notes

alex

January 8, 2020

Contents

1 Graph alignment

1.1 REGAL

1.1.1 Intro

- network alignment, or the task of identifying corresponding nodes in different networks, has applications across the social and natural sciences.
- REGAL (REpresentation learning-based Graph ALignment)
  - Motivated by recent advancements in node representation learning for single-graph tasks
  - a framework that leverages the power of automatically learned node representations to match nodes across different graphs.
- xNetMF, an elegant and principled node embedding formulation that uniquely generalizes to multi-network problems.
- network alignment or matching, which is the problem of finding corresponding nodes in different networks.
  - Crucial for identifying similar users in different social networks or analysing chemical compounds

- Many existing methods try to relax the computationally hard optimization problem, as designing features that directly compared for nodes in different networks is not an easy task.
- we propose network alignment via matching latent, learned node representations.
- **Problem:** Given two graphs G_1 and G_2 with nodesets V_1 and V_2 and possibly node attributes A_1 and A_2 resp., devise an efficient network alignment method that aligns nodes by learning directly comparable node representations Y_1 and Y_2, from which a node mapping $\phi : V_1 \rightarrow V_2$ between the networks can be inferred.
- REGAL is a framework that efficiently identifies node matchings by greedily aligning their latent feature representations.
- They use Cross-Network Matrix Factorization (xNetMF) to learn the representations
  - xNetMF preserves structural similarities rather than proximity-based similarities, allowing for generalization beyond a single network.
  - xNetMF is formulated as matrix factorization over a similarity matrix which incorporates structural similarity and attribute agreement between nodes in disjoint graphs.
  - Constructing the similarity matrix is tough, as is requires computing all pairs of similarities between nodes in the multiple networks, they extend the Nyström low-rank approximation, which is commonly used for large-scale kernel machines.
  - This makes xNetMF a principled and efficient implicit matrix factorization-based approach.
- our approach can be applied to attributed and unattributed graphs with virtually no change in formulation, and is unsupervised: it does not require prior alignment information to find high-quality matchings.
- Many well-known node embedding methods based on shallow architectures such as the popular skip-gram with negative sampling (SGNS) have been cast in matrix factorization frameworks. However, ours is the first to cast node embedding using SGNS to capture structural identity in such a framework

- we consider the significantly harder problem of learning embeddings that may be individually matched to infer node-level alignments.

1.1.2 REGAL Description

- Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two unweighted and undirected graphs (described in the setting of two graphs, but can be extended to more), with node sets V_1 and V_2 and edge sets E_1 and E_2; and possible node attribute sets A_1 and A_2.
  - Graphs does not have to be the same size
- Let $n = |V_{1|} + |V_{2|}$, so the amount of nodes across the two graphs.
- The steps are then:
  1. **Node Identity Extraction:** Extract structure and attribute-related info from all n nodes
  2. **Efficient Similarity-based Representation:** Obtains node embeddings, conceptually by factorising a similarity matrix of the node identities from step 1. However, the computation of this similarity matrix and the factorisation of it is expensive, so they extend the Nystrom Method for low-rank matrix approximation to perform an implicit similarity matrix factorisation by **(a)** comparing similarity of each node only to a sample of $p \ll n$ so-called "landmark nodes" and **(b)** using these node-to-landmark similarities to construct the representations from a decomposition of its low-rank approximation.
  3. **Fast Node Representation Alignment:** Align nodes between the two graphs by greedily matching the embeddings with an efficient data structure (KD-tree) that allows for fast identification of the top-a most similar embeddings from the other graph.
- The first two steps are the xNetMF method

1. Step 1
   - The goal of REGAL's representation learning module, xNetMF, is to define node "identity" in a way that generalizes to multi-network problems.

- As nodes in multi-network problems have no direct connections to each other, their proximity can't be sampled by random walks on separate graphs. This is overcome by instead focusing on more broadly comparable, generalisable quantities: Structural Identity which relates to structural roles and Attribute-Based Identity.
- **Structural Identity**: In network alignment, the well-established assumption is that aligned nodes have similar structural connectivity or degrees. Thus, we can use the degrees of the neighbours of a node as structural identity. They also consider neighbors up to k hops from the original node.
  - For some node $u \in V$, R_u^k is then the set of nodes at exactly (up to??) k hops from u. We could capture the degrees of these nodes within a vector of length the highest degree within the graph (D) d_u^k where the i'th entry of $d_u^k(i)$ then denotes the amount of nodes in R_u^k of degree i. This will however potentially be very long and very sparse, if a single node has a high degree, forcing up the length of d_u^k. Instead, nodes are bin'ned together into $b = [log_2(D)]$ logarithmically scaled buckets with entry i of d_u^k contains number of nodes $u \in R_u^k$ such that $floor([log_2(deg(u))]) = i$. Is both much shorter ($log_2(D)$) but also more robust to noise.
- **Attribute-Based Identity**: Given F node attributes, they create for each node u an F-dimensional vector f_u representing the values of u. So $f_u(i)$ = the i'th attribute of u.
- **Cross-Network Node Similarity**: Relies on the structural and attribute information rather than direct proximity: $sim(u, v) = exp[-\gamma_s \cdot \|d_u - d_v\|_2^2 - \gamma_a \cdot dist(f_u, f_v)]$