Attributed Graph Clustering

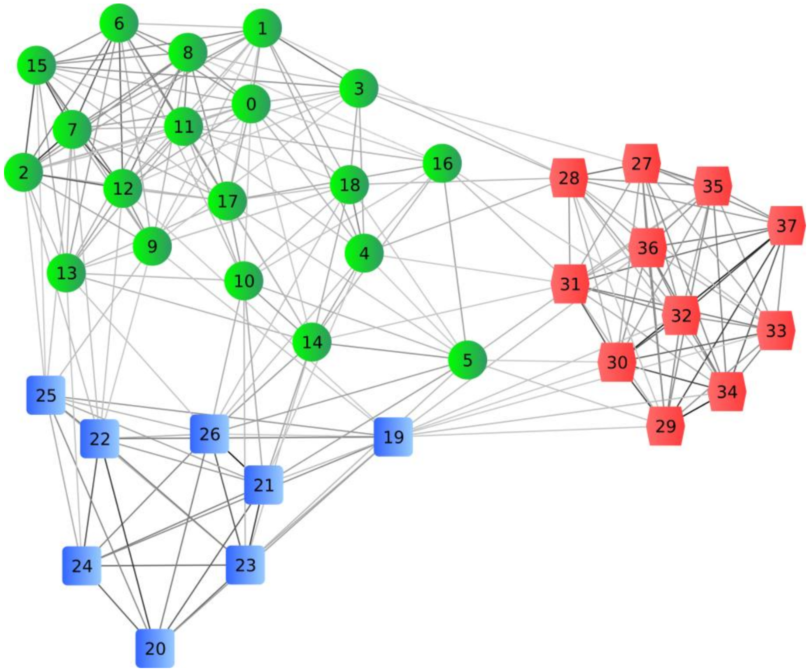
Presented by Zhao Zhou

Outline

- Motivation
- Related work
- Attributed Graph
- Distance-based Clustering Algorithm
- Model-based Clustering Algorithm
- Conclusion

Graph Clustering

- For a given set of objects, we would like to divide it into groups of similar objects.
- The similarity between objects is defined by objective functions



Real Applications

- Community Detection
  - Nodes represent members
  - Friendship relationship represent the corresponding links

- Telecommunication Networks
  - Individual phone numbers represent nodes
  - Phone calls represented as edges

- Email Analysis
  - Individuals represented as nodes
  - Emails sent represented as edges

Related work

- A number of algorithms for graph clustering proposed before
    1. The Minimum Cut Algorithm [3]
    2. Multi-way Graph Partition [4]
    3. k-medoid and k-means algorithm [5]
    4. Spectral Clustering method [6]
    5. ………

Attributed Graph

- Existing graph clustering methods
  - mainly focus on the topological structure for clustering
  - Ignore the vertex properties

- Proliferation of rich information for real objects
  - Vertices associated with a number of attributes
  - Its attributes describe the characteristics and properties
  - give rise to a new type of graphs

Applications

- Attributed graph clustering is useful in many domains
  - Service-oriented social network
  - Communication network

Service-oriented social network

- Each user can characterized by his profile
  - Interests, gender, education, etc

- Clustering users by considering both relationships and profile is more useful
  - Recommendation system
  - User-targeted online advertising

Communication network

- Each subscriber is associated with
  - Demographic information, service usage, etc

- Attributed Graph clustering on users is useful in
  - Design effective group-oriented marketing strategies

Problem Definition

- Given an attributed graph G and the number of clusters K, the clustering is to partition the vertex set V of G into K disjoint subsets $V_1,..,V_k$
  - Vertices within clusters are densely connected
  - Vertices in different clusters are sparsely connected
  - Vertices within clusters shave low diversity in their attribute values
  - Vertices in different clusters have diverse attribute values

Attributed Graph Clustering Algorithm

- The algorithms for attributed graph clustering can be categorized into two types
  - Distance-based Approaches
  - Model-based Approaches

