Instance Driven Hierarchical Clustering of Document Collections

and

Classification by Pattern-Based Hierarchical Clustering

Hassan H. Malik and John R. Kender

September 15th, 2008

Computer Science Department

Columbia University

in the city of New York

Outline

- Motivation
- Instance-driven Pattern Mining
- IDHC: A More Flexible Pattern-based Hierarchical Clustering Algorithm
- CPHC: Semi-supervised Classification by Pattern-based Hierarchical Clustering
- Conclusions

Outline

- **Motivation**
- Instance-driven Pattern Mining
- IDHC: A More Flexible Pattern-based Hierarchical Clustering Algorithm
- CPHC: Semi-supervised Classification by Pattern-based Hierarchical Clustering
- Conclusions

Motivation

- Traditional pattern-mining suffers from
  - Frequency-based pattern significance measures
  - Global thresholds
- Pattern-based hierarchical clustering suffers from
  - An unpredictable number of patterns
  - Unnecessary coupling between pattern size and node height
  - Artificial constraints on soft clustering

Motivation - continued

- Inductive classifiers may not fully exploit the distribution of test instances in the context of the whole dataset
- Existing semi-supervised classification algorithm weaknesses
  - Dependence on flat clustering requires the number of clusters to be known in advance
  - Unnecessary step of training a classifier on the expanded training set

Outline

- Motivation
- **Instance-driven Pattern Mining**
- IDHC: A More Flexible Pattern-based Hierarchical Clustering Algorithm
- CPHC: Semi-supervised Classification by Pattern-based Hierarchical Clustering
- Conclusions

Traditional Pattern Mining

- Aims to mine a set of globally significant patterns from the dataset
- Does not consider local pattern significance
- Traditionally uses a frequency-based pattern significance measure (i.e., Support), and a global threshold
- "Closed interesting" itemsets (Malik and Kender ICDM'06) replaced support with an interestingness measure
  - Still, no coverage guarantees
  - Thresholds not as stable on highly correlated datasets
  - An unpredictable number of resulting patterns

Instance-driven Pattern Mining

- Eliminate the global mining step altogether
- Allow each instance to “vote” for its representative size-2 patterns, balancing global and local pattern significance
  - Sort all patterns in decreasing order of local term frequency * global term interestingness
  - Select all patterns with scores exceeding *min_standard_deviation*
  - Number of patterns-per-instance upper bounded by a small constant *maxK*
  - Why size-2? Why not size-3 etc.?

Instance-driven Pattern Mining - advantages

- Coverage guaranteed
- No global threshold
- *min_standard_deviation* robust across datasets (experimented on 16 datasets)
- A small number of highly significant patterns for each instance
  - Central limit theorem for normally distributed scores
  - Chebyshev's inequality for the rest
- Number of size-2 patterns linear to the number of instances
  - *maxK* provide empirical upper limit guarantee

Instance-driven Patterns vs. Closed Interesting Itemsets

Dataset	*#instances*	***#features***	*Approx. number of size-2 patterns*		
			GPHC, MI	*GPHC, YulesQ*	*Ours*
mm	2,521	126,373	2.4 million	{fails}	**3,651**
reviews	4,069	126,373	2.6 million	{fails}	**5,952**
sports	8,580	126,373	1.4 million	{fails}	**12,607**
tr11	414	6,429	4.3 million	11.4 million	**604**
tr12	313	5,804	3.6 million	8.8 million	**464**
tr23	204	5,832	7.6 million	12.2 million	**282**
tr31	927	10,128	7.0 million	{fails}	**1,360**

Outline

- Motivation
- Instance-driven Pattern Mining
- **IDHC: A More Flexible Pattern-based Hierarchical Clustering Algorithm**
- CPHC: Semi-supervised Classification by Pattern-based Hierarchical Clustering
- Conclusions

Instance-driven Pattern-based Hierarchical Clustering

- Each size-2 pattern forms an initial cluster, patterns added to their selected-pattern-clusters
- Use instance-to-cluster relationships to prune duplicate (in content) clusters
  - Merge labels of duplicate clusters being removed, enhancing the cluster labels
- Generate rest of the cluster hierarchy by iteratively refining clusters
  - Make patterns progressively longer, and cluster memberships progressively sparser
  - Maintain instance-to-cluster pointers for local-only processing

