Label propagation for community detection: A review

Lovro Šubelj and Marko Bajec
University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia

December 2012

Abstract

Label propagation is an efficient approach for community detection initially proposed by Raghavan et al. (2007). We here give a comprehensive review various advances of label propagation that improve on its performance, stability, complexity and other.

Contents

1 Introduction

Complex real-world networks commonly consist of communities, i.e., groups of nodes that are densely linked within and only sparsely linked between (also dense subgraphs of sparse graphs). Label propagation is an efficient approach for community detection initially proposed by Raghavan et al. (2007). Here labels are propagated among the nodes thus each node is assigned a label shared by most of its neighbors (i.e., majority label). When an equilibrium is reached, connected groups of nodes sharing unique label are classified into the same community. Note that the approach has near linear time complexity and is applicable to networks with millions of nodes and links on a standard computer.

In this work we give a comprehensive review of various advances of standard label propagation that improve on its performance, stability, complexity and other.

Preliminaries

For the purposes of this work, a network is represented as a simple undirected weighted graph. In the case of directed weighted multi-networks, directed links are first treated as undirected, whereas all loops are discarded. Multiple links between nodes are then replaced by a single link with weight equal to the sum of weights of the original links (or equal to the number of links for unweighted networks).[1]

Terminology

A brief description of used terms is given below.

$\mathbf{G}$	network represented as a simple undirected weighted graph
$\mathbf{C}$	set of (non-overlapping) communities of network G
$\mathbf{N}$	list of nodes of network G, $n = \|N\|$
$\mathbf{L}$	set of links of network G, $m = \|L\|$
$\mathbf{\Theta}$	coloring of network G, $\chi = \|\Theta\|$
$\mathbf{N}^{\theta}$	nodes with color $\theta \in \Theta$, $N^{\theta} \subseteq N$
$\mathbf{N^c}$	nodes in community $c \in C$, $N^c \subseteq N$
$\Gamma_{\mathbf{i}}$	neighbors of node $i \in N$, $\Gamma_i \subseteq N$
$\Gamma_{\mathbf{i}}^{\mathbf{c}}$	neighbors of node $i \in N$ in $c \in C$, $\Gamma_i^c \subseteq \Gamma_i$
$\psi_{\mathbf{i}}$	normalized index of node $i \in N$, $\psi_i \in [0, 1)$
$\theta_{\mathbf{i}}$	color of node $i \in N$, $\theta_i \in \Theta$
$\mathbf{c_i}$	label of node $i \in N$, $c_i \in C$
$\mathbf{p_i}$	preference of node $i \in N$, $p_i \in \mathbb{R}$
$\mathbf{k_i}$	degree of node $i \in N$, $k_i = \|\Gamma_i\|$
$\mathbf{k_i^c}$	community degree of node $i \in N$, $k_i = \|\Gamma_i^c\|$
$\Gamma_{\mathbf{ij}}$	common neighbors of nodes $i, j \in N$, $\Gamma_{ij} = \Gamma_i \cap \Gamma_j$
$\mathbf{w_{ij}}$	weight of link $ij \in L$, $w_{ij} \in \mathbb{R}$
$\mathbf{k_{ij}}$	common degree of nodes $i, j \in N$, $k_{ij} = \|\Gamma_{ij}\|$

[1] Link weights must be taken into account when computing network statistics.

2 Label propagation

In the following we review different label propagation strategies.

2.1 Synchronous propagation

References. (Raghavan et al., 2007; Leung et al., 2009)

Description. Synchronous label propagation, where nodes' labels are identified in parallel (Alg. 1).

Comments. Problems with convergence in, e.g., near-bipartite or star-like networks.

Algorithm 1 Synchronous propagation

```
Input: network G
Output: communities C
{Label initialization.}
for i ∈ N do
   c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
   {Synchronous propagation.}
   for all i ∈ N do
      {Label identification.}
      c_i ← argmax_c k_i^c {Majority label.}
   end for
end while
```

2.2 Asynchronous propagation

References. (Raghavan et al., 2007; Leung et al., 2009)

Description. Asynchronous label propagation, where nodes' labels are identified sequentially (Alg. 2).

Comments. Problems with performance and stability in, e.g., large networks.