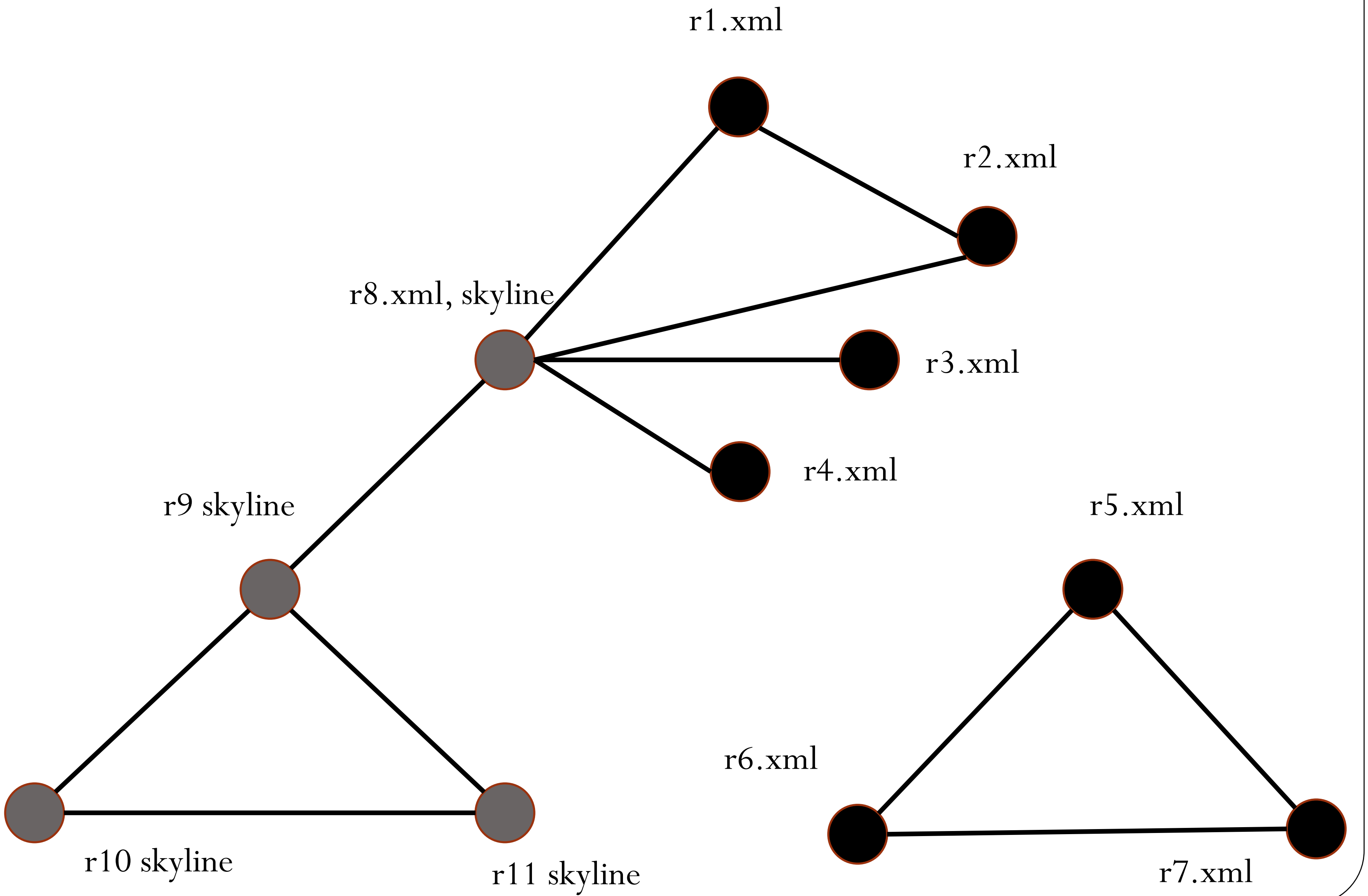
Distance-based Approaches(vldb09)

- The main idea is to design a distance measure for vertex pairs that combines both structural and attribute information

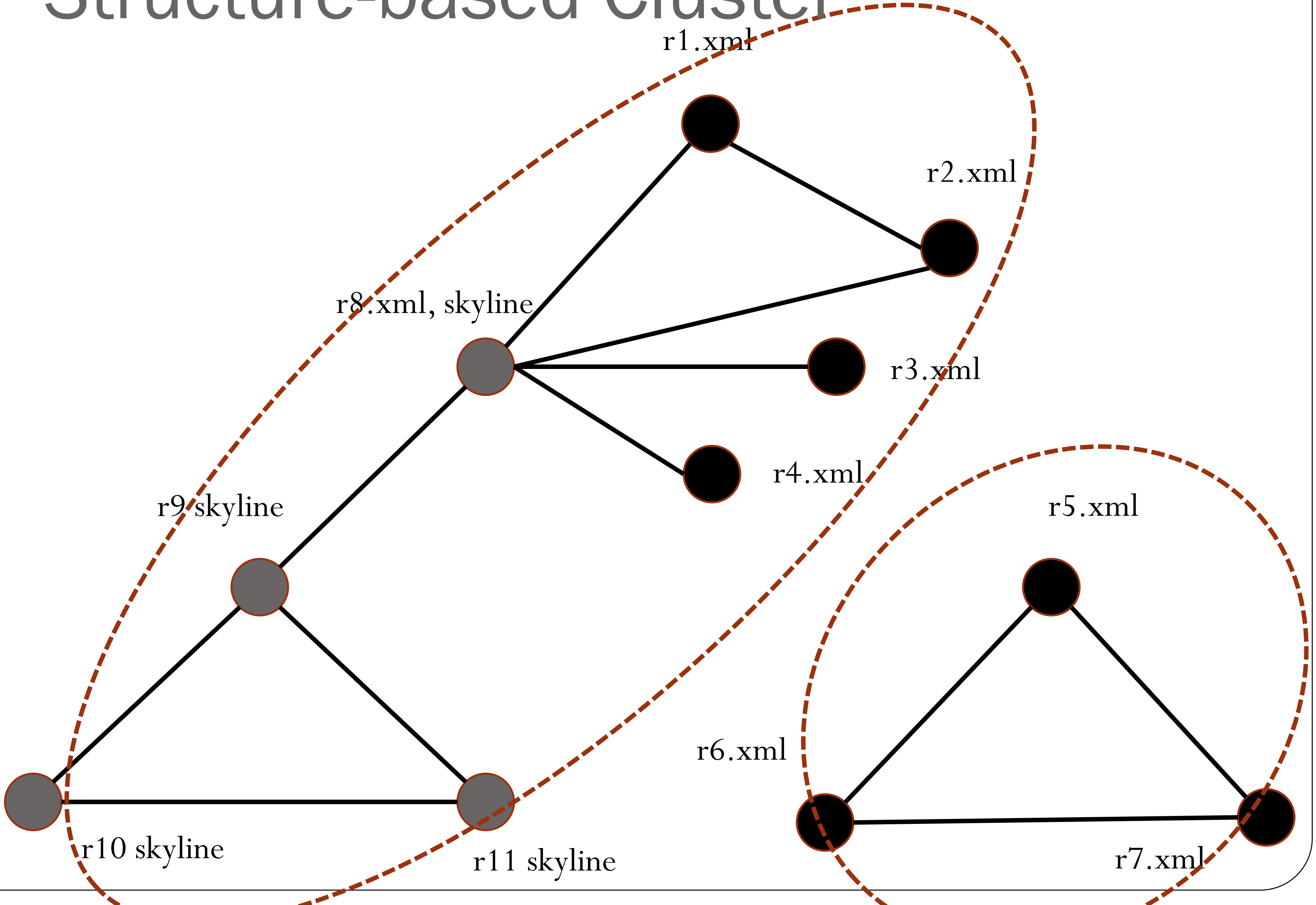
- Based on this measure, standard clustering algorithms like k-medoids are then applied to cluster the vertices