Cluster Refinement – an example

(a) A transaction dataset as running example

Instance ID	*Features and their local frequencies*
T1	(A:2), (B:4), (D:1), (H:2), (J:4), (L:1)
T2	(A:3), (C:1), (D:6), (E:1), (G:4)
T3	(B:2), (C:3), (D:1), (I:5), (K:2)
T4	(B:3), (C:1), (D:2), (E:4), (J:3), (K:3), (L:2)
T5	(B:7), (C:2), (D:1), (H:3), (I:2)
T6	(A:1), (B:1), (C:1), (E:1), (J:3), (K:1)
T7	(B:9), (C:3), (F:4), (H:5), (J:1), (L:5)
T8	(C:6), (D:2), (G:1), (I:1), (K:3)
T9	(B:3), (D:2), (J:4), (K:1), (L:8)
T10	(A:4), (B:2), (D:7), (F:3), (I:6)
T11	(C:1), (E:1), (F:1), (G:2), (H:1), (I:4), (J:1)

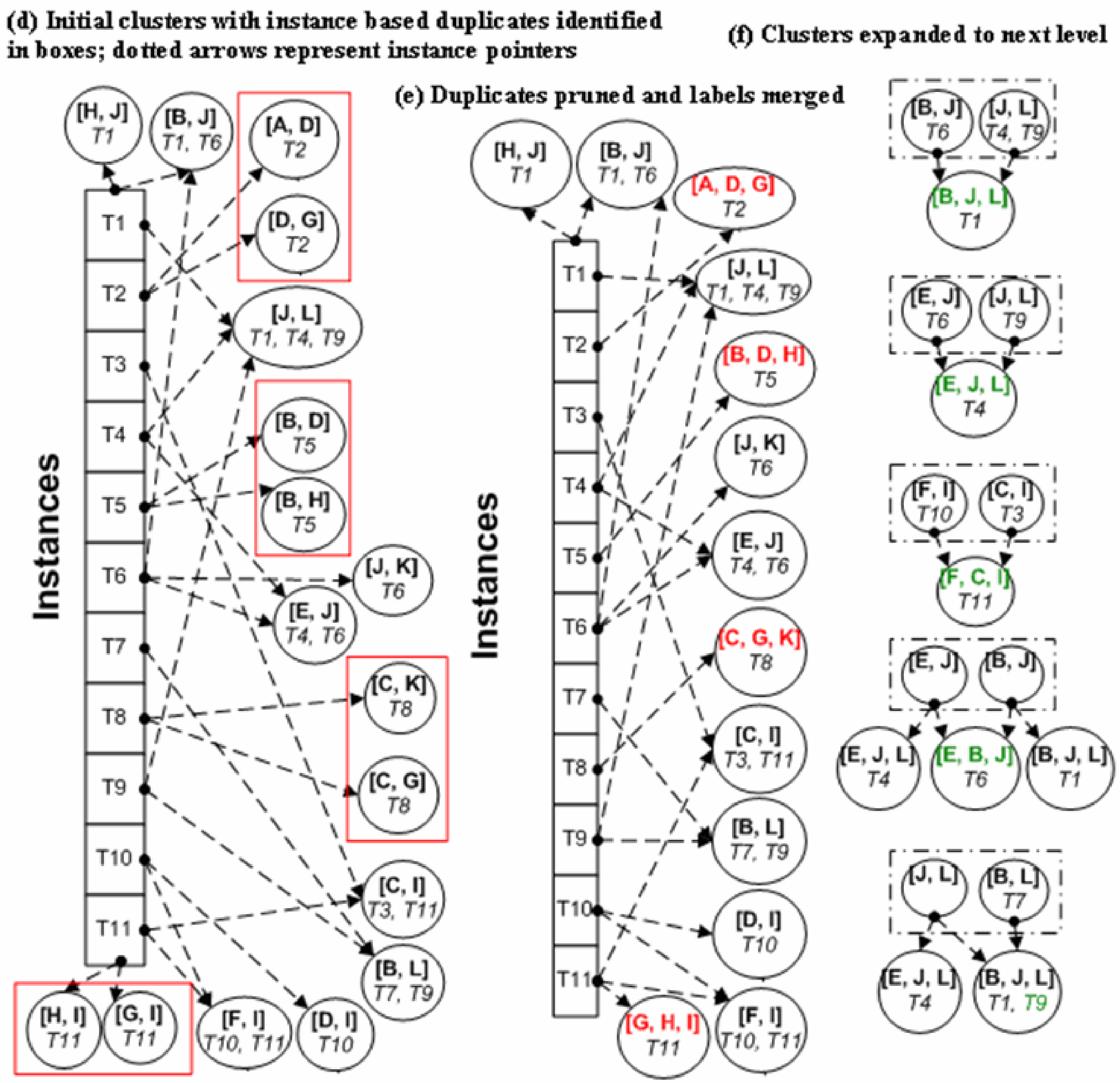
(b) Global significance values of some size-2 patterns using Added Value (transformed to positive scale)