Algorithm 2 Asynchronous propagation

```
Input: network G
Output: communities C
{Label initialization.}
for i ∈ N do
   c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
   {Asynchronous propagation.}
   shuffle(N)
   for i ∈ N do
      {Label identification.}
      c_i ← argmax_c k_i^c {Majority label.}
   end for
end while
```

2.3 Semi-synchronous propagation

References. (Cordasco & Gargano, 2010, 2011)

Description. Semi-synchronous label propagation, where independent nodes' labels are identified in parallel (Alg. 3). (Independent nodes are decided according to a coloring of the network.)

Comments. Network coloring can be computationally expensive.

Algorithm 3 Semi-synchronous propagation

Input: network G, coloring Θ
Output: communities C

{Label initialization.}
for $i \in N$ **do**
    $c_i \leftarrow i$ {Unique label.}
end for
{Label propagation.}
while not *terminated* **do**
    {Semi-synchronous propagation.}
    shuffle(Θ)
    for $\theta \in \Theta$ **do**
        for all $i \in N^\theta$ **do**
            {Label identification.}
            $c_i \leftarrow \mathrm{argmax}_c \, k_i^c$ {Majority label.}
        end for
    end for
end while

---

3 Label identification

In the following we review different majority label identification strategies.

3.1 General strategies

In the following we review general strategies for majority label identification.

3.1.1 Standard propagation

References. (Raghavan et al., 2007)

Description. Majority label identification based solely on the neighbors' labels (Alg. 4).

Algorithm 4 Standard propagation

Input: network G
Output: communities C

```
{Label initialization.}
for i ∈ N do
    c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
    for all i ∈ N do
        {Label identification.}
        c_i ← argmax_c k_i^c {Standard propagation.}
    end for
end while
```

3.1.2 Weighted propagation

References. (Šubelj & Bajec, 2011c)

Description. Majority label identification based on neighbors' labels and link weights (Alg. 5).

Algorithm 5 Weighted propagation

Input: network G
Output: communities C

```
{Label initialization.}
for i ∈ N do
    c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
    for all i ∈ N do
        {Label identification.}
        c_i ← argmax_c Σ_{j∈Γ_i^c} w_ij {Weighted propagation.}
    end for
end while
```

3.1.3 Preferenced propagation

References. (Leung et al., 2009)

Description. Majority label identification based on neighbors' labels and preferences (Alg. 6).

Algorithm 6 Preferenced propagation

Input: network G
Output: communities C
{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
  $p_i \leftarrow \dots$ {Initial preference.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $p_i \leftarrow \dots$ {Preference update.}
    $c_i \leftarrow \text{argmax}_c \sum_{j \in \Gamma_i^c} p_j$ {Preferenced propagation.}
    $p_i \leftarrow \dots$ {Preference update.}
  end for
end while

3.2 Performance-based strategies

In the following we review majority label identification strategies that improve performance.

3.2.1 Strength propagation

References. (Xie & Szymanski, 2011)

Description. Majority label identification based on neighbors' preferences that are decided according to a strength propagation strategy (Alg. 7). (Parameter w is set to 1.)

Comments. Main rationale here is to improve performance by considering neighbors equivalence.

Algorithm 7 Strength propagation

Input: network G, parameter w
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \mathrm{argmax}_c \sum_{j \in \Gamma_i^c} w_{ij} + wk_{ij}$ {Strength propagation.}
  end for
end while

3.2.2 Degree propagation

References. (Leung et al., 2009)

Description. Majority label identification based on neighbors' preferences that are decided according to a degree propagation strategy (Alg. 8). (Parameter α is set to 0.1.)

Comments. Presumably, rationale here is to improve performance in scale-free real-world networks.

Algorithm 8 Degree propagation

Input: network G, parameter α
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \mathrm{argmax}_c \sum_{j \in \Gamma_i^c} k_j^{\alpha} w_{ij}$ {Degree propagation.}
  end for
end while

3.2.3 Defensive propagation

References. (Šubelj & Bajec, 2011c, 2010)

Description. Majority label identification based on neighbors' preferences that are decided according to a defensive propagation strategy (Alg. 9).

Comments. Main rationale here is to improve performance in (denser) real-world networks.

