train* $L' = \{C,!C\}$	*SL Train* $L' = \{D,!D\}$
d0,{A,D}	d0,A	d0,!B	d0,!C	d0,D
d1,{C,D}	d1,!A	d1,!B	d1,C	d1,D
d2,{A}	d2,A	d2,!B	d2,!C	d2,!D
d3,{B,C}	d3,!A	d3,B	d3,C	d3,!D
Single-label Test:	dx,!A	dx,!B	dx,C	dx,D

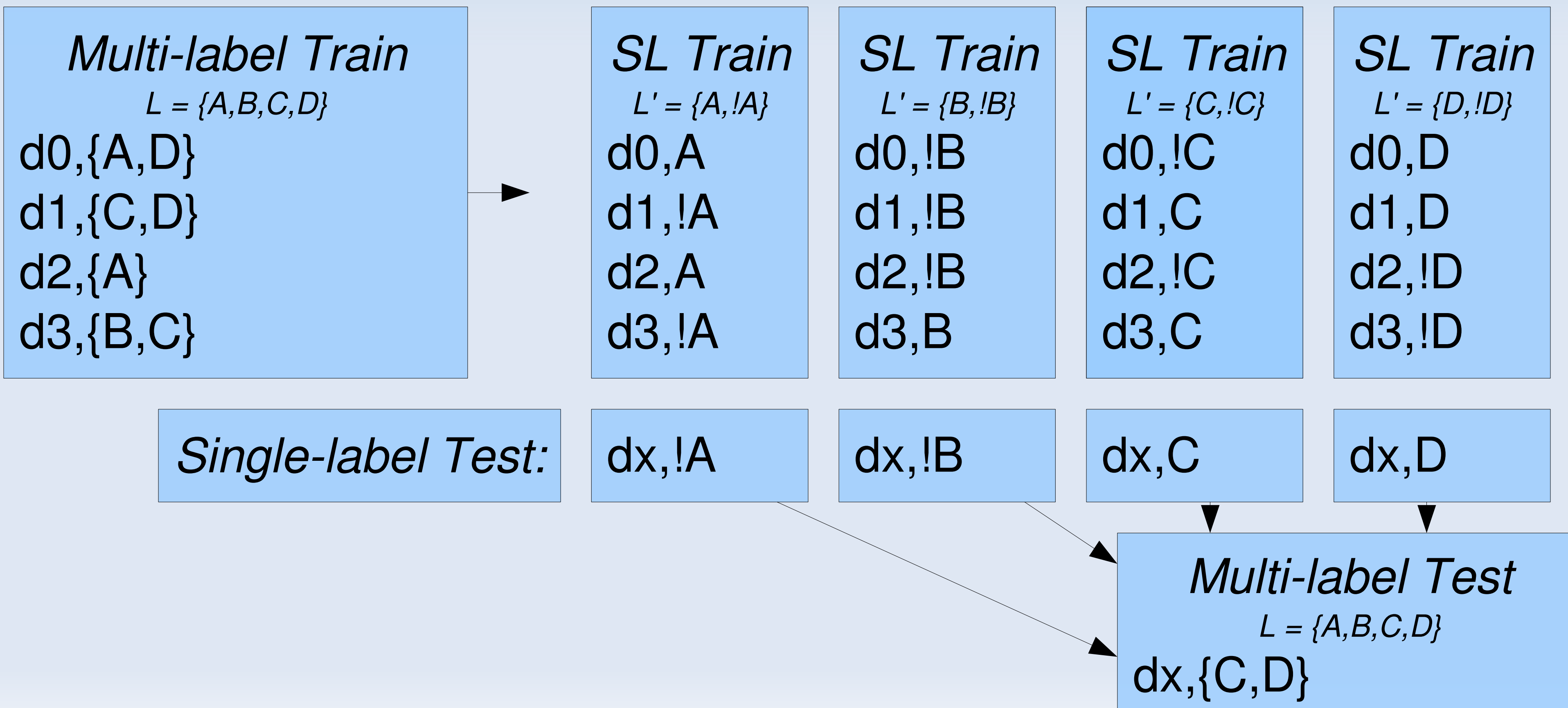
Multi-label Test

$L = \{A,B,C,D\}$

dx,*???*

Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant



Binary Method

- One binary classifier for each label
- A label is either relevant or !relevant

Multi-label Train $L = \{A,B,C,D\}$	*SL Train* $L' = \{A,!A\}$	*SL Train* $L' = \{B,!B\}$	*SL Train* $L' = \{C,!C\}$	*SL Train* $L' = \{D,!D\}$
d0,{A,D}	d0,A	d0,!B	d0,!C	d0,D
d1,{C,D}	d1,!A	d1,!B	d1,C	d1,D
d2,{A}	d2,A	d2,!B	d2,!C	d2,!D
d3,{B,C}	d3,!A	d3,B	d3,C	d3,!D
Single-label Test:	dx,!A	dx,!B	dx,C	dx,D

