Random walk graph-based Semisupervised Classification

Jerónimo Arenas-García

Universidad Carlos III de Madrid

jeronimo.arenas@uc3m.es

January 17, 2018

Contents

Sections 1–3 are covered in "Semi-supervised Classification of Network Data using very few Labels," by Frank Lin and William W. Cohen. Rest of the material will probably be presented in ICASSP 2018.

Semisupervised learning

Motivation

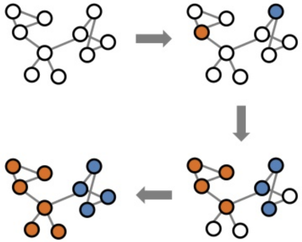
- Exploit unlabeled training data for learning
- Reduce the requiring labeling effort

According to reference paper, we can consider two approaches:

- Active Learning: that selects iteratively additional patterns to label
- SS methods: that exploit the distribution of unlabeled data without implying further labeling.

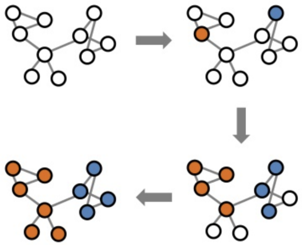
The overall objective of both approaches is to reduce the amount of labeled instances required to achieve a given level of accuracy.

Graph-based SSL



- Instance space is viewed as a graph
- Patterns are nodes
- Similarities between instances define weighted edges
- Possible with any kind of data, provided that we can compute similarities
- Specially interesting when no features are available (e.g., with network data)
- Scalability properties can make them superior to other feature-based methods

Graph-based SSL: Existing approaches



- Iterative label propagation
  - Propagate class labels over graph edges
- Kernel on graphs
  - Use graph methods to compute distances among any two nodes (i.e., to compute kernel values)
- Graph partitioning algorithms
  - Based on Community detection or spectral clustering
  - Label clusters according to available labels
- Random walk methods
  - Core computation: calculus of the dominant eigenvector of a transition matrix (Power Method)
  - Can be highly scalable

Notation

- A graph $\mathcal{G}$ is a collection of nodes and edges: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
- $W_{i,j}$ is the weight of the directed edge from node j to node i.
  - $W_{i,j} > 0$ if $(v_j, v_i) \in \mathcal{E}$
- All nodes belong to one class, and only one class.
  - $\mathcal{U}$ is the set of unlabeled nodes.
  - $\mathcal{L}$ is the set of labeled nodes.
  - $\mathcal{L} = \mathcal{V} \setminus \mathcal{U}$
  - $\mathcal{L}_k$ is the set of labeled nodes with label $y_i = k$.
  - $\mathcal{Y} = \{1, 2, \ldots, c\}$, where c is the number of classes.

1. Introduction to Graph-Based SSL
2. **Random Walk in Graphs**
3. Random Walk with Restart
4. Tuning the seeding vectors
5. Results

Random Walk in Graphs

1. Assume a Markov chain with one state per node and transitions between states with probabilities

$$Pr\left\{X_t = v_i | X_{t-1} = v_j\right\} = \frac{W_{i,j}}{\sum_\ell W_{\ell,j}} = \widetilde{W}_{i,j}$$

2. Matrix $\widetilde{\mathbf{W}}$, with $[\widetilde{\mathbf{W}}]_{i,j} = \widetilde{W}_{i,j}$ is a column stochastic matrix.
   - All elements are positive
   - Column elements sum up to one
3. The transition matrix $\widetilde{\mathbf{W}}$ characterizes how probability evolves from one state to the next one.
4. Multiplying from the right by a vector $\mathbf{r}_t$ containing the probability of each state, gives the probability distribution of states for $t+1$, i.e.,

$$\mathbf{r}_{t+1} = \widetilde{\mathbf{W}}\,\mathbf{r}_t$$

Random Walk in Graphs (II)

4. Multiplying from the right by a vector $\mathbf{r}_t$ containing the probability of each state, gives the probability distribution of states for $t+1$, i.e.,

$$\mathbf{r}_{t+1} = \widetilde{\mathbf{W}}\,\mathbf{r}_t$$

   - Since $\mathbf{r}_t$ is defined as the vector containing the probability of each state at time t, necessarily $\mathbf{r}_t \geq 0$ and $\mathbf{r}_t^\top \mathbf{1} = 1$
   - Since $\widetilde{\mathbf{W}}$ is column stochastic, this condition propagates over time. (Demo)

5. Lemma: If the graph is strongly connected, the dominant right eigenvector of $\widetilde{\mathbf{W}}$ provides the *stable distribution* of the "random walker", i.e., the probability the walker is at each node as $t \to \infty$.

Random Walk in Graphs (III): Stable Distribution (Demo)

- The stable distribution $\boldsymbol{\pi}$ verifies $\boldsymbol{\pi} = \widetilde{\mathbf{W}}\ \boldsymbol{\pi}$
  - $\rho(\widetilde{\mathbf{W}})$ should be 1
  - Its corresponding eigenvector has non-negative components.
- Demonstration follows from the Perron-Frobenius theorem for irreducible matrices.

`https://en.wikipedia.org/wiki/PerronFrobenius_theorem`

Random Walk in Graphs (IV): Exploiting sparsity

- If $\widetilde{\mathbf{W}}$ is very sparse, rather than eigenvalue decomposition, is is more efficient to apply the Power Method, which scales linearly with $|\mathcal{E}|$:

$$\boldsymbol{\pi} = \lim_{t \to \infty} (\widetilde{\mathbf{W}})^t \, \mathbf{u}(0)$$

- Stopping criterion based on the norm of differences among the solution vectors at adjacent iterations.