Algorithm 9 Defensive propagation

Input: network G
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
  $p_i \leftarrow 1/n$ {Initial preference.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \text{argmax}_c \sum_{j \in \Gamma_i^c} p_j w_{ij}$ {Defensive propagation.}
    $p_i \leftarrow \sum_{j \in \Gamma_i^{c_i}} p_j / k_j^{c_i}$ {Defensive preference.}
  end for
end while

3.2.4 Offensive propagation

References. (Šubelj & Bajec, 2011c, 2010)

Description. Majority label identification based on neighbors' preferences that are decided according to a offensive propagation strategy (Alg. 10).

Comments. Main rationale here is to improve performance in sparser real-world networks.

Algorithm 10 Offensive propagation

Input: network G
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
  $p_i \leftarrow 1/n$ {Initial preference.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \text{argmax}_c \sum_{j \in \Gamma_i^c} (1 - p_j) w_{ij}$ {Offensive propagation.}
    $p_i \leftarrow \sum_{j \in \Gamma_i^{c_i}} p_j / k_j^{c_i}$ {Offensive preference.}
  end for
end while

3.2.5 Modularity propagation

References. (Barber & Clark, 2009; Liu & Murata, 2009a)

Description. Majority label identification based on neighbors' preferences that are decided according to a modularity optimization strategy (Alg. 11).

Comments. Main rationale here is to improve performance by directly optimizing community quality function denoted modularity Q (Newman & Girvan, 2004).

Algorithm 11 Modularity propagation

Input: network G
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \text{argmax}_c \sum_{j \in \Gamma_i^c} 2w_{ij} - k_i k_j / m$ {Modularity propagation.}
  end for
end while

3.2.6 Potts model propagation

References. (Boldi et al., 2011a; Ronhovde & Nussinov, 2010; Tibély & Kertész, 2008)

Description. Majority label identification based on neighbors' preferences that are decided according to a Potts model strategy (Alg. 12). (Parameter γ should be set between 0 and 1.)

Comments. Main rationale here is to improve performance by directly optimizing Hamiltonian of the corresponding Potts model. (Density of each community is above $\gamma/(\gamma+1)$.)

Algorithm 12 Potts model propagation

Input: network G, parameter γ
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
end for
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \text{argmax}_c\, k_i^c - \gamma(N^c - k_i^c)$ {Potts model propagation.}
  end for
end while

3.3 Stability-based strategies

In the following we review majority label identification strategies that improve stability.

3.3.1 Momentum propagation

References. (Soman & Narang, 2011)

Description. Majority label identification based on nodes' preferences that are decided according to a momentum propagation strategy (Alg. 13). (Parameter w should be set to 1.)

Comments. Main rationale here is to improve stability by suppressing the occurrence of an epidemic spread (i.e., major community). (Realized by adding a loop with weight w to each node.)

Algorithm 13 Momentum propagation

```
Input: network G, parameter w
Output: communities C
{Label initialization.}
for i ∈ N do
  c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
  for all i ∈ N do
    {Label identification.}
    c_i ← argmax_c Σ_{j∈Γ_i^c} w_ij + wδ(c, c_i) {Momentum propagation.}
  end for
end while
```

3.3.2 Controlled propagation

References. (Soman & Narang, 2011)

Description. Majority label identification based on neighbors' preferences that are decided according to a controlled propagation strategy (Alg. 14).

Comments. Main rationale here is to improve stability by suppressing the occurrence of an epidemic spread (i.e., major community). (Realized by restricting the spread of larger communities.)

Algorithm 14 Controlled propagation

```
Input: network G
Output: communities C
{Label initialization.}
for i ∈ N do
  c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
  for all i ∈ N do
    {Label identification.}
    c_i ← argmax_c Σ_{j∈Γ_i^c} (2 − Σ_{l∈N^c} k_l/m) w_ij {Controlled propagation.}
  end for
end while
```

3.3.3 Attenuated propagation

References. (Leung et al., 2009; Šubelj & Bajec, 2011c)

Description. Majority label identification based on neighbors' preferences that are decided according to a attenuated propagation strategy (Alg. 15). (Parameter $\delta_{\max}$ is fixed to 0.5, whereas δ_{step} should be set around 0.05. Alternatively, one can set parameter δ to, e.g., 0.15.)