Multi-label Test

$L = \{A,B,C,D\}$

dx,{C,D}

- Assumes label independence
- Often unbalanced by many negative examples

Combination Method

- One decision involves multiple labels
- Each subset becomes a single label

Combination Method

- One decision involves multiple labels
- Each subset becomes a single label

Multi-label Train

L = {A,B,C,D}

d0,{A,D}
d1,{C,D}
d2,{A}
d3,{B,C}

Combination Method

- One decision involves multiple labels
- Each subset becomes a single label

Multi-label Train

L = {A,B,C,D}

d0,{A,D}
d1,{C,D}
d2,{A}
d3,{B,C}

Single-label Train

L' = {A,AD,BC,CD}

d0,AD
d1,CD
d2,A
d3,BC

Combination Method

- One decision involves multiple labels
- Each subset becomes a single label

Multi-label Train

L = {A,B,C,D}

d0,{A,D}
d1,{C,D}
d2,{A}
d3,{B,C}

→

Single-label Train

L' = {A,AD,BC,CD}

d0,AD
d1,CD
d2,A
d3,BC

Single-label Test

L' = {A,AD,BC,CD}

dx,*???*

Combination Method

- One decision involves multiple labels
- Each subset becomes a single label

Multi-label Train

L = {A,B,C,D}

d0,{A,D}
d1,{C,D}
d2,{A}
d3,{B,C}

Single-label Train

L' = {A,AD,BC,CD}

d0,AD
d1,CD
d2,A
d3,BC

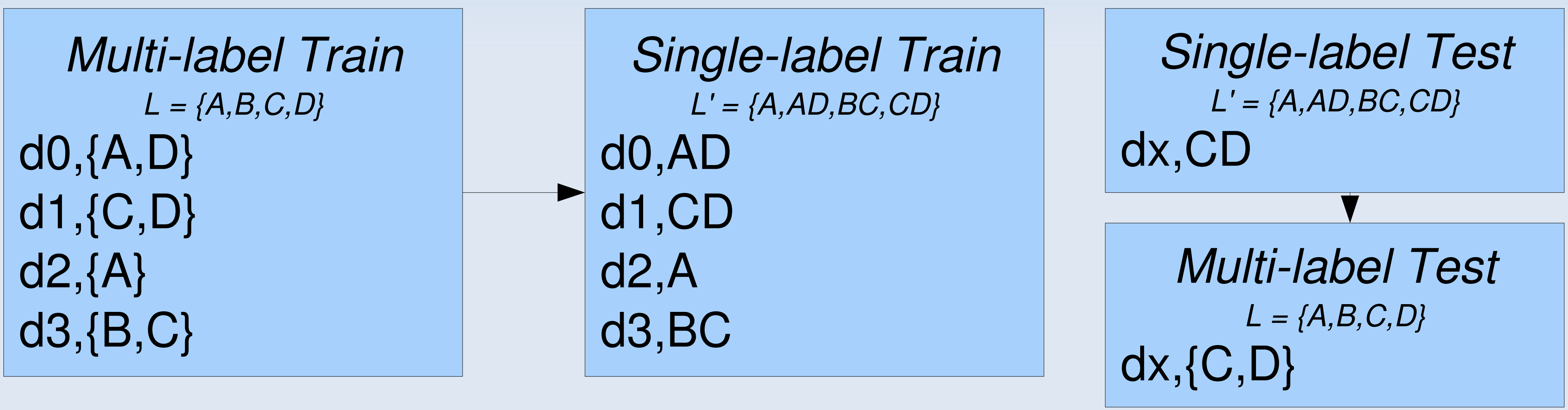
Single-label Test

L' = {A,AD,BC,CD}

dx,CD

Combination Method

- One decision involves multiple labels
- Each subset becomes a single label



Combination Method

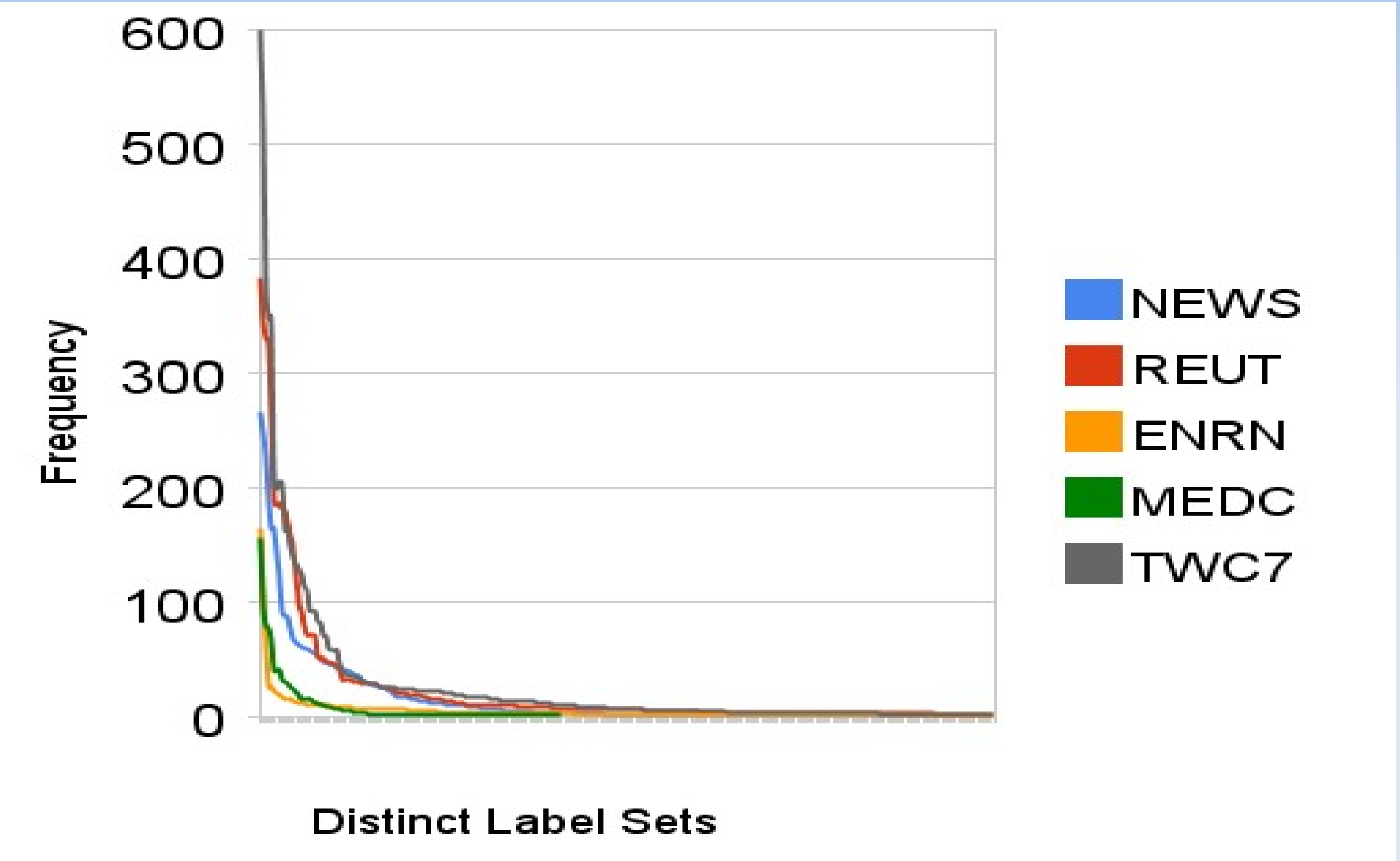
- One decision involves multiple labels
- Each subset becomes a single label

- May generate too many single labels
- Can only predict combinations seen in the training set

A Pruned Sets Method (PS)

- Binary Method
  - Assumes label independence
- Combination Method
  - Takes into account combinations
  - Can't adapt to new combinations
  - High complexity (~ distinct label sets)
- Pruned Sets Method
  - Use *pruning* to focus on *core* combinations

A Pruned Sets Method (PS)



Concept:

- Prune away and break apart infrequent label sets
- Form new examples with more frequent label sets

A Pruned Sets Method (PS)

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d06,{Animation,Comedy,Family,Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}
d12,{Adult,Animation}

A Pruned Sets Method (PS)

1. **Count label sets**

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d06,{Animation,Comedy,Family,Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}
d12,{Adult,Animation}

{Animation,Comedy}	3
{Animation,Family}	2
{Adult}	3
{Animation,Comedy,Family,Musical}	1
{Musical}	2
{Adult,Animation}	1

A Pruned Sets Method (PS)

1. Count label sets
2. **Prune infrequent sets (e.g. count < 2)**

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}

d12,{Adult,Animation}
d06,{Animation,Comedy,Family,Musical}

{Animation,Comedy}	3
{Animation,Family}	2
{Adult}	3
{Animation,Comedy,Family,Musical}	1
{Musical}	2
{Adult,Animation}	1

Information loss!

A Pruned Sets Method (PS)

1. Count label sets
2. Prune infrequent sets (e.g. count < 2)
3. **Break up infrequent sets into frequent sets (e.g. count >= 2)**

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}

d12,{Adult,Animation}
d12,{Adult}
d06,{Animation,Comedy,Family,Musical}
d06,{Animation,Comedy}
d06,{Animation,Family}
d06,{Musical}

{Animation,Comedy}	3
{Animation,Family}	2
{Adult}	3
{Animation,Comedy,Family,Musical}	1
{Musical}	2
{Adult,Animation}	1

A Pruned Sets Method (PS)

1. Count label sets
2. Prune infrequent sets (e.g. count < 2)
3. Break up infrequent sets into frequent sets (e.g. count >= 2)

4. Decide which subsets to reintroduce

(!) Too many (esp. small) sets will:

- 'dillute' the dataset with single-labels
- vastly increase the training set size

i.e. frequent item sets not desireable

{Animation,Comedy}	3
{Animation,Family}	2
{Adult}	3
{Animation,Comedy,Family,Musical}	1
{Musical}	2
{Adult,Animation}	1