- The state-of-the-art approaches are the SA-cluster [1] and its extended version [7,8]

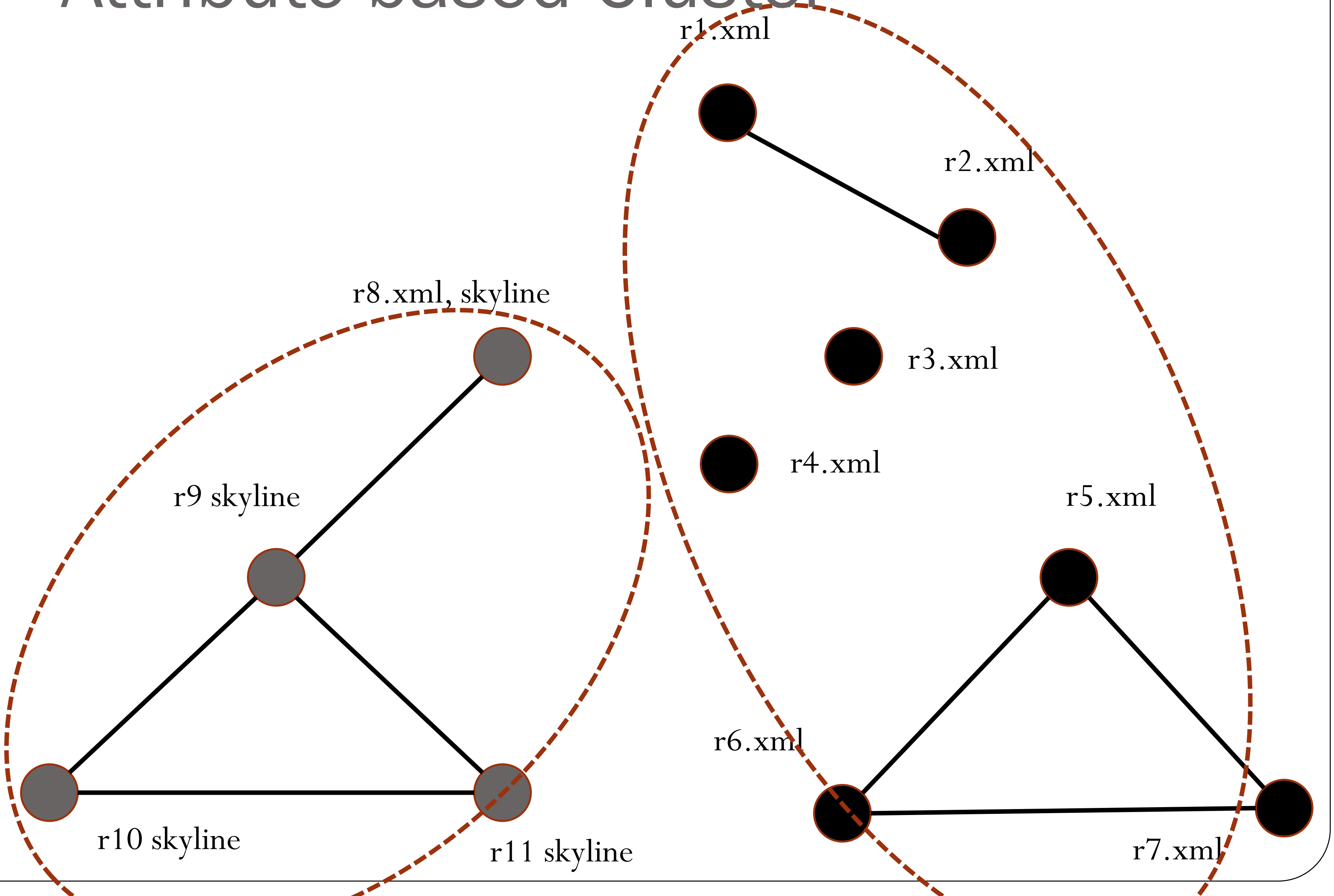
Co-author Graph



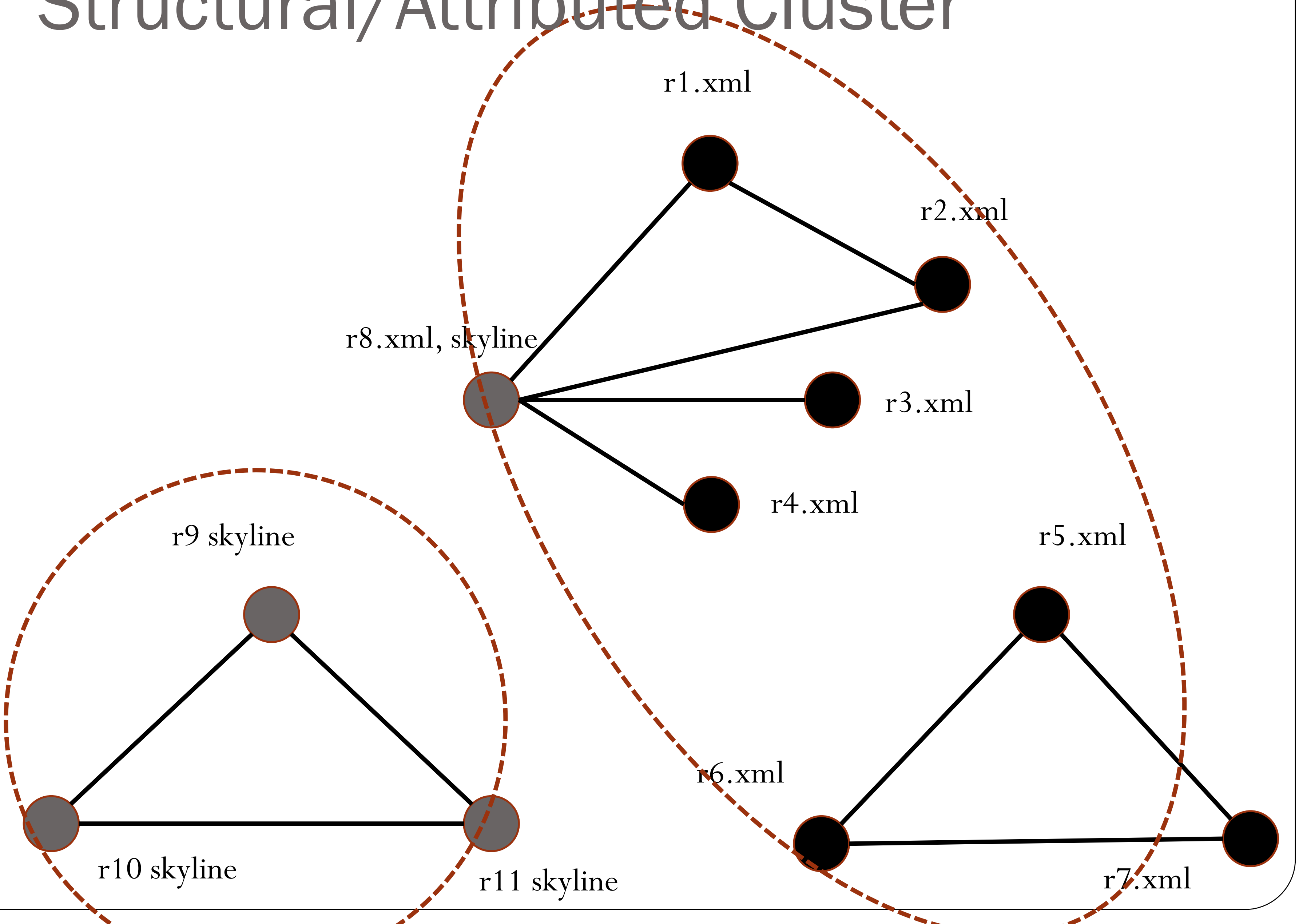
Structure-based Cluster



Attribute-based Cluster