Pattern	*AV*	*Pattern*	*AV*	*Pattern*	*AV*	*Pattern*	*AV*
(B,D)	0.52	(B, K)	0.57	(E, J)	0.70	(J, K)	0.55
(B, E)	0.38	(B, L)	0.77	(E, K)	0.54	(J, L)	0.95
(B, J)	0.60	(D, E)	0.38	(E, L)	0.38	(K, L)	0.54

(c) Instance pattern selection

Instance ID	*#size-2 patterns*	*Significance range*		*Min sig.*	*Selected patterns*
T1	15	0.52	2.42	1.95	(B, J) (J, L), (H, J)
T2	10	0.77	2.38	2.14	(D, G), (A, D)
T3	10	0.78	2.29	1.76	(C, I)
T4	21	0.57	2.46	1.93	(E, J) (J, L)
T5	10	0.59	2.61	2.05	(B, H) (B, D)
T6	15	0.33	1.40	1.03	(E, J) (B, J) (J, K)
T7	15	0.90	5.40	3.70	(B, L)
T8	10	0.71	2.70	2.16	(C, G) (C, K)
T9	10	0.85	5.72	3.79	(J, L) (B, L)
T10	10	1.19	3.72	2.95	(D, I) (F, I)
T11	21	0.38	2.13	1.33	(G, I) (F, I) (C, I) (H, I)

Cluster Refinement – an example



Instance-driven Hierarchical Clustering - advantages

- Number of initial patterns predictable
- Cluster refinement avoids global processing
- No coupling between node heights and pattern-lengths
  - More meaningful cluster labels
- More flexible soft clustering
  - Instances allowed to exist at multiple levels in the hierarchy
  - Instances not forced to their longest-pattern clusters
- Parameter values robust across datasets

Clustering Quality on Text Datasets

Dataset	*FScores*			*Entropies*		
	bi-k I_2	*GPHC*	*Ours*	*bi-k I_2*	*GPHC*	*Ours*
reuters	0.835	**0.851**	0.846	0.075	0.155	**0.005**
classic	0.782	**0.88**	0.759	0.06	0.025	**0.021**
hitech	0.528	0.54	**0.544**	0.224	0.172	**0.074**
k1a	0.668	0.654	**0.676**	0.106	0.045	**0.041**
k1b	0.882	**0.903**	0.897	0.042	0.042	**0.021**
la12	0.741	0.661	**0.748**	0.12	0.062	**0.038**
mm	0.774	**0.943**	0.909	0.073	0.053	**0.014**
ohscal	**0.601**	0.53	0.554	0.198	0.237	**0.081**
re0	0.61	**0.672**	0.615	0.115	0.077	**0.016**
reviews	0.801	0.818	**0.833**	0.073	0.048	**0.013**
sports	0.882	**0.886**	0.87	0.03	0.016	**0.005**
tr11	**0.795**	0.519	0.79	0.107	0.141	**0.038**
tr12	0.689	0.604	**0.769**	0.133	0.161	**0.037**
tr23	0.667	0.487	**0.679**	0.136	0.042	**0.038**
tr31	0.837	0.584	**0.84**	0.041	0.114	**0.013**
wap	**0.683**	0.663	0.67	0.106	0.047	**0.043**
average	0.736	0.7	**0.75**	0.102	0.09	**0.031**

Outline

- Motivation
- Instance-driven Pattern Mining
- IDHC: A More Flexible Pattern-based Hierarchical Clustering Algorithm
- **CPHC: Semi-supervised Classification by Pattern-based Hierarchical Clustering**
- Conclusions