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}

d12,{Adult,Animation}
d12,{Adult}
d06,{Animation,Comedy,Family,Musical}
d06,{Animation,Comedy}
d06,{Animation,Family}
d06,{Musical}

A Pruned Sets Method (PS)

1. Count label sets
2. Prune infrequent sets (e.g. count < 2)
3. Break up infrequent sets into frequent sets (e.g. count >= 2)
4. **Decide which subsets to reintroduce**

<u>Strategies:</u>

A. Keep the top *n* subsets
(ranked by *number of labels* and *count*)
-or-
B. Keep all subsets of size greater than *n*

{Animation,Comedy}	3
{Animation,Family}	2
{Adult}	3
{Animation,Comedy,Family,Musical}	1
{Musical}	2
{Adult,Animation}	1

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}

d12,{Adult,Animation}
d12,{Adult}
d06,{Animation,Comedy,Family,Musical}
d06,{Animation,Comedy}
d06,{Animation,Family}
d06,{Musical}

A Pruned Sets Method (PS)

1. Count label sets
2. Prune infrequent sets (e.g. count < 2)
3. Break up infrequent sets into frequent sets (e.g. count >= 2)
4. Decide which subsets to reintroduce
5. **Add new instances**

E.g. 12 examples, 6 combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}

d12,{Adult,Animation}
d12,{Adult}
d06,{Animation,Comedy,Family,Musical}
d06,{Animation,Comedy}
d06,{Animation,Family}
d06,{Musical}

{Animation,Comedy}	3
{Animation,Family}	2
{Adult}	3
{Animation,Comedy,Family,Musical}	1
{Musical}	2
{Adult,Animation}	1

A Pruned Sets Method (PS)

1. Count label sets
2. Prune infrequent sets (e.g. count < 2)
3. Break up infrequent sets into frequent sets (e.g. count >= 2)
4. Decide which subsets to reintroduce
5. Add new instances
6. *Use Combination Method transformation*

E.g. **15** examples, **4** combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}
d06,{Animation,Comedy}
d06,{Animation,Family}
d12,{Adult}

{Animation,Comedy}	4
{Animation,Family}	3
{Adult}	4
{Musical}	2

A Pruned Sets Method (PS)

1. Count label sets
2. Prune infrequent sets (e.g. count < 2)
3. Break up infrequent sets into frequent sets (e.g. count >= 2)
4. Decide which subsets to reintroduce
5. Add new instances
6. *Use Combination Method transformation*

- Accounts for label relationships
- Reduced complexity
- Cannot form new combinations (e.g. {Animation,Family,Musical})

{Animation,Comedy}	4
{Animation,Family}	3
{Adult}	4
{Musical}	2

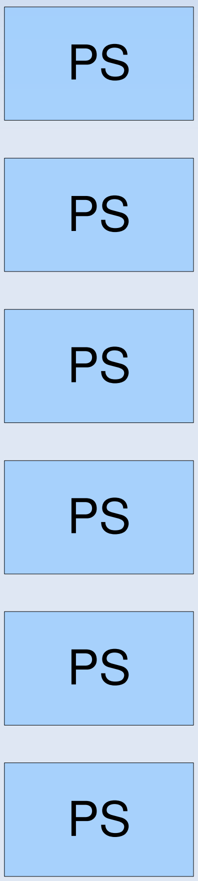
E.g. **15** examples, **4** combinations

d01,{Animation,Family}
d02,{Musical}
d03,{Animation,Comedy }
d04,{Animation,Comedy}
d05,{Musical}
d07,{Adult}
d08,{Adult}
d09,{Animation,Comedy}
d10,{Animation,Family}
d11,{Adult}
d06,{Animation,Comedy}
d06,{Animation,Family}
d12,{Adult}

Ensembles of Pruned Sets (E.PS)

Creating new label set classifications

1. Train an Ensemble of PS e.g. Bagging (introduces variation!)



Ensembles of Pruned Sets (E.PS)

Creating new label set classifications

1. Train an Ensemble of PS e.g. Bagging (introduces variation!)
2. Get preditions

PS	{Musical}
PS	{Animation,Family}
PS	{Animation, Comedy}
PS	{Animation, Family}
PS	{Musical}
PS	{Musical}

Ensembles of Pruned Sets (E.PS)

Creating new label set classifications

1. Train an Ensemble of PS e.g. Bagging (introduces variation!)
2. Get preditions
3. **Calculate a score**

PS {Musical}

PS {Animation,Family}

PS {Animation, Comedy}

PS {Animation, Family}

PS {Musical}

PS {Musical}

Musical:	3 (0.33)
Animation:	3 (0.33)
Family:	2 (0.22)
Comedy:	1 (0.11)

Ensembles of Pruned Sets (E.PS)

Creating new label set classifications

1. Train an Ensemble of PS e.g. Bagging (introduces variation!)
2. Get preditions
3. Calculate a score
4. **Form a classification set**

dx,{Animation, Family, Musical}

PS {Musical}

PS {Animation,Family}

PS {Animation, Comedy}

PS {Animation, Family}

PS {Musical}

PS {Musical}

Musical:	3 (0.33)
Animation:	3 (0.33)
Family:	2 (0.22)
Comedy:	1 (0.11)