Structural/Attributed Cluster



Basic solution

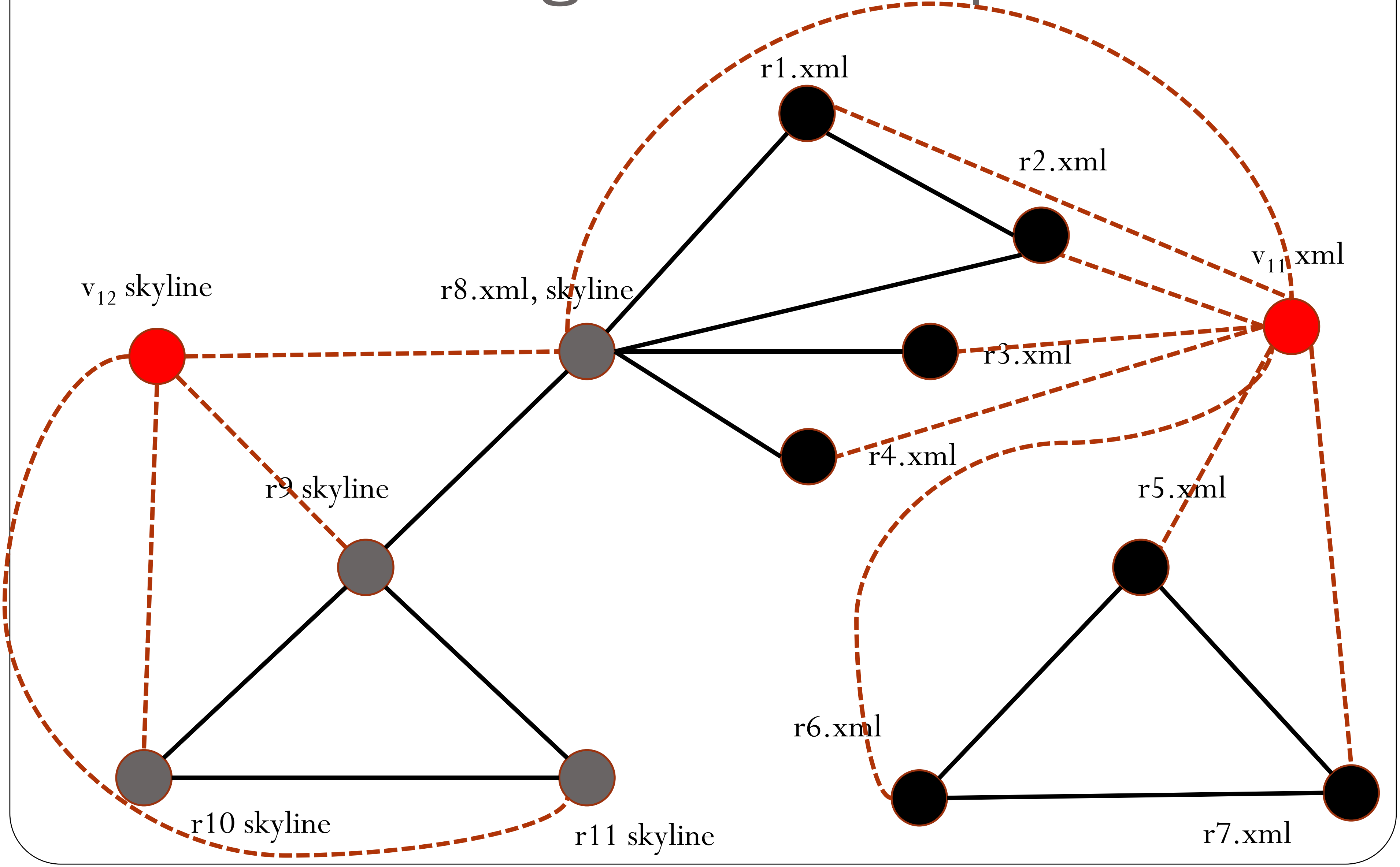
- Design a distance function between v_i and v_j
  - $d(v_i, v_j) = \alpha \cdot d_S(v_i, v_j) + \beta \cdot d_A(v_i, v_j)$
- Hard to set the parameters α and β