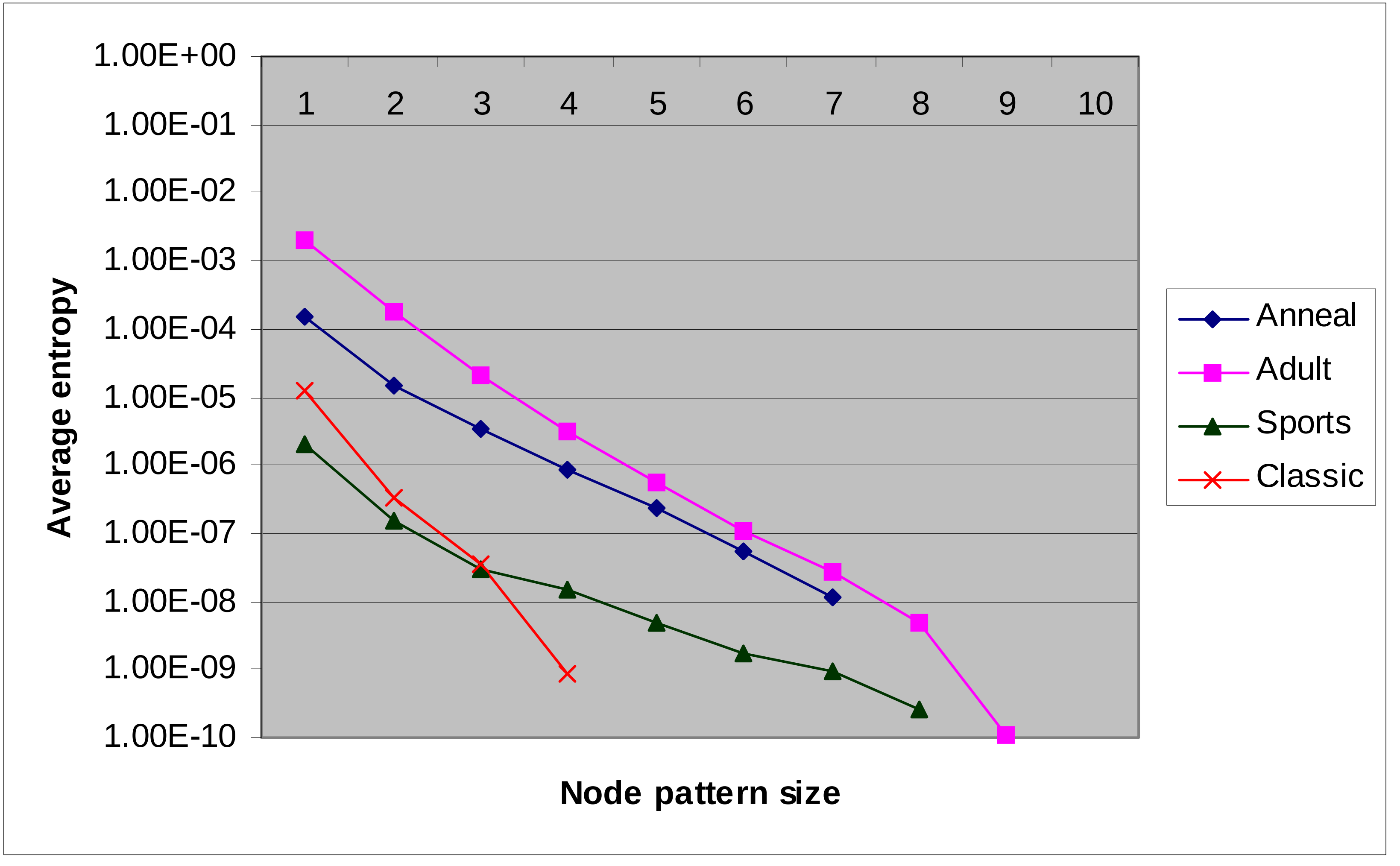
Existing Classifiers

- Inductive classifiers
  - Use instances in the training set to obtain a classification model
  - Use this classification model to determine class labels for test instances
  - Cons:
    - May not fully exploit the distribution of test instances in the context of the whole dataset
    - Poor classification performance when training data is sparse
- Semi-supervised classification algorithms
  - First (flat) cluster training and test sets together
  - Use the resulting clustering solution to enhance the training set
  - Cons:
    - Flat clustering requires the number of clusters to be known in advance
    - Extra step of training a classifier on the expanded training set

Pattern-based Cluster Hierarchies and Significance of Pattern Lengths

- Lower overall *Entropy* = a higher percentage of nodes that contain most instances that belong to the same ground truth class
- IDHC only assigns instances to their "selected" pattern clusters
  - Intuition: Nodes with longer patterns should have lower *Entropies*
- Experimented with 4 datasets to understand the class-label distributions over nodes with varying pattern-lengths