Threshold = 0.15

Ensembles of Pruned Sets (E.PS)

Creating new label set classifications

1. Train an Ensemble of PS e.g. Bagging (introduces variation!)
2. Get preditions
3. Calculate a score
4. Form a classification set

dx,{Animation, Family, Musical}

- Can form new combinations

PS	{Musical}
PS	{Animation,Family}
PS	{Animation, Comedy}
PS	{Animation, Family}
PS	{Musical}
PS	{Musical}

Musical:	3 (0.33)
Animation:	3 (0.33)
Family:	2 (0.22)
Comedy:	1 (0.11)

Threshold = 0.15

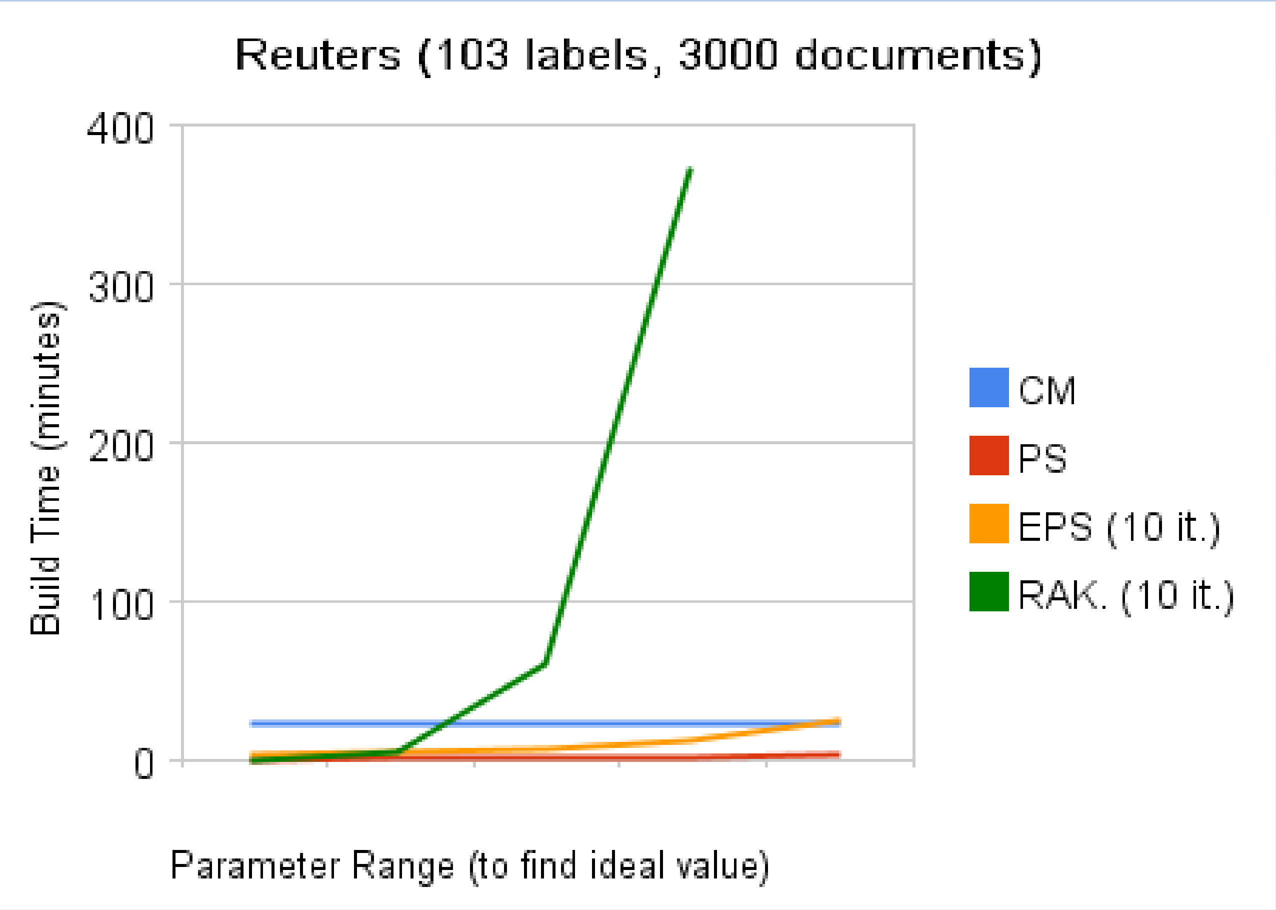
Results – F1 Measure

D.SET	size /	#lbls /	avg.lbls	BM	[CM]	PS	E.PS	RAK.
Scene	2407	6	1.1	*0.671*	0.729	0.730	**0.752**	0.735
Medical	978	45	1.3	**0.791**	0.767	0.766	0.764	0.784
Yeast	2417	14	4.2	0.630	0.633	0.643	**0.665**	**0.664**
Enron	1702	53	3.4	0.504	0.502	0.520	**0.543**	**0.543**
Reuters	6000	103	1.5	*0.421*	0.482	0.496	**0.499**	*0.418*

- Combination Method (CM) improves Binary Method (BM)
- Puned Sets method (PS) improves Combination Method (CM)
  - Except *Medical*: maybe label relationships not as important
- E.PS is best overall.
- RAKEL ~ EPS similar
- What about complexity?

