Sequence Clustering

CS 145

Fall 2015

ApproxMAP

- Sequential Pattern Mining
- Support Framework
- Multiple Alignment Framework
- Evaluation
- Conclusion

Inherent Problems

- **Exact match**
  - **A pattern gets support from a sequence in the database if and only if the pattern is exactly contained in the sequence**
  - **Often may not find general long patterns in the database**
  - **For example, many customers may share similar buying habits, but few of them follow an exactly same pattern**
- **Mines complete set: Too many trivial patterns**
  - **Given long sequences with noise**
    - **too expensive and too many patterns**
  - **Finding max / closed sequential patterns is non-trivial**
    - **In noisy environment, still too many max/close patterns**

⇒ Not Summarizing Trend

Multiple Alignment

- **line up the sequences to detect the trend**
  - **Find common patterns among strings**
  - **DNA / bio sequences**

P	A	T	T	T	E	R	N
P	A	0	0	T	E	R	M
P	0	0	T	T	0	R	N
O	A	0	T	T	E	R	B
P	0	S	Y	Y	R	T	N
P	A	0	T	T	E	R	N

Edit Distance

- ✓ Pairwise Score = edit distance=dist(S_1,S_2)
  - Minimum # of ops required to change S_1 to S_2
  - Ops = INDEL(a) and/or REPLACE(a,b)

P	A	T	T	T	E	R	N
P	A	()	()	T	E	R	M
		INDEL	INDEL				REPL

- **Multiple Alignment Score**
  - **$\sum PS(seq_i, seq_j)$ ($\forall$ 1 ≤ i ≤ N and 1≤ j ≤ N)**
  - **Optimal alignment : minimum score**

Weighted Sequence

- **Weighted Sequence : profile**
  - **Compress a set of aligned sequences into one sequence**

seq_1	(A)		(B)	(DE)	
seq_2	(AE)	(H)	(BC)	(E)	
seq_3	(A)		(BCG)	(D)	
Weighted Sequence	(A:3,E:1):3	(H:1):1	(B:3,C:2, G:1):3	(D:2, E:2):3	3

Consensus Sequence

- **strength(i, j) = # of occurrences of item i in position j / total # of sequences**
- **Consensus itemset (j)**
  - **$\{ i_a \mid \forall i_a \in (I \cup ()) \ \&\ \text{strength}(i_a, j) \geq min\_strength \}$**
- **Consensus sequence : min_strength=2**
  - **concatenation of the consensus itemsets for all positions excluding any null consensus itemsets**

seq_1	(A)		(B)	(DE)	
seq_2	(AE)	(H)	(BC)	(E)	
seq_3	(A)		(BCG)	(D)	
Weighted Sequence	(A:3,E:1):3	(H:1):1	(B:3,C:2, G:1):3	(D:2, E:2):3	3
Consensus Sequence	(A)		(BC)	(DE)	

Multiple Alignment Pattern Mining

- **Given**
  - N sequences of sets,
  - Op costs (INDEL & REPLACE) for itemsets, and
  - Strength threshold for consensus sequences
    - can specify different levels for each partition
- **To**
  - (1) partition the N sequences into K sets of sequences such that the sum of the K multiple alignment scores is minimum, and
  - (2) find the optimal multiple alignment for each partition, and
  - (3) find the pattern consensus sequence and the variation consensus sequence for each partition

ApproxMAP
(Approximate Multiple Alignment Pattern mining)

- **Exact solution : Too expensive!**
- **Approximation Method**
  - **Group : $O(kN) + O(N^2L^2I)$**
    - **partition by Clustering (k-NN)**
    - **distance metric**
  - **Compress : $O(nL^2)$**
    - **multiple alignment (greedy)**
  - **Summarize : $O(1)$**
    - **Pattern and Variation Consensus Sequence**
  - **Time Complexity : $O(N^2L^2I)$**

Multiple Alignment : Weighted Sequence

seq_3	(A)		(B)	(DE)		
seq_2	(AE)	(H)	(B)	(D)		
WS_1	(A:2,E:1):2	(H:1):1	(B:2):2	(D:2,E:1):2		2
seq_4	(A)		(BCG)	(D)		
WS_2	(A:3,E:1):3	(H:1):1	(B:3,C:1,G:1):3	(D:3,E:1):3		3