Attribute augmented Graph

- An attribute augmented graph is $G_a = (V \cup V_a, E \cup E_a)$
  - V_a is the set of attribute vertex, v_{ij} represents attribute I takes the jth value

  - E_a is attribute edge, (v_i, v_{ij}) means vertex vi takes a value of a_{ij} on attribute i

Attribute Augmented Graph



Transition probab

weight of structure edge

- Transition probability betw n struct

weight of attribute edge

  - $P(v_i,v_j)=\frac{w_o}{|N(v_i)|*w_0+w_1+\cdots+w_m}$
- Transition probability through structure vertex to attribute vertex
  - $P(v_i,v_j) = \frac{w_j}{|N(v_i)|*w_0+w_1+\cdots+w_m}$
- Transition probability through attribute vertex to structure vertex
  - $P(v_{ik},v_j) = \frac{1}{|N(v_{ik})|}$

Transition probability matrix

- Presentation of transition probability matrix
  - $P_A = \begin{bmatrix} P_V & A \\ B & o \end{bmatrix}$
  - P_V transition probability between vertices
  - A transition probability between vertices and attribute vertices
  - B transition probability between attribute vertices and vertices
- P_A can be incrementally computed by
  - $P_A{}^l = \begin{bmatrix} T_l & T_{l\,\_\,1}A \\ BT_{l\,\_\,1} & BT_{l\,\_\,2}A \end{bmatrix}$
  - $T_l = P_V T_{l-1} + ABT_{l-2}$

Random Walk Distance

- Random walk distance
  - $R_A{}^l = \sum_{\tau=1}^{l} c(1-c)^{\tau} P_A{}^{\tau}$
  - τ length of path
  - c decay factor
  - P_A probability transition matrix

Initialization of Clustering Method

- Density Function
  - $f\ (v_i) = \sum_{v_j \in V}(1 - e^{\frac{d(v_i,vj)^2}{2\sigma^2}})$
  - σ is a user-specified parameter

- Select the densest k vertices on $f\ (v_i)$ as the initial centroids $\{c1^0,\ldots,c1^k\}$

Clustering Process

- Assign vertex to its closest centroid c*
  - $c^* = \text{argmax}_{c_j^t} d(v_i, c_j^t)$

- Compute the "average point" of a cluster V_i
  - $R_A^l(\text{avg}(v_i), v_j) = \frac{1}{|V_i|} \sum_{v_k \in V_i} R^l{}_A(v_k, v_j)$

- Find the new centroid c_i^{t+1} in cluster V_i
  - $c_i^{t+1} = \text{argmin}\ v_j \in V_i\ ||R_A^1(v_j) - R_A^1(\text{avg}(v_i))||$

Weight Self-Adjustment

- If a large portion of vertices within clusters shares the same value of a certain attribute a_i, it means that ai has a good clustering accuracy.

- If vertices within clusters have a very random distribution on values of a certain attribute a_i, then a_i is not a good clustering attribute.

Weight Self-Adjustment

- $Vote_i(v_p,v_q) = 1$ if v_p and v_q share the same value on a_i
  -
    $$\omega_i^{t+1} = \frac{1}{2}(\omega_i^t + \Delta\omega_i^t) = \frac{1}{2}(\omega_i^t + \frac{m \sum_{j=1}^{k} \sum_{v \in V_j} vote_i(c_j, v)}{\sum_{p=1}^{m} \sum_{j=1}^{k} \sum_{v \in V_j} vote_p(c_j, v)})$$

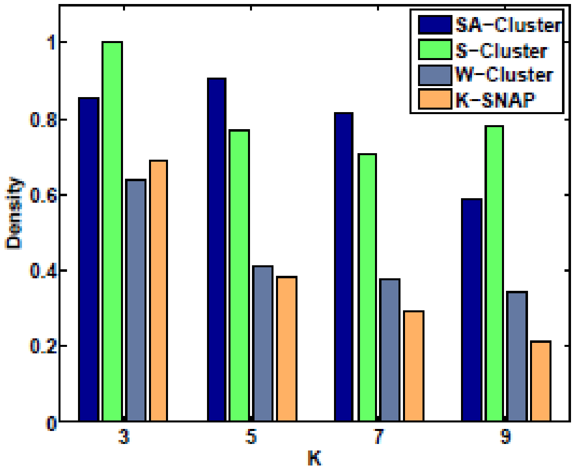
Clustering Procedure

1. Assign vertices to their closest centroids
2. Update cluster centroids
3. Adjust attribute edge weights $\{w_1,\ldots,w_m\}$
4. Re-calculate the random walk distance matrix R_A