J. Read, B. Pfahringer, G. Homes. To Appear ICDM 08.

Complexity – Build Time



- RAKEL may not be able to find ideal parameter value
- '*Worst case*' scenarios are similar, but different in practice

J. Read, B. Pfahringer, G. Homes. To Appear ICDM 08.

Complexity – Memory Use

Reuters Dataset

•**PS** transformation:	~**2,500**	instances	
•**EPS** transformation:	~**25,000**	instances	(for 10 iterations)
•**RAKEL** transformation:	**3,090,000**	instances	(for 10 iterations)

Number of instances generated during the *Problem Transformation* procedure for *most complex* parameter setting

J. Read, B. Pfahringer, G. Homes. To Appear ICDM 08.

Outline

- Multi-label Classification
- Problem Transformation
  - Binary Method
  - Combination Method
- Pruned Sets Method (PS)
- Results
- On-line Applications
- Summary

On-line Multi-label Classification

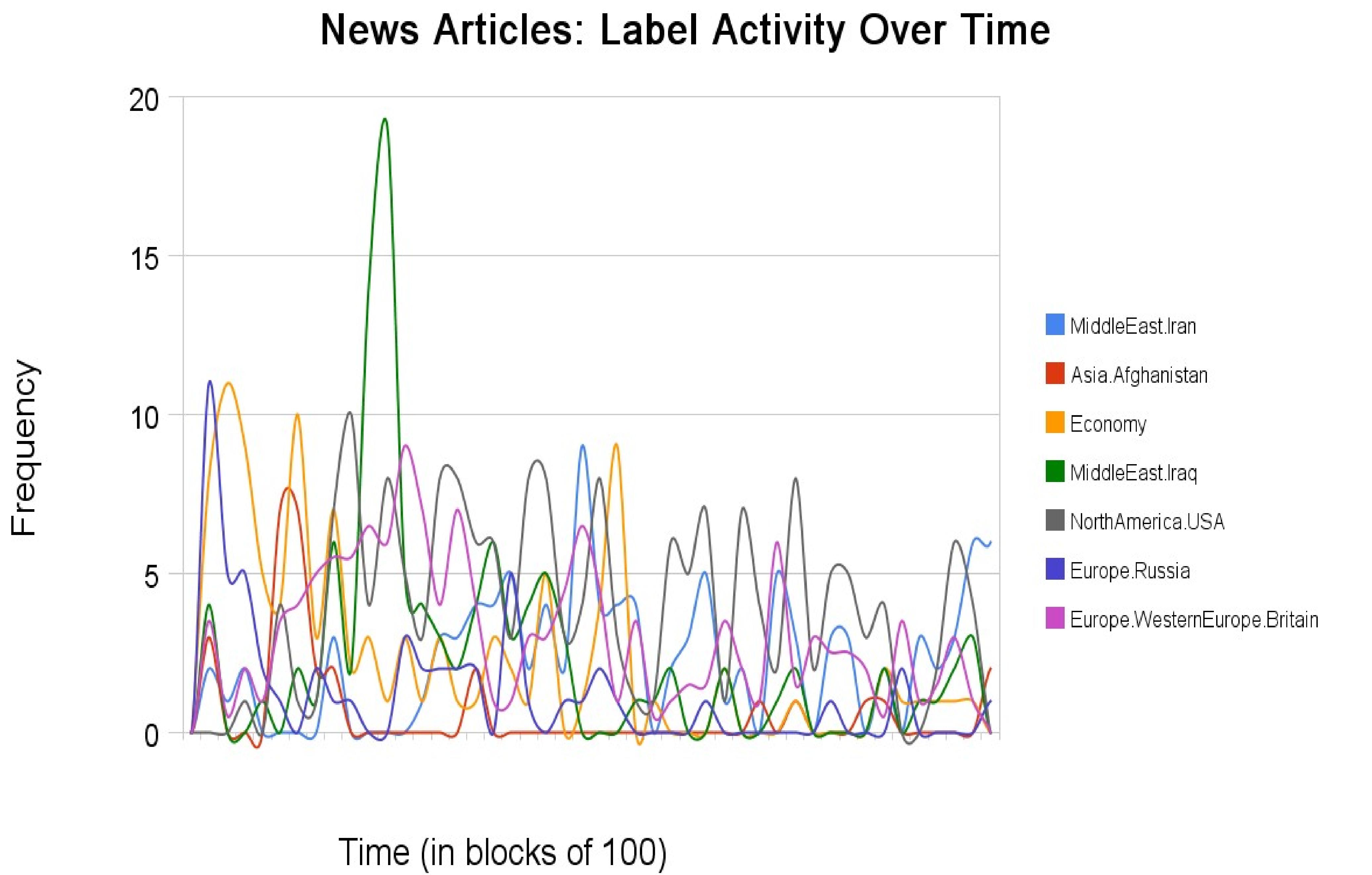
Many multi-label data sources are *on-line*:

- New instances incoming
- Data can be time ordered
- Possibly large collections
- Concept drift

An on-line multi-label algorithm should be:

- Adaptive
- Efficient

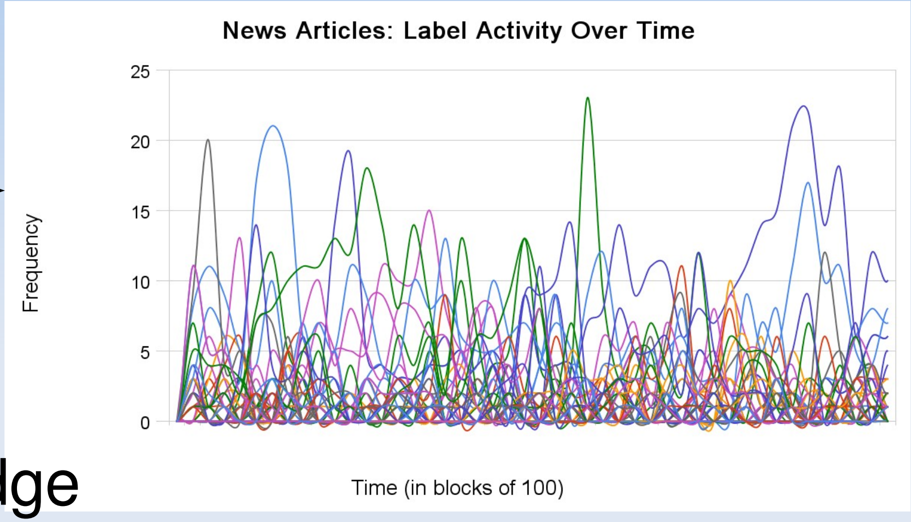
On-line Multi-label Classification



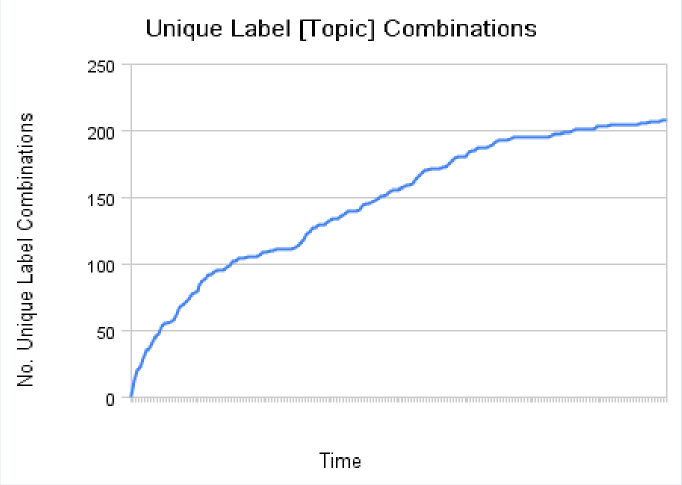
Multi-label Concept Drift

Measuring concept drift

- Observing indiv. labels?
  - Complicated
    (may be 1000's of labels)
  - May need domain knowledge