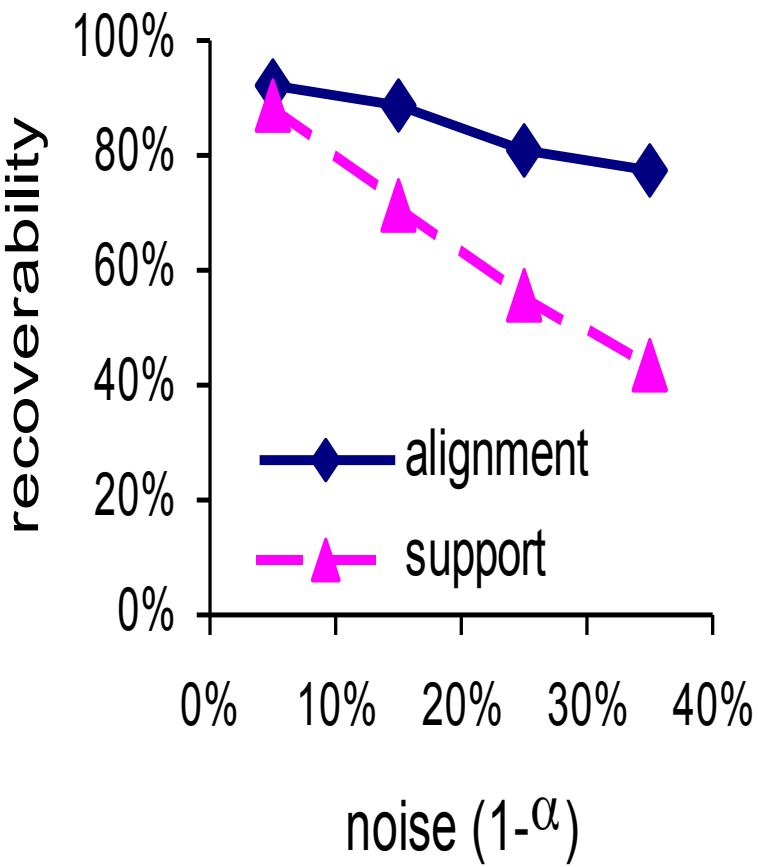
Evaluation Method: Criteria & Datasets

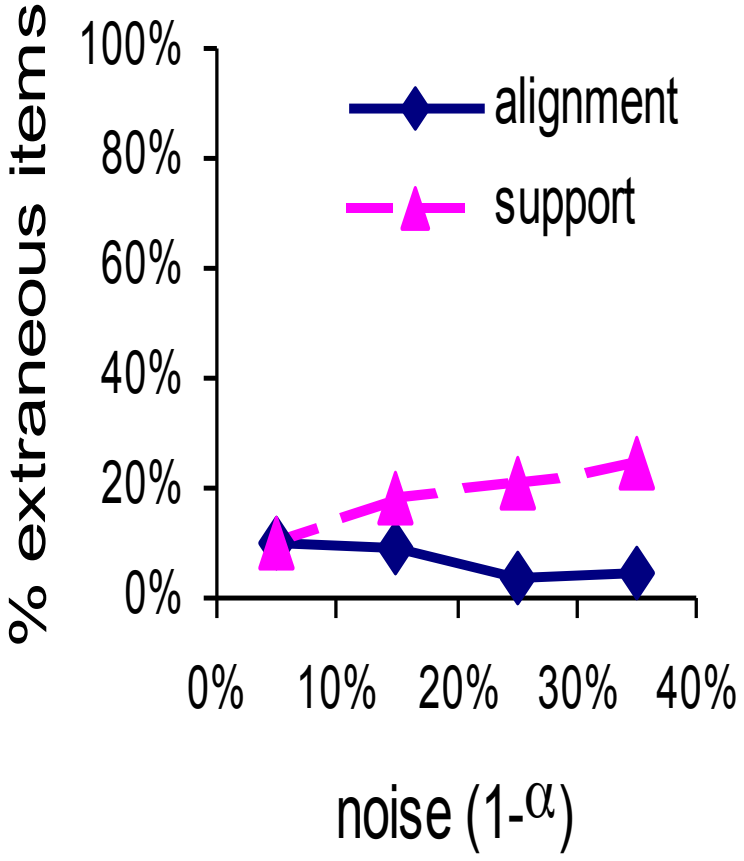
- **Criteria**
  - **Recoverability : max patterns**
    - **degree of the underlying patterns in DB detected**
    - **$\sum E(F_B) * [\max_{\text{res pat } B}(|B \otimes P|) / E(L_B)]$**
    - **Cutoff so that $0 \leq R \leq 1$**
  - **# of spurious patterns**
  - **# of redundant patterns**
  - **Degree of extraneous items in the patterns**
    - **total # of extraneous items in P / total # of items in P**
- **Datasets**
  - **Random data : Independence between and across itemsets**
  - **Patterned data : IBM synthetic data (Agrawal and Srikant)**
  - **Robustness w.r.t. noise : alpha (Yang – SIGMOD 2002)**
  - **Robustness w.r.t. random sequences (outliers)**

Evaluation : Comparison

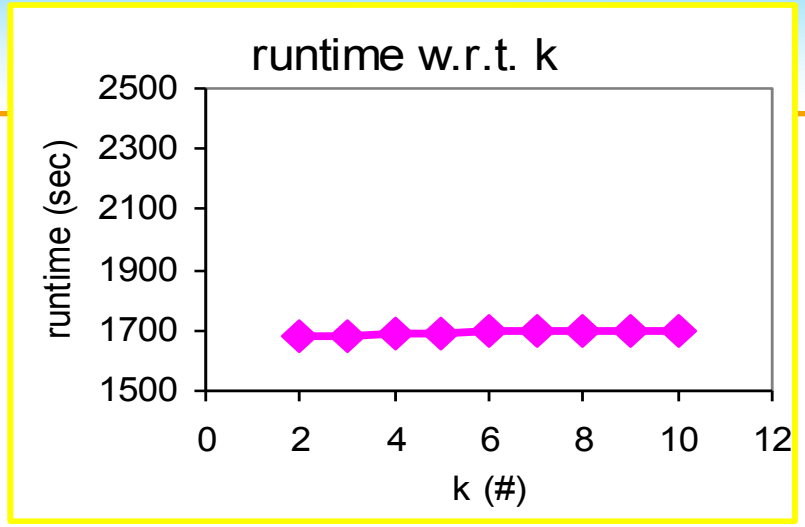
	ApproxMAP	Support Framework
Random Data	**No patterns with more than 1 item returned**	**Lots of spurious patterns**
Patterned Data **10 patterns embedded into 1000 seqs**	**k=6 & MinStrgh=30%** **Recoverability : 92.5%** **10 patterns returned** **2 redundant patterns** **0 spurious patterns** **0 extraneous items**	**MinSup=5%** **Recoverability : 91.6%** **253,924 patterns returned** **247,266 redundant patterns** **6,648 spurious patterns** **93,043=5.2% extraneous items**
Noise	**Robust**	**Not Robust** **Recoverability degrades fast**
Outliers	**Robust**	**Robust**

