



# Efficient Fusion of a Set of Attributed Graphs

## EM-MCMC Approach

Michael Stearns, Alex Nikolaev

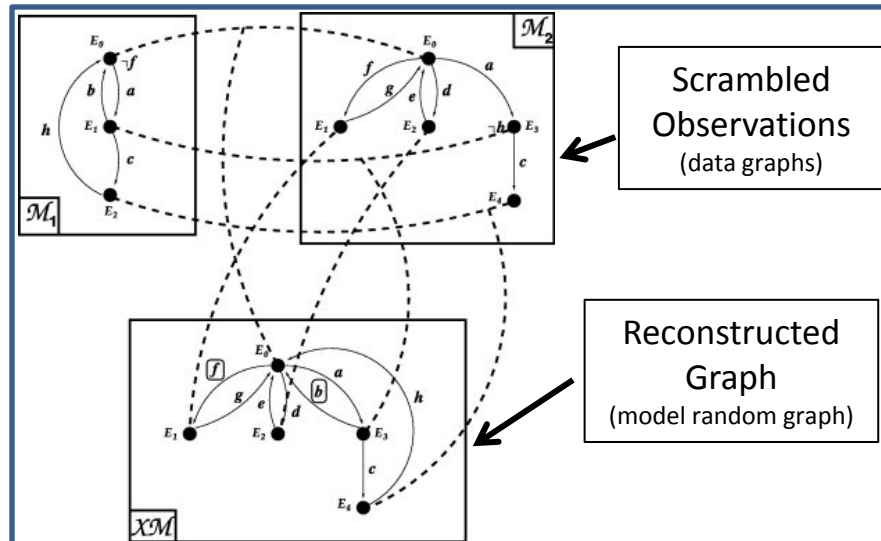


### Objectives

- Develop an efficient, scalable method for attributed graph association
- Example: reconstruct an event based on scrambled and possibly conflicting reports

### Scientific/Technical Approach

- Find a model random graph most likely to have generated the observed data graphs
- Association variables are latent
- Expectation-Maximization to minimize the model graph entropy
- Markov Chain Monte Carlo sampling to evaluate the fit at every iteration



### Accomplishments

- Concept validation performed (error-free data)
- Testing underway with error-embedded data
- Algorithm provably optimal
- Linear runtime in the number of data graphs

### Challenges

- Algorithm parameter adjustment (adaptive?)
- Working with partially observed data (graph extension latent variables?)

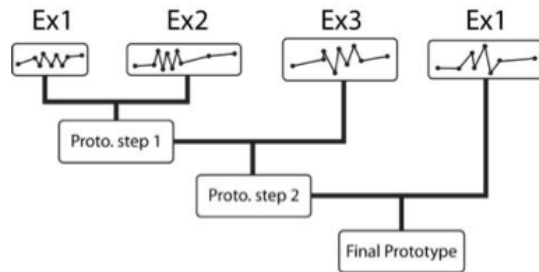


# Efficient Fusion of a Set of Attributed Graphs

## Graph Synthesis Methods Overview

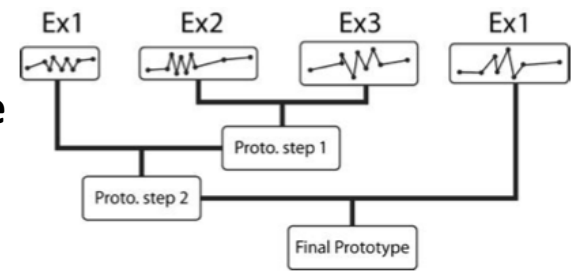


### Incremental Synthesis



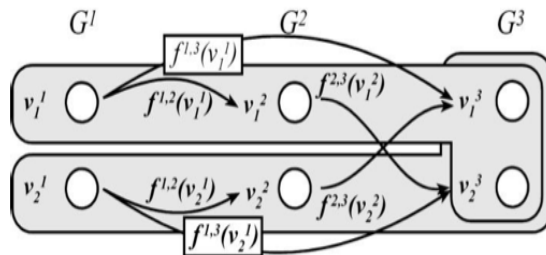
- Match data graphs as they are sequentially introduced
- Simultaneous learning and recognition
- Order Matters

### Agglomerative Synthesis



- Iteratively match pairs with smallest relative incremental distances
- Order-independent
- Early errors affect the overall result

### Consistent Multiple Isomorphism



- Decouple common labeling from synthesis: match pair-wise under consistency rules
- Too much attention to local knowledge
- Resolving rules is computationally hard

### Direct Synthesis Methods

- Holistic common labeling without a-priori pair-wise stage
  - Genetic search for optimal label array
  - N-dimensional graduated assignment
- Global knowledge guides search
- Computationally feasible only for a small number of data graphs



# Efficient Fusion of a Set of Attributed Graphs

## Expectation Maximization (EM)



- Avoid the matching altogether!
- Find an optimal parameterization of a model graph that is most likely to have generated the observed data.
- Common labeling variables are latent: can be computed for each data graph individually after an optimal model graph is identified.
- Algorithm is provably convergent: iteratively updates the model graph parameters to maximize the combined likelihood of the data graphs.
- Challenge: the distribution of latent variables under a given parameterization is complex. Sampling can be used to estimate the likelihood function but how to do it efficiently?



# Efficient Fusion of a Set of Attributed Graphs

## Markov Chain Monte Carlo Sampling (MCMC)



- Approximate the probability distribution of latent variables by constructing a Markov chain that has this distribution as its equilibrium distribution.
- Sample quality improves iteratively.
- Rule to traverse latent space: release current solution nodes at random.
- MCMC setup variables can be adjusted to accelerate convergence.

### **Benefits EM graph synthesis with MCMC**

- Global knowledge within the set of observed graphs is the driver, theoretically providing the most accurate synthesis.
- Both EM and MCMC are shown to be convergent.
- Contribution of each data graph to the model graph update can be assessed separately: overall computational time is linear!



# Efficient Fusion of a Set of Attributed Graphs

## Observed Results (Proof of Concept)



- Generate a cluster of identical AGs, with a random initial labeling to the model graph.
- Run the algorithm to optimality ( entropy = 0 ) for various cluster sizes to determine time and number of iterations necessary to attain optimality.
- Variables
  - Number of MCMC samples = 250
  - Tolerance (t) = adaptive, adjusts to maintain the desired rate of node release in MCMC, e.g., if the node release rate falls below a stated level, the tolerance increases.

Cardinality	Cluster size	Time (sec)	Number Iterations
20	25	6.3169	157.45
20	50	11.8949	149.25
20	100	22.2518	130.75
20	200	44.1427	143.4
20	500	128.2908	157.55



# Efficient Fusion of a Set of Attributed Graphs

## Current and Future Work



- Embed error in AGs as they are created (node/edge attributes will have a certain % chance of being incorrectly reported)
- Compare against the existing synthesis approaches
  - assess accuracy as the entropy of the generated random graph prototype
  - assess efficiency as the time to complete synthesis
- Allow incomplete records – AGs with partial information of the real network.
  - expand the method with graph extension capability
- Let node and arc attributes follow non-standard distributions
  - incorporate empirical likelihood computation
- Consider random graphs with higher level dependencies as model graphs
  - First-Order Random Graphs, Second-Order Random Graphs, etc.