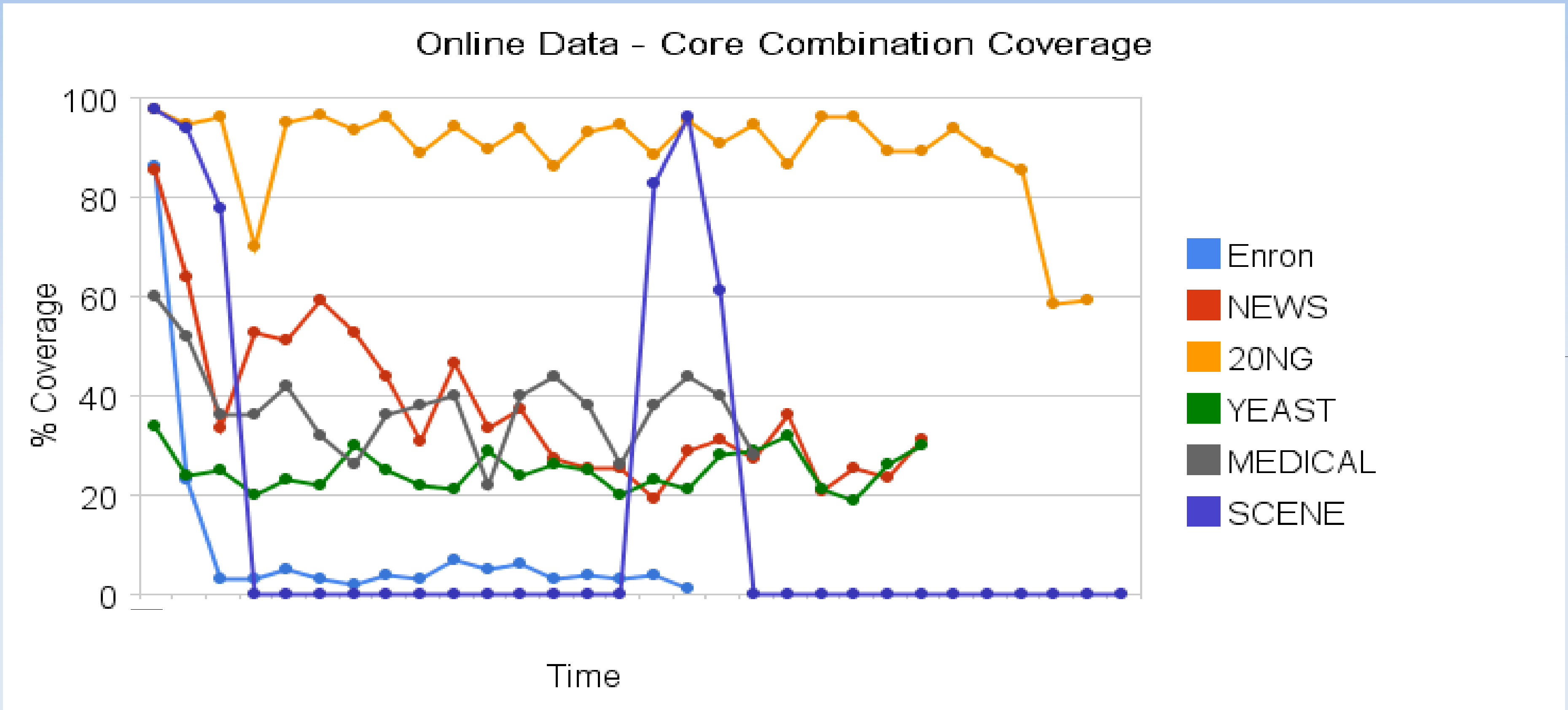


- Counting distinct label *sets*?
  - Doesn't tell us much



- PS Transformation?
  - Focus on core combinations

Multi-label Concept Drift



20NG; News; Enron –(***On-line data***)– Slow; medium; rapid **concept drift**

YEAST – ***Randomised***

SCENE – ***Ordered*** *Train/Test Split*

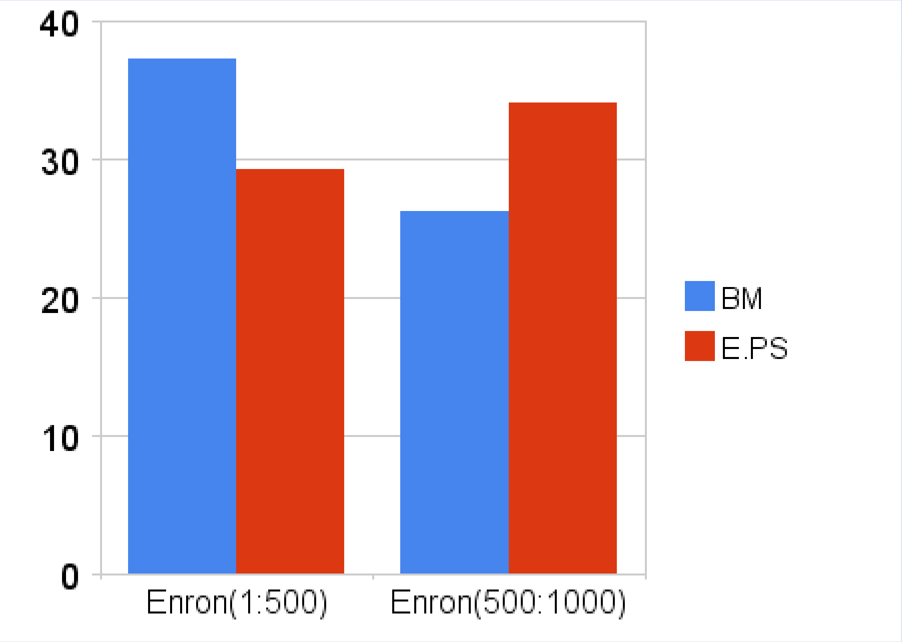
MEDICAL – ***???***

1. PS transformation on first 50 instances
2. Measure the % coverage
3. Measure on the next 50, and etc ..

Preliminary Results

- 'On-line' Binary Method vs E.PS
  - Model(s) built on 100 instances
  - Thresholds updated every instance
  - Model(s) rebuilt every 25 instances

Enron Dataset - Subsets - Accuracy



Summary

- Multi-label Classification
- Problem Transformation
  - Binary Method (BM), Combination Method (CM)
- Pruned Sets (PS) and Ensembles of PS (E.PS)
  - Focus on core label relationships via pruning
  - Outperforms standard and state-of-the-art methods
- Multi-label Classification in an On-line Context
  - Naive methods (eg. BM) can perform better than EPS in an on-line context (future work!)

Questions

?