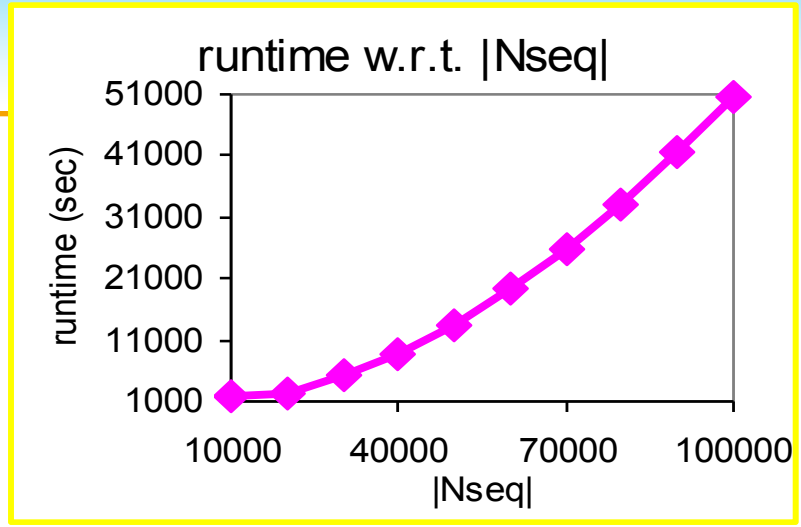
Robustness w.r.t. noise

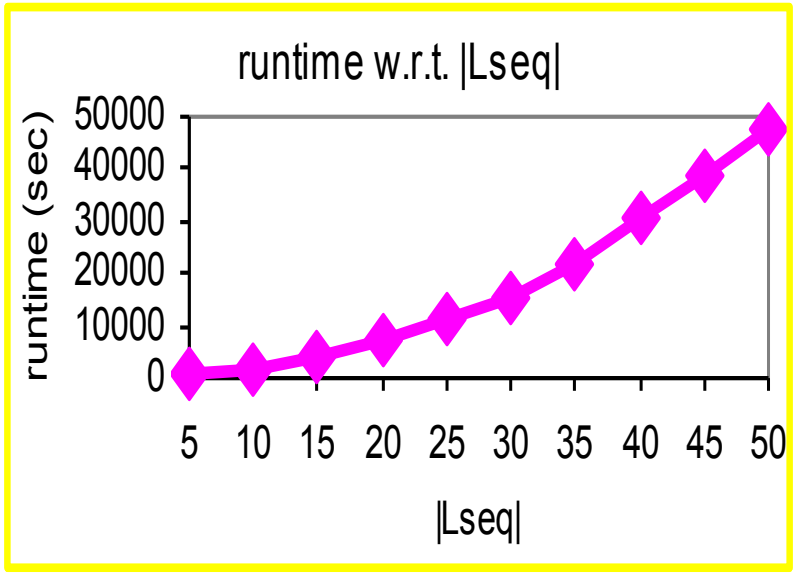


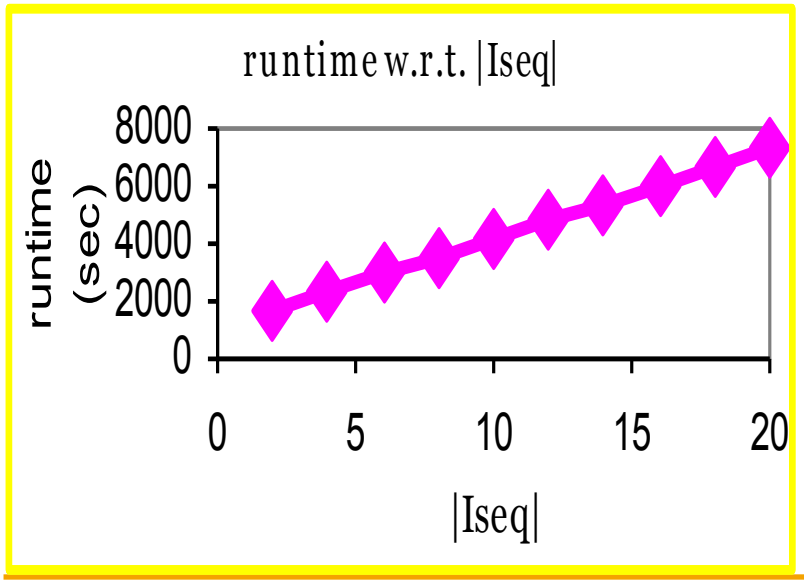


Results : Scalability









Evaluation : Real data

- **Successfully applied ApproxMAP to sequence of monthly social welfare services given to clients in North Carolina**
- **Found interpretable and useful patterns that revealed information from the data**

Conclusion : why does it work well?

- **Robust on random & weak patterned noise**
  - **Noises can almost never be aligned to generate patterns, so they are ignored**
  - **If some alignment is possible, the pattern is detected**
- **Very good at organizing sequences**
  - **when there are "enough" sequences with a certain pattern, they are clustered & aligned**
  - **When aligning, we start with the sequences with the least noise and add on those with progressively more noise**
  - **This builds a center of mass to which those sequences with lots of noise can attach to**
- **Long sequence data that are not random have unique signatures**

Conclusion

- **Works very well with market basket data**
  - **High dimensional**
  - **Sparse**
  - **Massive outliers**
- **Scales reasonably well**
  - **Scales very well w.r.t # of patterns**
  - **k : scales very well = O(1)**
  - **DB : scales reasonably well=$O(N^2 L^2 I)$**

CLUSEQ

- **The primary structures of many biological (macro)molecules are "letter" sequences despite their 3D structures.**
  - **Protein has 20 amino acids.**
  - **DNA has an alphabet of four bases {A, T, G, C}**
  - **RNA has an alphabet {A, U, G, C}**
- **Text document**
- **Transaction logs**
- **Signal streams**
- **Structural similarities at the sequence level often suggest a high likelihood of being functionally/ semantically related.**

Problem Statement

- **Clustering based on structural characteristics can serve as a powerful tool to discriminate sequences belonging to different functional categories.**
  - **The goal is to create a grouping of sequences such that sequences in each group have similar features.**
  - **The result can potentially reveal unknown structural and functional categories that may lead to a better understanding of the nature.**
- **Challenge: how to measure the structural similarity?**

Measure of Similarity

- **Edit distance:**
  - computationally inefficient
  - only captures the optimal global alignment but ignore many other local alignments that often represent important features shared by the pair of sequences.
- ***q*-gram based approach:**
  - ignores sequential relationship (e.g., ordering, correlation, dependency, etc.) among *q*-grams
- **Hidden Markov model:**
  - capture some low order correlations and statistics
  - vulnerable to noise and erroneous parameter setting
  - computationally inefficient

Measure of Similarity

- **Probabilistic Suffix Tree**
  - Effective in capturing significant structural features
  - Easy to compute and incrementally maintain
- **Sparse Markov Transducer**
  - Allows wild cards

Model of CLUSEQ

- **CLUSEQ: exploring significant patterns of sequence formation.**
  - **Sequences belonging to one group/cluster may subsume to the same probability distribution of symbols (conditioning on the preceding segment of a certain length), while different groups/clusters may follow different underlying probability distributions.**
  - **By extracting and maintaining significant patterns characterizing (potential) sequence clusters, one can easily determine whether a sequence should belong to a cluster by calculating the likelihood of (re)producing the sequence under the probability distribution that characterizes the cluster.**

Model of CLUSEQ

Sequence: $\sigma = s_1 s_2 \ldots s_l$

Cluster S: $P_S(\sigma) = P_S(s_1) \times P_S(s_2 \mid s_1) \times \cdots \times P_S(s_l \mid s_1 \cdots s_{l-1})$

$$= \prod_{i=1}^{l} P_S(s_i \mid s_1 \cdots s_{i-1})$$

Random process: $P^r(\sigma) = P^r(s_1) \times P^r(s_2) \times \cdots \times P^r(s_l)$

$$= \prod_{i=1}^{l} P^r(s_i)$$

If $P_S(\sigma)$ is high, we may consider σ a member of S

If $P_S(\sigma) >> P^r(\sigma)$, we may consider σ a member of S

Model of CLUSEQ

- **Similarity between σ and S**

$$sim_S(\sigma) = \frac{P_S(\sigma)}{P^r(\sigma)} = \frac{\prod_{i=1}^{l} P_S(s_i \mid s_1 \ldots s_{i-1})}{\prod_{i=1}^{l} p(s_i)} = \prod_{i=1}^{l} \left(\frac{P_S(s_i \mid s_1 \ldots s_{i-1})}{p(s_i)} \right)$$