Average Node *Entropies* With Respect to Pattern Sizes



The CPHC Classification Algorithm

- **Feature selection**
  - Use a supervised method for training instances
  - Use an unsupervised method for test instances
  - Ensure coverage
- **Clustering**
  - Apply the instance-driven, pattern-based hierarchical clustering algorithm (IDHC) on all training and test instances
  - Track interestingness values
- **Classification**
  - For each test instance *t*, traverse the hierarchy to identify the set *S* of clusters that contain *t*
  - Use interestingness values of clusters in *S*, and pattern-lengths as weights to compute class scores for *t*

Improving The Chances of Classifying Isolated Test Instances

- Classification model produced by inductive classifiers limited to patterns in training instances
  - No way of classifying isolated test instances
- Improving the chances of classifying such test instances by inducing a type of transitivity
  - Isolated test instances may be clustered together in a “logical” node with test instances that overlap the training set
  - The “logical” node contributes towards score

Breakeven Performance on Top 10 Reuters 21578 Categories

Category	*Harmony*	*Find Sim*	*Naïve Bayes*	*Bayes Nets*	*Trees*	*SVM (linear)*	*ARC-BC*	*Ours*
acq	**95.3**	64.7	87.8	88.3	89.7	93.6	90.9	94.5
corn	78.2	48.2	65.3	76.4	**91.8**	90.3	69.6	77.2
crude	85.7	70.1	79.5	79.6	85	88.9	77.9	**90.7**
earn	**98.1**	92.9	95.9	95.8	97.8	98	92.8	96.5
grain	91.8	67.5	78.8	81.4	85	**94.6**	68.8	91.1
interest	77.3	63.4	64.9	71.3	67.1	77.7	70.5	**81**
money-fx	80.5	46.7	56.6	58.8	66.2	74.5	70.5	**84.3**
ship	**86.9**	49.2	85.4	84.4	74.2	85.6	73.6	78.3
trade	**88.4**	65.1	63.9	69	72.5	75.9	68	87.9
wheat	62.8	68.9	69.7	82.7	**92.5**	91.8	84.8	83.6
micro-avg	92	64.6	81.5	85	88.4	92	82.1	**92.1**
macro-avg	84.5	63.7	74.8	78.8	82.2	**87.1**	76.7	86.5

Classification Accuracies on 13 Small and 2 Large UCI Datasets

	FOIL	***CPAR***	***SVM***	***Harmony***	***Ours***
anneal	**96.9**	90.2	83.83	91.51	93.82
auto	46.1	48	55.5	61	**73**
breast	94.4	94.8	**96.8**	92.42	93.33
glass	49.3	48	46	49.8	**70**
heart	57.4	51.1	**60.36**	56.46	58.33
hepatitus	77.5	76.5	81.83	83.16	**83.33**
horsecolic	**83.5**	82.3	83.31	82.53	73.61
ionoSphere	89.5	**92.9**	89.44	92.03	92.57
iris	94	**94.7**	94.67	93.32	94.67
pima	73.8	**75.6**	74.18	72.34	73.16
tic-tac-toe	**96**	72.2	70.78	92.29	72.74
wine	86.4	92.5	**94.9**	91.94	88.24
zoo	96	96	86	93	**97**
average	80.06	78.06	78.28	80.91	**81.83**

	FOIL	***CPAR***	***SVM***	***Harmony***	***Ours***
adult	82.5	76.7	84.16	81.9	**84.95**
mushroom	99.5	98.8	99.67	99.94	**99.98**
average	91	87.85	91.92	90.92	**92.46**