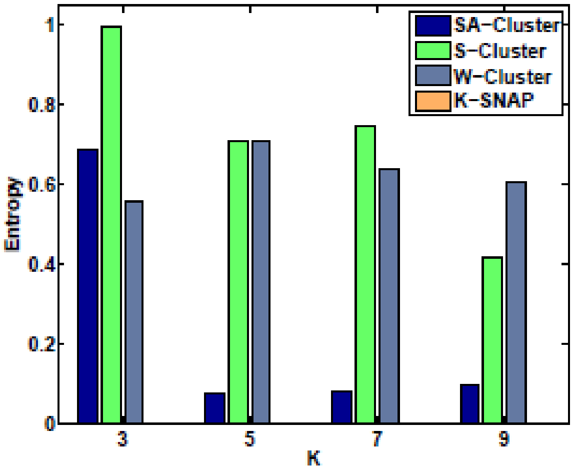
Experimental Result

- SA-Cluster: considers both structural and attribute similarities
- S-Cluster: only considers structural distance
- W-Cluster: consider structural and attribute similarities equally important
- K-SNAP: partition vertices with the same attribute values into a cluster

Experimental Result



(a) density



(b) entropy

Model-based Approach(SIGMOD12)

- Avoids the artificial design of a distance measure

- Based on probabilistic model
  - Fuses both structural and attribute information
  - Vertices from the same cluster behave similar, while vertices from different clusters can behave differently.

- Clustering with the proposed model can be transformed into a probabilistic inference problem
  - Model enforces the intra-cluster similarity by asserting that the attribute values and edge connections of a vertex should depend on its cluster label

Problem Statement

- An attributed graph G is a 4-tuple (V,E,Λ,F)
  - $V = \{v_1,v_2,\ldots,v_N\}$: a set of N vertices
  - $E = \{(v_i,v_j)\}$:a set of edges
  - $\Lambda = \{a^1,a^2,\ldots,a^T\}$: a set of T categorical attributes
  - $F = \{f_1,f_2,\ldots,f_T\}$ **:** a set of functions map each element in attribute vector to attribute value i.e. $f_t(v_i)$

Bayesian Model for Clustering

- Given a set of vertices V, a set of attributes Λ, and the number of clusters K, the model defines a joint probability distribution over all possible partitions over V.

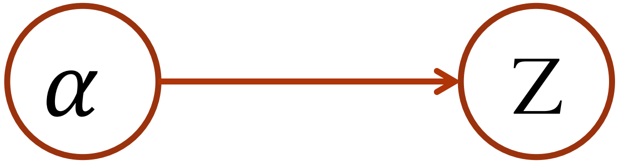
- The goal is to find the most probable clustering with the maximum probability.

Notations

- X : adjacency matrix (N*N)
  - X_{ij} is a binary random variable
  - Indicates the existence of edge
- Y : attribute matrix (N*T)
  - Y_i^t is a categorical random variable
  - Denotes the value of attribute a^t associated with vertex v_i
- Z : clustering of vertices (N*1)
  - Z_i is a categorical random variable
  - Denotes the label of vertex v_i

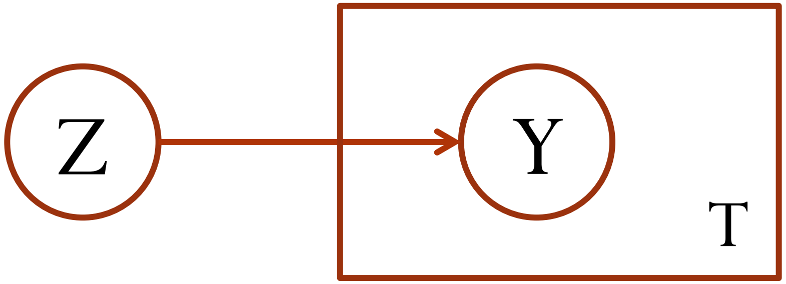
Generating Z

- Cluster label Z_i of each vertex v_i is sampled from multinomial distribution independently
  - $P(Z_i = k \mid \alpha) = \alpha_k$, $k = 1,2,\ldots K$
  - α_k denotes the proportion of the vertices belonging to cluster k
  - $\sum_{k=1}^{K} \alpha_k = 1, \alpha_k \in [0, 1]$



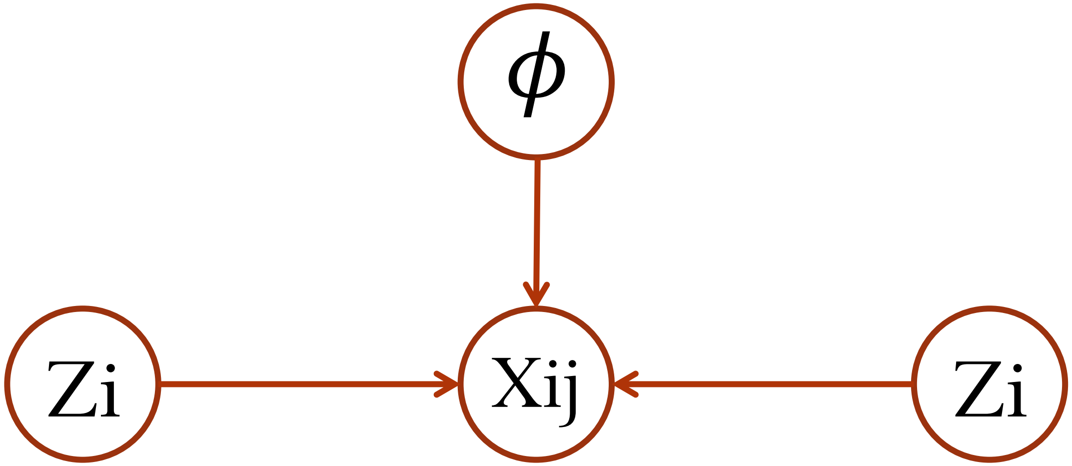
Generating Y

- Given cluster label Z_i for vertex v_i, the attribute values of v_i is by
  - $P(Y_i^t = m \mid \theta_{Zi}{}^t) = \theta_{Zim}{}^t$, m = 1,2,…, M^t
  - M^t is the size of the domain dom(a^t)
  - $\theta_{Zim}{}^t$ denotes the proportion of vertices in cluster Z_i that take the m-th value in dom(a^t)
  - $\sum_{m=1}^{Mt} \theta_{Zim}{}^t = 1, \theta_{Zim}{}^t \in [0, 1]$