- **Noise may be present.**
- **Different portions of a (long) sequence may subsume to different conditional probability distributions.**

$$SIM_S(\sigma) = \max_{1 \le i \le j \le l} sim_S(s_i \ldots s_j)$$

Model of CLUSEQ

- **Give a sequence $\sigma = s_1 s_2 \ldots s_l$ and a cluster S, a dynamic programming method can be used to calculate the similarity $SIM_S(\sigma)$. Via a single scan of σ. Let**

$$X_i = \frac{P_S(s_i \mid s_1 \ldots s_{i-1})}{p(s_i)}$$

$$Y_i = \max_{1 \le j \le i} sim_S(s_j \ldots s_i)$$

$$Z_i = \max_{1 \le i1 \le i\,2 \le i} sim_S(s_{i1} \ldots s_{i2})$$

- **Intuitively, X_i, Y_i, and Z_i can be viewed as the similarity contributed by the symbol on the ith position of σ (i.e., s_i), the maximum similarity possessed by any segment ending at the ith position, and the maximum similarity possessed by any segment ending prior to or on the ith position, respectively.**

Model of CLUSEQ

- **Then, $SIM_S(\sigma) = Z_l$, which can be obtained by**

$$Y_i = \max\{Y_{i-1} \times X_i, X_i\}$$
$$Z_i = \max\{Z_{i-1}, Y_i\}$$
$$Y_1 = Z_1 = X_1$$

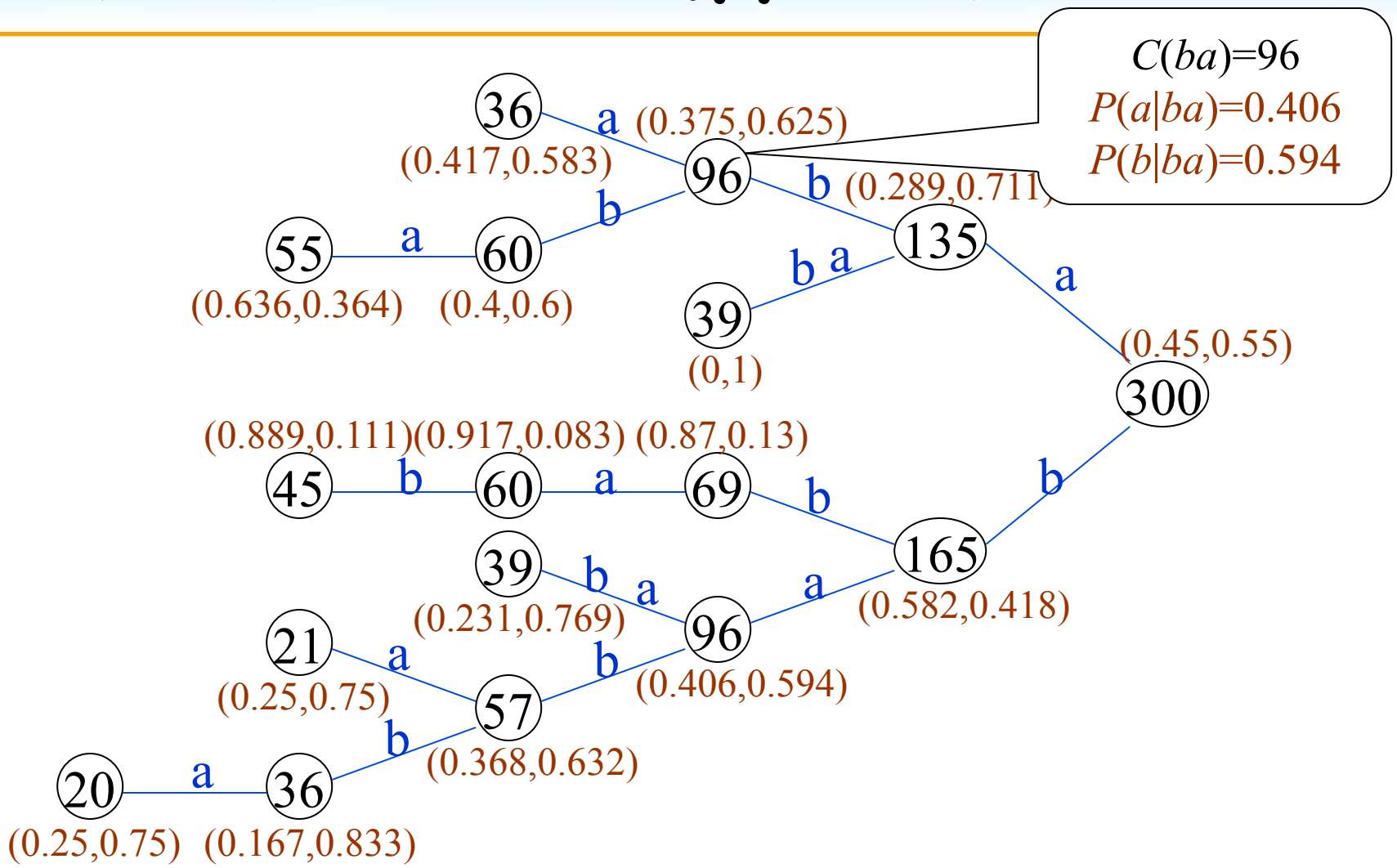
- **For example, $SIM_S(bbaa) = 2.10$ if $p(a) = 0.6$ and $p(b) = 0.4$.**

Sequence	*b*	*b*	*a*	*a*
$P_S(s_i \mid s_1 \ldots s_{i-1})$	0.55	0.418	0.87	0.406
X_i	1.38	1.05	1.45	0.677
Y_i	1.38	1.45	2.10	1.42
Z_i	1.38	1.45	2.10	**2.10**

Probabilistic Suffix Tree

- a compact representation to organize the derived CPD for a cluster
- built on the *reversed* sequences
- Each node corresponds to a segment, σ, and is associated with a counter $C(\sigma)$ and a probability vector $P(s_i|\ \sigma)$.