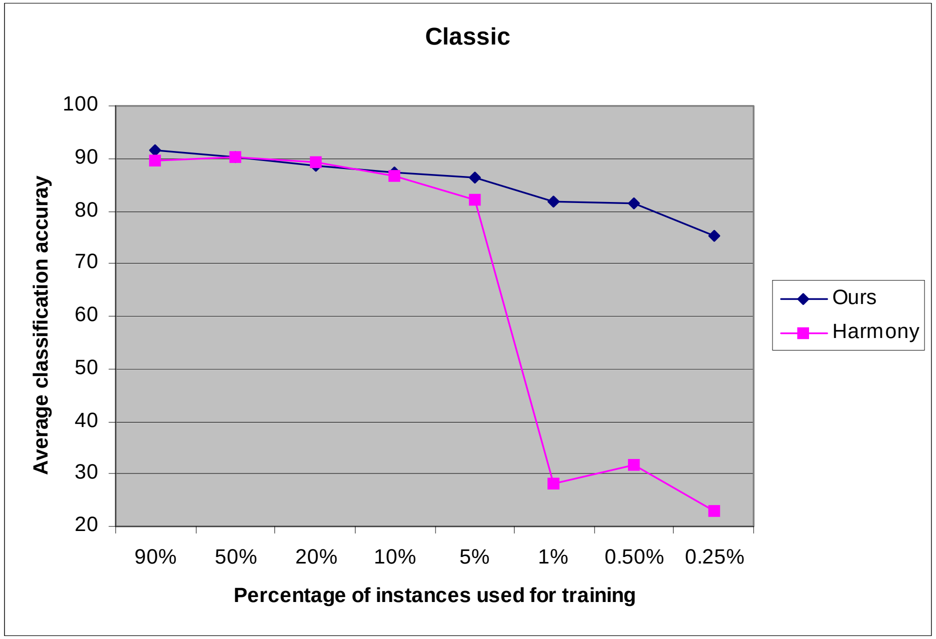
Classification Accuracies on Sports with Various Parameter Values

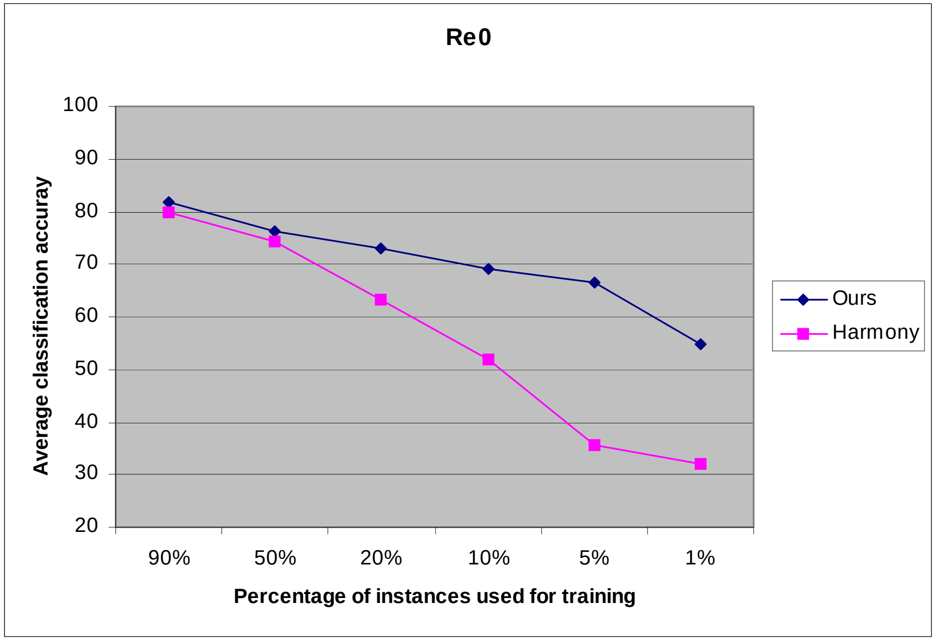
Harmony (Min support)			
75	100	125	150
94.2	94.9	94.3	94.1

SVM (C)			
2	1	0.5	0.25
95.79	95.79	95.76	95.72

Ours (min_supp)			
5	10	20	30
96.4	96.24	96.12	95.98

Classification Accuracies on Classic and Re0 with Increasingly Sparser Training Data





Outline

- Motivation
- Instance-driven Pattern Mining
- IDHC: A More Flexible Pattern-based Hierarchical Clustering Algorithm
- CPHC: Semi-supervised Classification by Pattern-based Hierarchical Clustering
- **Conclusions**

Conclusions

- Pattern mining
  - Interestingness measures outperform frequency-based measures
  - Instance-driven pattern mining more stable than global pattern mining
    - Local thresholds more robust than global thresholds
- Pattern-based hierarchical clustering
  - Instance-driven approach more stable than global approach

Conclusions - continued

- Pattern-based hierarchical clustering
  - Use instance-to-cluster pointers to avoid global refinement
  - Tight coupling between node heights and pattern lengths unnecessary
- Classification
  - Relying on training data alone may result in suboptimal classification results, specially with sparse training data

Conclusions - continued

- Classification
  - Using a pattern-based cluster hierarchy as a direct mean for semi-supervised classification
    - No need to know the number of clusters in advance
    - No extra step of training on an expanded training set
    - Exploits pattern lengths
    - May improve classification of isolated test instances

Questions?