Comments. Main rationale here is to improve stability by suppressing the occurrence of an epidemic spread (i.e., major community). (Realized by restricting how far a community can spread.)

Algorithm 15 Attenuated propagation

Input: network G, parameters $\delta_{\max}$, δ_{step}
Output: communities C

{Label initialization.}
for $i \in N$ **do**
  $c_i \leftarrow i$ {Unique label.}
  $p_i \leftarrow 0$ {Initial preference.}
end for
{Parameter initialization.}
$\delta \leftarrow \delta_{\max}$ or $\delta \leftarrow 0$ {Initial parameter.}
{Label propagation.}
while not *terminated* **do**
  for all $i \in N$ **do**
    {Label identification.}
    $c_i \leftarrow \text{argmax}_c \sum_{j \in \Gamma_i^c} (1 - \delta p_j) w_{ij}$ {Attenuated propagation.}
    if c_i *changed* **then**
      $p_i \leftarrow 1 + \min_{j \in \Gamma_i^{c_i}} p_j$ {Attenuated preference.}
    end if
  end for
  {Parameter estimation.}
  $\delta \leftarrow \max\{0, \delta - \delta_{\text{step}}\}$ or $\delta \leftarrow \min\{\delta_{\max}, \textit{ratio of labels changed}\}$ {Attenuated parameter.}
end while

3.3.4 Balanced propagation

References. (Šubelj & Bajec, 2011b)

Description. Majority label identification based on neighbors' preferences that are decided according to a balanced propagation strategy (Alg. 16). (Parameter λ is fixed to 0.5, whereas μ should be set between 0 and 2.)

Comments. Main rationale here is to improve stability of the asynchronous label propagation. (Increasing parameter μ improves stability, but it also increases complexity.)

Algorithm 16 Balanced propagation

Input: network G, parameters λ, μ
Output: communities C

```
{Label initialization.}
for i ∈ N do
   c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
   shuffle(N)
   for i ∈ N do
      {Label identification.}
      p_i ← ψ_i or p_i ← 1/(1 + e^{−μ(ψ_i−λ)}) {Balanced preference.}
      c_i ← argmax_c Σ_{j∈Γ_i^c} p_j w_ij {Balanced propagation.}
   end for
end while
```

Algorithm 17 Selective propagation

Input: network G, parameter ω
Output: communities C

```
{Label initialization.}
for i ∈ N do
   c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
   for all i ∈ N do
      {Label identification.}
      if k_i^{c_i}/k_i ≤ ω then
         c_i ← argmax_c Σ_{j∈Γ_i^c} w_ij {Selective propagation.}
      end if
   end for
end while
```

Algorithm 18 Passive propagation

Input: network G
Output: communities C

```
{Label initialization.}
for i ∈ N do
   c_i ← i {Unique label.}
end for
{Label propagation.}
while not terminated do
   for all i ∈ N do
      {Label identification.}
      if ∃c ∈ C\{c_i} : k_i^c ≥ k_i^{c_i} then
         c_i ← argmax_c Σ_{j∈Γ_i^c} w_ij {Passive propagation.}
      end if
   end for
end while
```

3.4 Complexity-based strategies

In the following we review majority label identification strategies that improve complexity.

3.4.1 Selective propagation

References. (Leung et al., 2009)

Description. Majority label identification based on neighbors' labels and link weights, and a selective propagation strategy (Alg. 17). (Parameter ω is set to 0.5.)

Comments. Main rationale here is to improve complexity by propagating the labels selectively. (Realized by discarding the nodes whose labels are unlikely to change.)

3.4.2 Passive propagation

References. (Xie & Szymanski, 2011)

Description. Majority label identification based on neighbors' labels and link weights, and a passive propagation strategy (Alg. 18).

Comments. Main rationale here is to improve complexity by propagating the labels passively. (Realized by discarding the nodes whose labels cannot change.)

4 Label ties

In the following we review different strategies for resolving majority label ties.

4.1 Random label

References. (Raghavan et al., 2007)

Description. Ties are broken uniformly at random.

4.2 Label retention

References. (Barber & Clark, 2009)

Description. Ties are broken uniformly at random, while a label is retained if among majority labels.