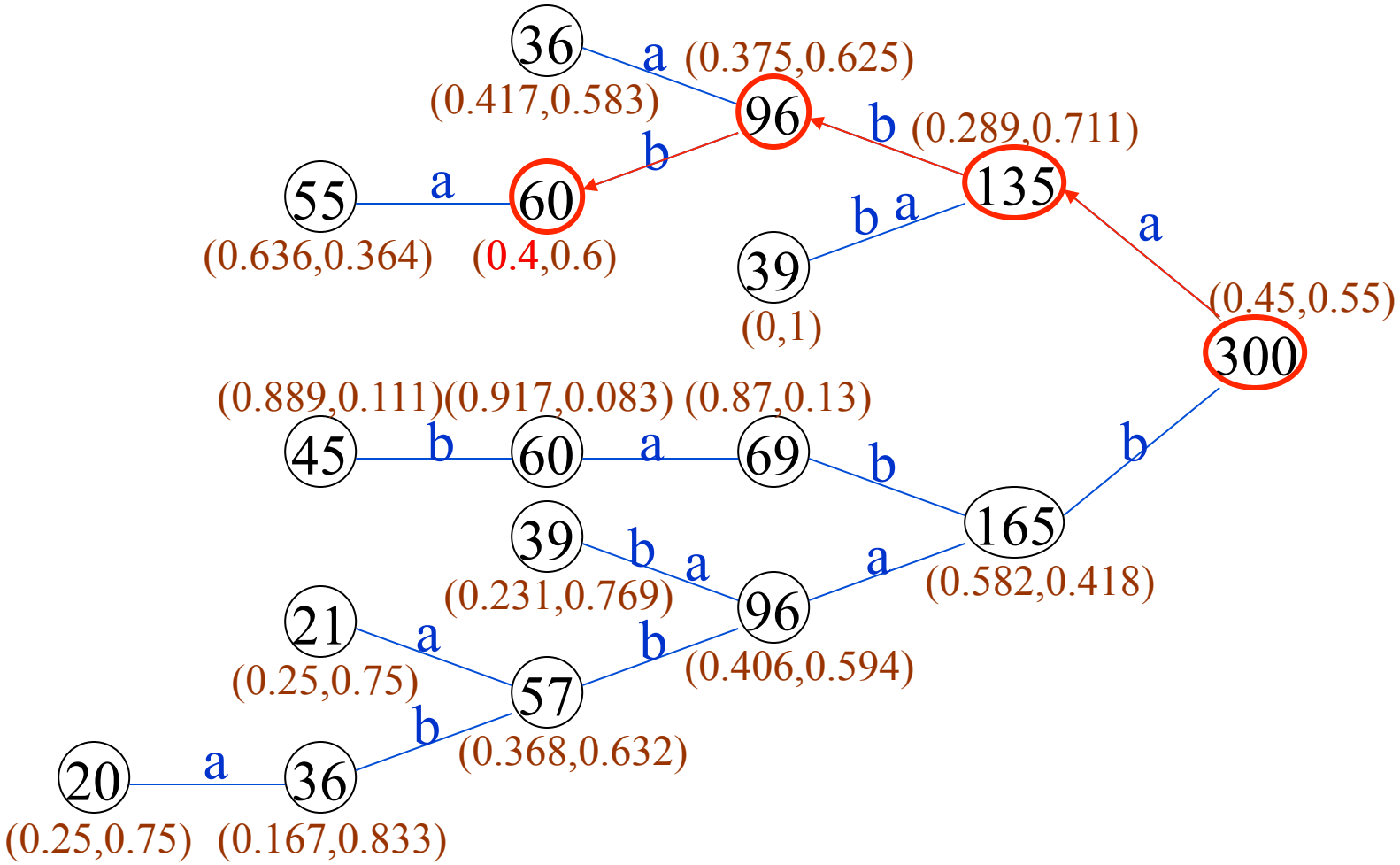
Probabilistic Suffix Tree



Model of CLUSEQ

- Retrieval of a CPD entry $P(s_i|s_1 \ldots s_{i-1})$
- The *longest suffix* $s_j \ldots s_{i-1}$
  - can be located by traversing from the root along the path "$\rightarrow s_{i-1} \rightarrow \ldots \rightarrow s_2 \rightarrow s_1$" until we reach either the node labeled with $s_1 \ldots s_i$ or a node where no further advance can be made.
  - takes $O(\min\{i, h\})$ where h is the height of the tree.
- Example: $P(a|bbba)$