Measuring node influence

If we want to measure the influence of different labels on the nodes, we need to introduce an additional mechanism: Random Walk with Restart.

Random Walk with Restart: Teleportation

At every iteration a coin is tossed, and the walk is restarted from a random node with probability vector $\boldsymbol{\theta}$:

$$Pr\left\{X_t = v_i | X_{t-1} = v_j; \boldsymbol{\theta}, d\right\} = \begin{cases} \widetilde{W}_{i,j} & w.p. \quad 1-d \\ \theta_i & w.p. \quad d \end{cases}$$

- d is the teleportation probability.
- $\boldsymbol{\theta}^\top \mathbf{1} = 1$ and $\theta_i \geq 0$

New Transition Probability

$$\widetilde{\widetilde{\mathbf{W}}} = (1-d)\widetilde{\widetilde{\mathbf{W}}} + d\boldsymbol{\theta}\mathbf{1}^\top$$

- As before, a column stochastic matrix
- The power method can be used to find the stable distribution

Random Walk with Restart (II): MultiRankWalk

Proposal: Use the stable distribution when $\boldsymbol{\theta} = \mathbf{e}_i$ to estimate the influence of node v_i on every other node in the graph.

MultiRankWalk algorithm for multiclass problems

1. Define a seed vector for each class $\boldsymbol{\theta}(k)$, $k = 1, \ldots, c$, where:

$$[\boldsymbol{\theta}(k)]_i = \begin{cases} \frac{1}{|\mathcal{L}_k|}, & i \in \mathcal{L}_k \\ 0, & \text{otherwise} \end{cases}$$

2. Obtain the corresponding stable distributions:

$$\boldsymbol{\pi}(k) = \widetilde{\widetilde{\mathbf{W}}}(\boldsymbol{\theta}(k))\, \boldsymbol{\pi}(k)$$

3. Classify each node in the graph according to

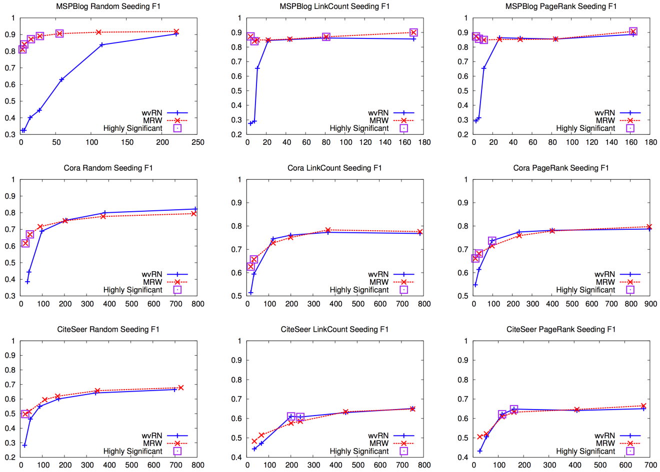
$$\hat{y}_i = \arg\max_k \pi_i(k)$$

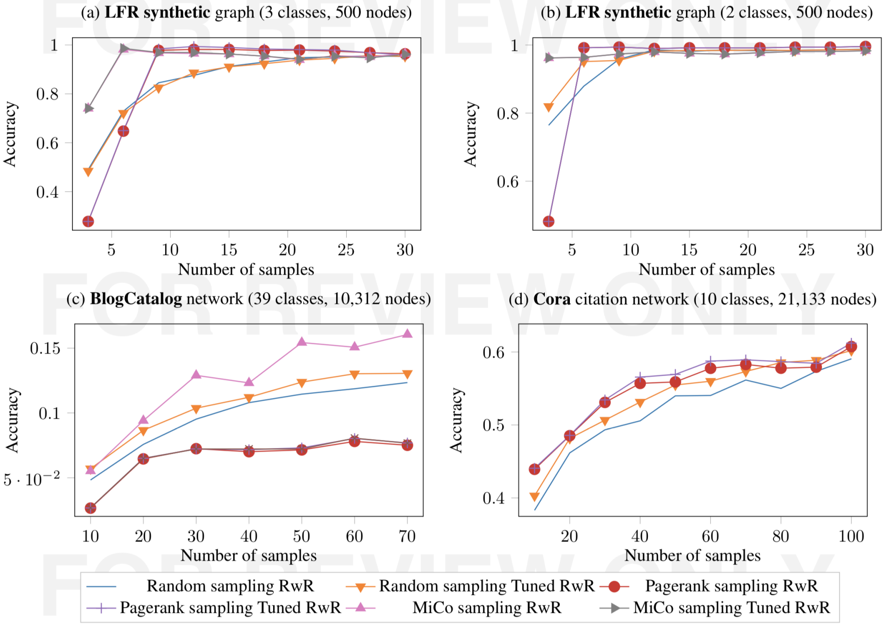
(alternatively: use influence vector to train a classifier)

RwR (III): Authorative Seed Selection

Use authorative sites as the label nodes seems advantageous since:

- They are probably easier to categorize
- Their link structure facilitates the spread of label information




(a) LFR synthetic graph (3 classes, 500 nodes)
(b) LFR synthetic graph (2 classes, 500 nodes)
(c) BlogCatalog network (39 classes, 10,312 nodes)
(d) Cora citation network (10 classes, 21,133 nodes)
Accuracy
Number of samples
Random sampling RwR
Random sampling Tuned RwR
Pagerank sampling RwR
Pagerank sampling Tuned RwR
MiCo sampling RwR
MiCo sampling Tuned RwR