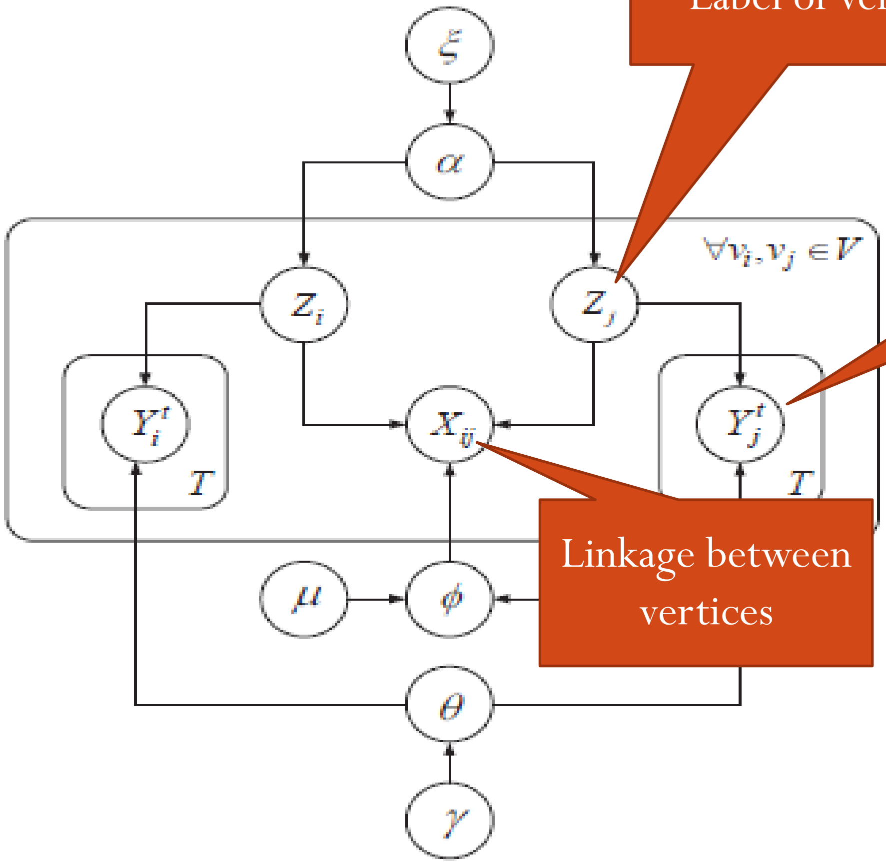


Generating X

- Given the cluster labels Z_i and Z_j for vertex v_i and v_j, the indicator X_{ij} denotes the existence of an edge between them
  - X_{ij} is a binary variable taking value 0 or 1
  - $P(X_{ij} | \phi_{ZiZj}) = (1-\phi_{ZiZj})^{1-Xij}(\phi_{ZiZj})^{Xij}$
  - $\phi_{ZiZj} \in [0, 1]$



Graphical representation



Label of vertex

Attribute value of vetex

Linkage between vertices

Model Definition

- Given the hyper-parameters $\epsilon, \gamma, \mu, \nu$, the joint distribution over α, θ, ϕ, X, Y, Z can be decomposed as

$$\begin{aligned} & p(\alpha, \theta, \phi, \mathbf{X}, \mathbf{Y}, \mathbf{Z} | \xi, \gamma, \mu, \nu) \\ = \; & p(\alpha | \xi) p(\theta | \gamma) p(\phi | \mu, \nu) p(\mathbf{Z} | \alpha) p(\mathbf{X} | \mathbf{Z}, \phi) p(\mathbf{Y} | \mathbf{Z}, \theta), \end{aligned}$$

Model-based Clustering of Attributed Graph

- The problem of clustering a given attributed graph (X,Y) can be transformed into a standard probabilistic inference problem
- Find the clustering Z conditioning on X,Y such that
  - $Z^* = \text{argmax}_z\, p(Z|X,Y)$

Challenges

- The first difficulty is the maximization over the N variables $Z = \{Z_1, Z_2, \ldots, Z_N\}$. For large N, the global maximization is computationally prohibitive

- The second difficulty is in the calculation of the posterior distri

$$p(\mathbf{Z}|\mathbf{X},\mathbf{Y}) = \iiint p(\alpha,\theta,\phi,\mathbf{Z}|\mathbf{X},\mathbf{Y})\mathrm{d}\alpha\mathrm{d}\theta\mathrm{d}\phi,$$

  - where

$$p(\alpha,\theta,\phi,\mathbf{Z}|\mathbf{X},\mathbf{Y}) = \frac{p(\alpha,\theta,\phi,\mathbf{X},\mathbf{Y},\mathbf{Z})}{\sum_{\mathbf{Z}}\iiint p(\alpha,\theta,\phi,\mathbf{X},\mathbf{Y},\mathbf{Z})\mathrm{d}\alpha\mathrm{d}\theta\mathrm{d}\phi}.$$