$$P(a|bbba) \approx P(a|bba) = 0.4$$

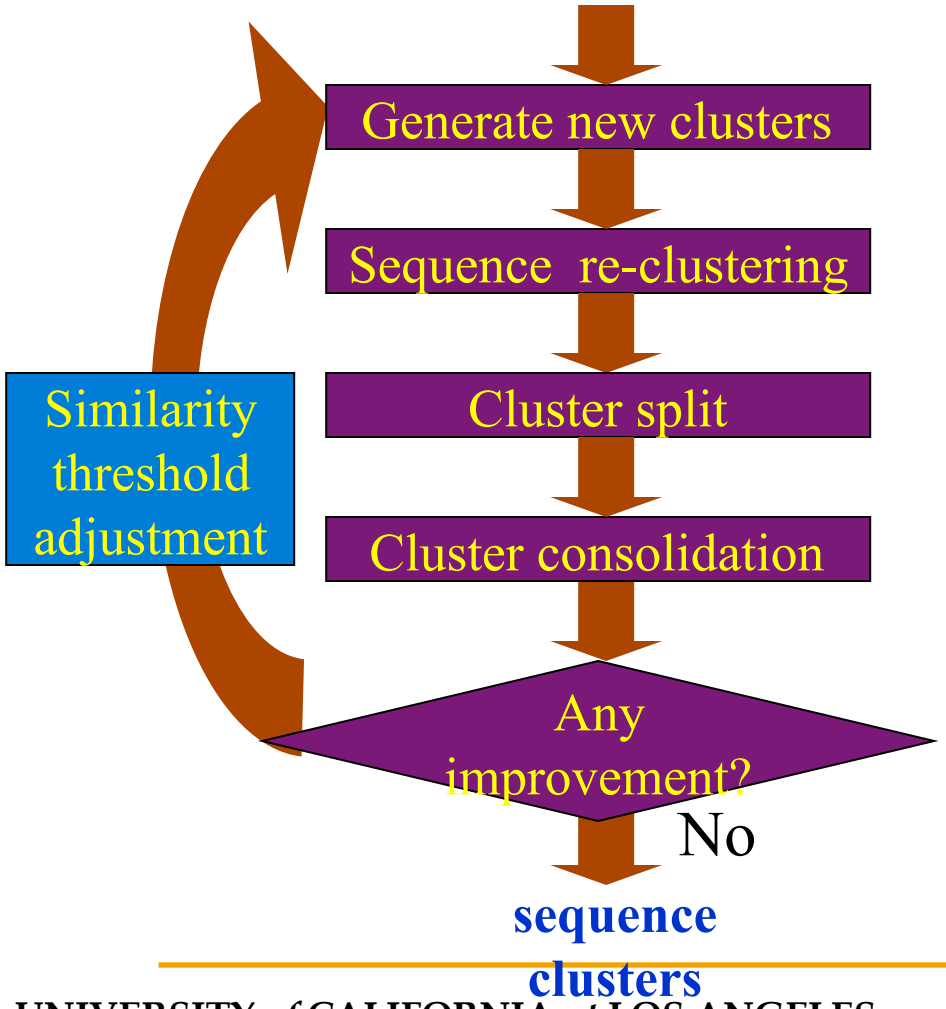


CLUSEQ

- **Sequence Cluster: a set of sequences *S* is a** sequence cluster **if, for each sequence σ in *S*, the similarity $SIM_S(\sigma)$ between σ and *S* is greater than or equal to some similarity threshold *t*.**
- Objective: **automatically group a set of sequences into a set of *possibly overlapping* clusters.**

Algorithm of CLUSEQ

- **An iterative process**
  - **Each cluster is represented by a probabilistic suffix tree.**
  - **The optimal number of clusters and the number of outliers allowed can be adapted by CLUSEQ automatically**
    - **new cluster generation, cluster split, and cluster consolidation**
    - **adjustment of similarity threshold**



New Cluster Generation

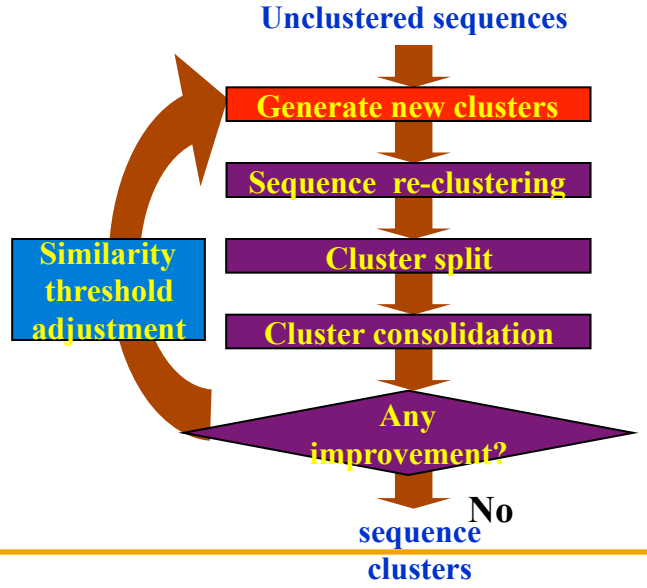
- **New clusters are generated from *un-clustered* sequences at the beginning of each iteration.**
  - ***k'*** × ***f*** **new clusters**

number of consolidated clusters

$$f = \frac{\max\{k'_n - k'_c, 0\}}{k'_n}$$

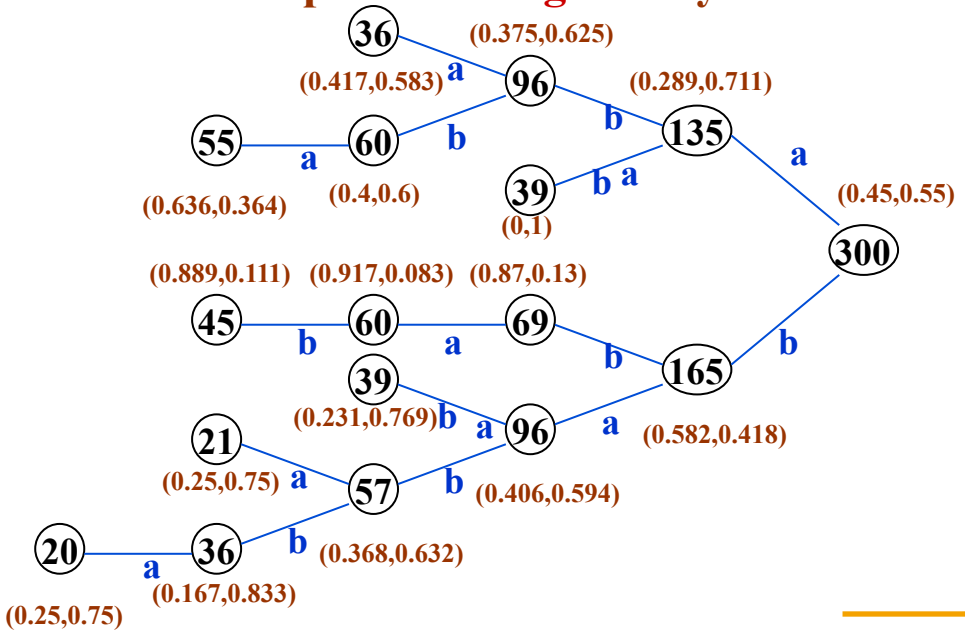
number of clusters

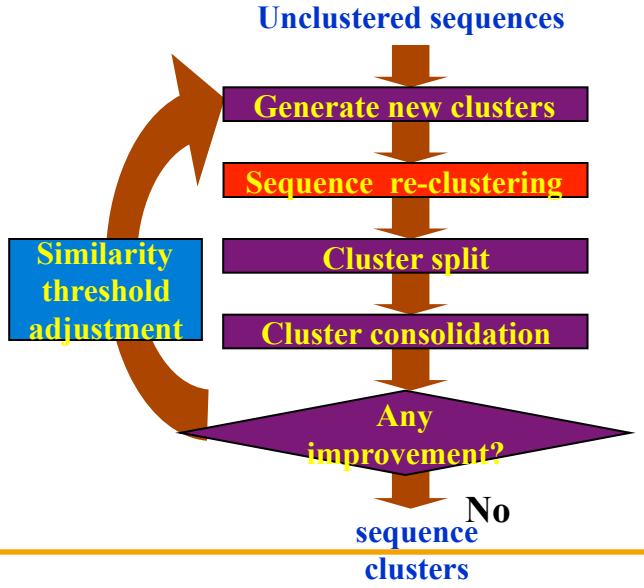
number of new clusters generated at the previous iteration