Solution

- Variational algorithm to solve the problem
  - Approximate distribution p(α,θ,ϕ,Z|X,Y) by variational distribution Q = $q(\alpha)q(\theta)q(\phi)\prod_i q(Z_i)$
  - $$\begin{aligned}\mathbf{Z}^\star &= \arg\max_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\mathbf{Y}) \\ &= \arg\max_{\mathbf{Z}} \iiint p(\alpha,\theta,\phi,\mathbf{Z}|\mathbf{X},\mathbf{Y})\mathrm{d}\alpha\mathrm{d}\theta\mathrm{d}\phi \\ &\approx \arg\max_{\mathbf{Z}} \iiint q(\alpha,\theta,\phi,\mathbf{Z})\mathrm{d}\alpha\mathrm{d}\theta\mathrm{d}\phi \\ &= \arg\max_{\mathbf{Z}} \iiint q(\alpha)q(\theta)q(\phi)\prod_i q(Z_i)\mathrm{d}\alpha\mathrm{d}\theta\mathrm{d}\phi \\ &= \arg\max_{\mathbf{Z}} \prod_i q(Z_i) \\ &= \left[\arg\max_{Z_1} q(Z_1), \arg\max_{Z_2} q(Z_2), \ldots, \arg\max_{Z_N} q(Z_N)\right]\end{aligned}$$
  - The global maximization over Z reduces to local maximization over each Z_i independently

Variational Approximation

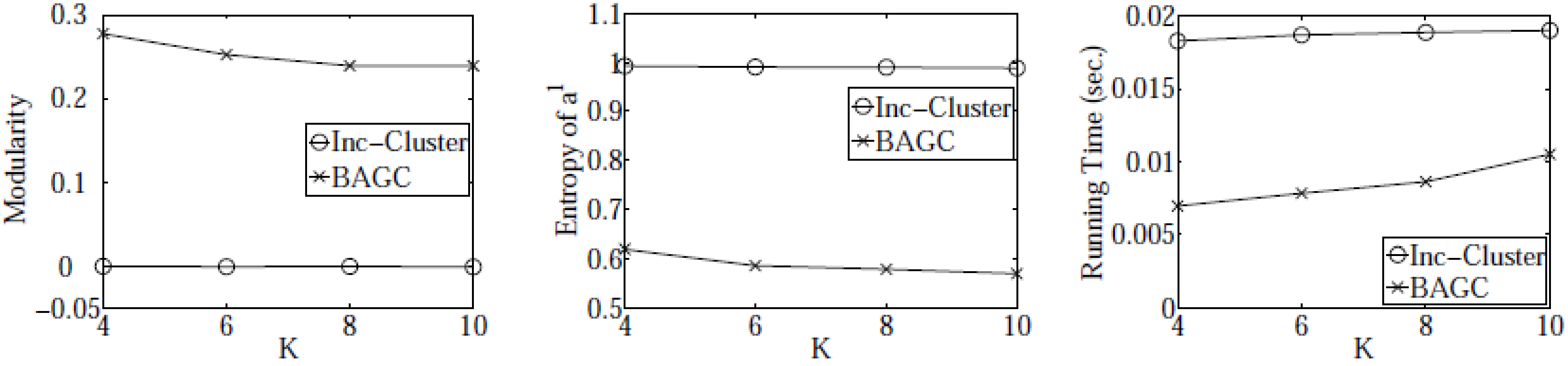
- Approximate P(Z|X,Y) by distribution Q(Z) is to maximize the distribution :L $= \sum_z Q(Z) \log \frac{P(Z,X,Y)}{Q(Z)}$
  - $D_{KL}(Q||P) = \sum_z Q(Z) log \frac{Q(Z)}{P(Z|X,Y)}$
  - $D_{KL}(Q||P) = \sum_z Q(Z) log \frac{Q(Z)}{P(Z,X,Y)} + \log p(X,Y)$
  - $\log p(X,Y) = D_{KL}(Q||P) - \sum_z Q(Z) log \frac{Q(Z)}{P(Z,X,Y)}$
  - $\log p(X,Y) = D_{KL}(Q||P) + \sum_z Q(Z) log \frac{P(Z,X,Y)}{Q(Z)}$

Maximize L

In order to maximize the objective function L, we take the derivatives of L with respect to variational parameters and set these derivatives to zeros.

$$\nabla \widetilde{L}(q) = \left(\frac{\partial \widetilde{L}}{\partial \tilde{\xi}}, \frac{\partial \widetilde{L}}{\partial \tilde{\gamma}}, \frac{\partial \widetilde{L}}{\partial \tilde{\mu}}, \frac{\partial \widetilde{L}}{\partial \tilde{\nu}}, \frac{\partial \widetilde{L}}{\partial \tilde{\beta}} \right) = 0.$$

Experimental Result



(a) Modularity (b) Entropy of Attribute 1 (c) Running Time

Figure 2: Clustering Performance on Political Blogs

Conclusion

- Introduce the problem of attributed graph clustering and its wide applications

- Introduce two state-of-the-art approaches: distance-based attributed graph clustering and model-based attributed graph clustering

- Demonstrate the performance of both algorithms on the real data set.

Reference

1. Yang Zhou, Hong Cheng, Jeffrey Xu Yu: Graph Clustering Based on Structural/Attribute Similarities. PVLDB 2(1):718-729 (2009)
2. A Model-based Approach to Attributed Graph Clustering
3. Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin: Network flows - theory, algorithms and applications. Prentice Hall 1993
4. Lixin Tao, Yongchang Zhao: Multi-way graph partition by stochastic probe. Computers & OR (COR) 20(3):321-347 (1993)
5. Matthew J. Rattigan, Marc E. Maier, David Jensen: Graph clustering with network structure indices. ICML 2007:783-790
6. Charu C. Aggarwal, Haixun Wang: A Survey of Clustering Algorithms for Graph Data. Managing and Mining Graph Data 2010:275-301
7. Hong Cheng, Yang Zhou, Jeffrey Xu Yu: Clustering Large Attributed Graphs: A Balance between Structural and Attribute Similarities. TKDD 5(2):12 (2011)
8. Yang Zhou, Hong Cheng, Jeffrey Xu Yu: Clustering Large Attributed Graphs: An Efficient Incremental Approach. ICDM 2010:689-698

Thank you