Sequence Re-Clustering

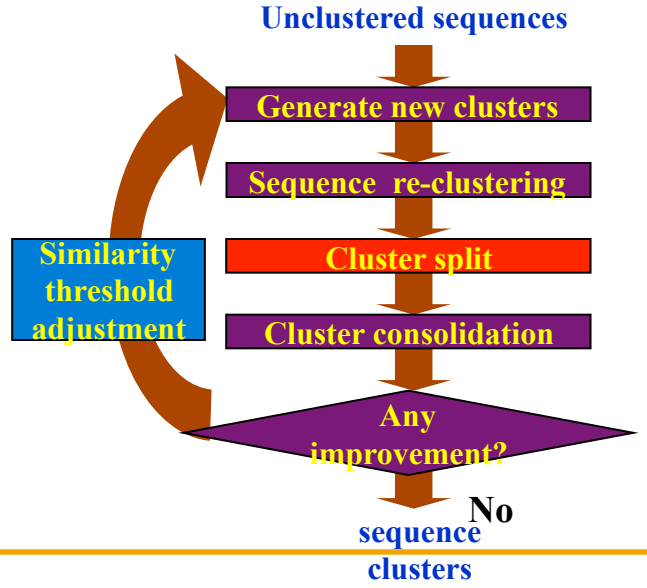
- For each (sequence, cluster) pair
  - Calculate similarity
  - PST update if necessary
    - Only similar portion is used
    - The update is weighted by the similarity value





Cluster Split

- **Check the convergence of each existing cluster**
  - **Imprecise probabilities are used for each probability entry in PST**
  - **Split non-convergent cluster**



Imprecise Probabilities

- **Imprecise probabilities uses two values (p_1, p_2) (instead of one) for a probability.**
  - **p_1 is called lower probability and p_2 is called upper probability.**
  - **The true probability lies somewhere between p_1 and p_2.**
  - **$p_2 - p_1$ is called imprecision.**

Update Imprecise Probabilities

- **Assuming the prior knowledge of a (conditional) probability is (p_1, p_2) and the occurrences in the new experiment is *a* out of *b* trials.**

$$p'_1 = \frac{a + s \times p_1}{b + s} \qquad p'_2 = \frac{a + s \times p_2}{b + s}$$

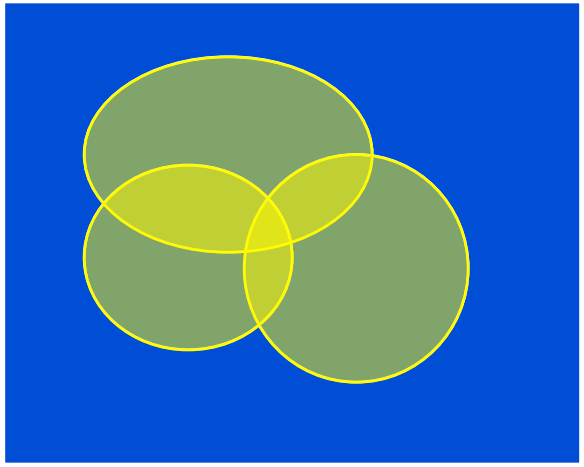
where *s* is the learning parameter which controls the weight that each experiment carries.

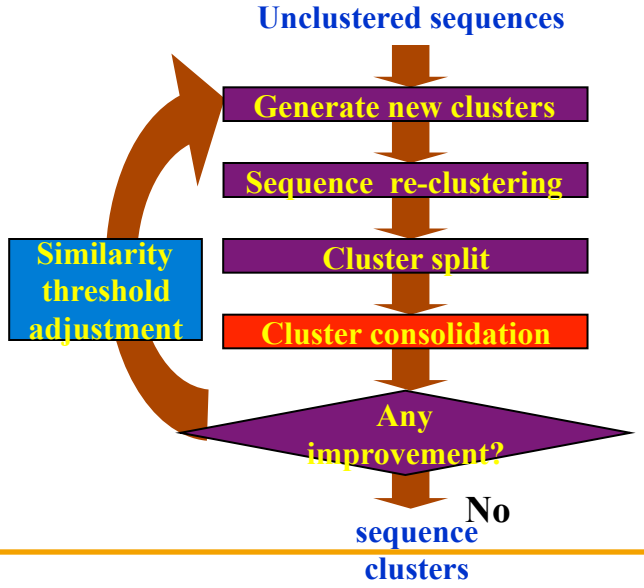
Properties

- **The following two properties are very important.**
  - **If the probability distribution stays static, then p_1 and p_2 will converge to the true probability.**
  - **If the experiment agrees with the prior assumption, the range of imprecision decreases after applying the new evidence, e.g., $p_2' - p_1' < p_2 - p_1$.**
- **The clustering process terminates when the imprecision of all significant nodes is less than a small threshold.**

Cluster Consolidation

- Starting from the smallest cluster
- Dismiss clusters that have few sequence **not covered** by other clusters

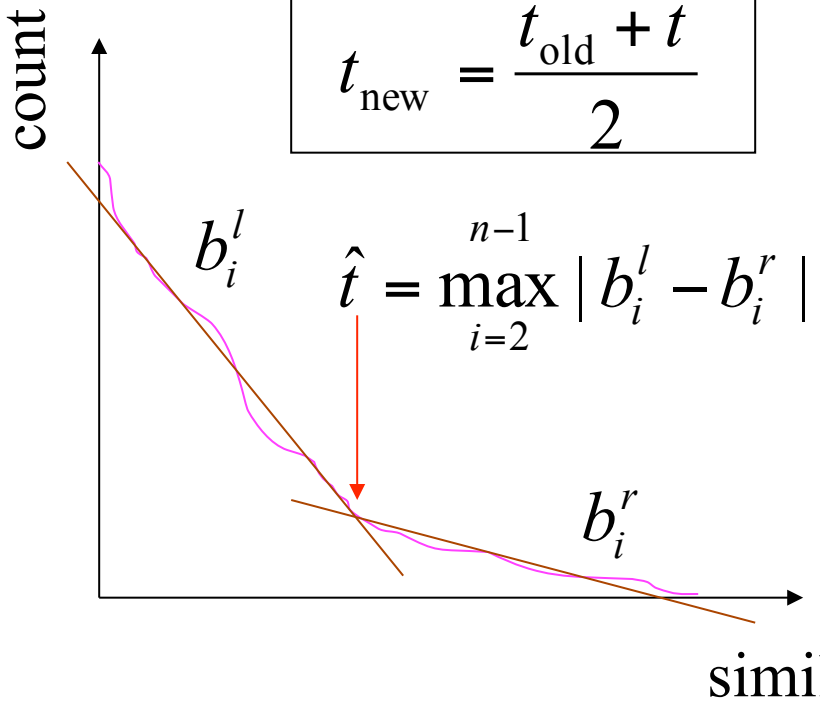


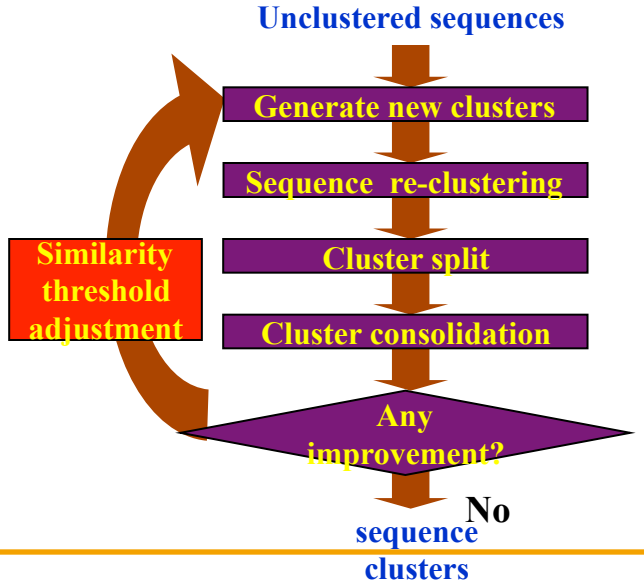


Adjustment of Similarity Threshold

- **Find the sharpest turn of the similarity distribution function**

$$t_{new} = \frac{t_{old} + \hat{t}}{2}$$





Algorithm of CLUSEQ

- **Implementation issues**
  - **Limited memory space**
    - **Prune the node with smallest count first.**
    - **Prune the node with longest label first.**
    - **Prune the node with expected probability vector first.**
  - **Probability smoothing**
    - **Eliminates zero empirical probability**
  - **Other considerations**
    - **Background probabilities**
    - **A priori knowledge**
    - **Other structural features**

Experimental Study

- **We have experimented with a protein database of 8000 proteins from 30 families from SWISS-PROT database.**

Model	CLUSEQ	Edit Distance	Edit Distance with Block Operations	Hidden Markov Model	Q-gram
Accuracy	92%	23%	90%	91%	75%
Response Time (sec)	144	487	13754	3117	132

Experimental Study

Synthetic data

Initial *t*	1.05	1.5	2	3
Final *t*	1.99	2.01	2	1.99
Response time	8011	7556	6754	7234
precision	81.3%	83.1%	83.4%	81.9%
recall	82.1%	82.8%	83.6%	82.7%

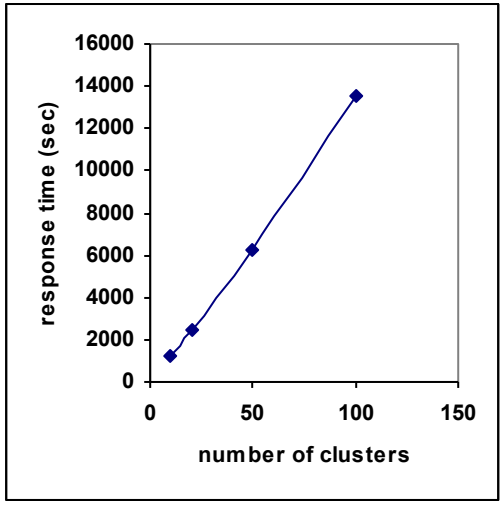
Experimental Study

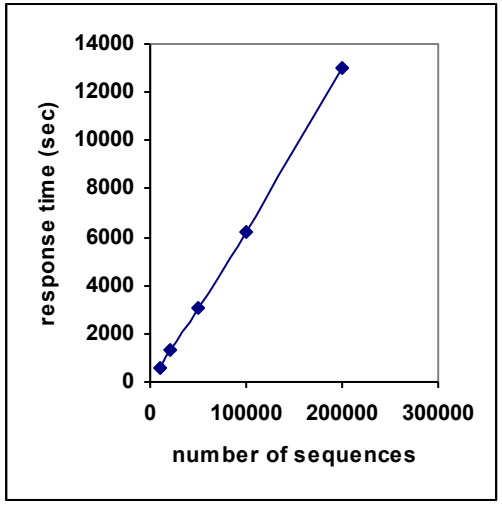
Synthetic data

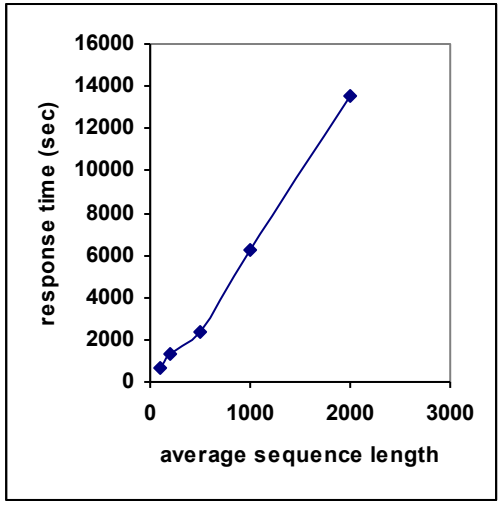
Initial cluster number	1	20	100	200
Final cluster number	102	99	101	102
Response time	10112	9023	6754	8976
precision	81.3%	82.1%	82.6%	81%
recall	81.6%	82%	83.4%	81.7%

Experimental Study

- **CLUSEQ has linear scalability with respect to the number of clusters, number of sequences, and sequence length.**







Remarks

- **Similarity measure**
  - **Powerful in capturing high order statistics and dependencies**
  - **Efficient in computation — linear complexity**
  - **Robust to noise**
- **Clustering algorithm**
  - **High accuracy**
  - **High adaptability**
  - **High scalability**
  - **High reliability**

References

- CLUSEQ: efficient and effective sequence clustering, *Proceedings of the 19th IEEE International Conference on Data Engineering (ICDE)*, 2003.
- A frame work towards efficient and effective protein clustering, *Proceedings of the 1st IEEE CSB*, 2002.