4.3 Label inclusion

References. (Leung et al., 2009)

Description. Ties are broken uniformly at random, while a label is included in the majority label identification.

4.4 Label priority

References. (Cordasco & Gargano, 2010, 2011)

Description. Ties are broken due to label priority.

Comments. Label priority is an arbitrary number defined *a priori*.

5 Propagation criteria

In the following we review different propagation termination criteria.

5.1 Label equilibrium

References. (Raghavan et al., 2007)

Description. Propagation is terminated, when each node's label equals the majority label.

5.2 Label convergence

References. (Barber & Clark, 2009)

Description. Propagation is terminated, when each node's label equals the label on the previous step.

5.3 Label semi-convergence

References. (Cordasco & Gargano, 2010, 2011)

Description. Propagation is terminated, when each node's label equals the label on the previous step, or the step before.

5.4 Threshold convergence

References. (Šubelj & Bajec, 2011c, 2010)

Description. Propagation is terminated, when the number of steps exceeds the defined threshold. (The threshold is set to, e.g., 100.)

6 Advanced propagation

In the following we give references to other advances of label propagation not reviewed in this work.

6.1 Hierarchical propagation

References. (Leung et al., 2009; Šubelj & Bajec, 2010, 2011c, 2012a; Xie & Szymanski, 2012)

6.2 Refining propagation

References. (Šubelj & Bajec, 2011c, 2010; Coscia et al., 2012)

6.3 Hybrid propagation

References. (Barber & Clark, 2009; Liu & Murata, 2009a; Šubelj & Bajec, 2010, 2011c,b; Gregory, 2010; Wu et al., 2012)

6.4 Divisive propagation

References. (Pang et al., 2009a)

6.5 Online propagation

References. (Pang et al., 2009b; Leung et al., 2009)

6.6 Parallel propagation

References. (Soman & Narang, 2011; Leung et al., 2009; Cordasco & Gargano, 2010, 2011; Boldi et al., 2011a; Rees & Gallagher, 2012a)

7 Other networks

In the following we give references to extensions of label propagation to other types of network.

Bipartite networks

References. (Liu & Murata, 2009b,c,a; Xie & Szymanski, 2012)

Multi-partite networks

References. (Liu & Murata, 2011)

8 Other groups

In the following we give references to extensions of label propagation to other groups of network nodes.

Overlapping community detection

References. (Gregory, 2010; Xie et al., 2011; Xie & Szymanski, 2012; Coscia et al., 2012; Wu et al., 2012; Rees & Gallagher, 2012b,a; Leung et al., 2009; Soman & Narang, 2011)

Community and module detection

References. (Šubelj & Bajec, 2012b, 2011a, 2012a)

9 Applications

In the following we give references to different applications of label propagation.

Network robustness

References. (Boldi et al., 2011b)

Network compression

References. (Boldi et al., 2011a)

References

Barber, M. J., & Clark, J. W. (2009). Detecting network communities by propagating labels under constraints. *Physical Review E*, *80*(2), 026129.

Boldi, P., Rosa, M., Santini, M., & Vigna, S. (2011a). Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the International World Wide Web Conference*, (pp. 587–596).

Boldi, P., Rosa, M., & Vigna, S. (2011b). Robustness of social networks: Comparative results based on distance distributions. *e-print arXiv:11104474v1*.

Cordasco, G., & Gargano, L. (2010). Community detection via semi-synchronous label propagation algorithms. In *Proceedings of the IMSAA Workshop on Business Applications of Social Network Analysis*, (pp. 1–8).

Cordasco, G., & Gargano, L. (2011). Label propagation algorithm: A semi–synchronous approach. *International Journal of Social Network Mining*, *1*(1), 3–26.

Coscia, M., Rossetti, G., Giannotti, F., & Pedreschi, D. (2012). DEMON: a local-first discovery method for overlapping communities. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Gregory, S. (2010). Finding overlapping communities in networks by label propagation. *New Journal of Physics*, *12*(10), 103018.

Leung, I. X. Y., Hui, P., Liò, P., & Crowcroft, J. (2009). Towards real-time community detection in large networks. *Physical Review E*, *79*(6), 066107.

Liu, X., & Murata, T. (2009a). Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, *389*(7), 1493.

Liu, X., & Murata, T. (2009b). Community detection in large-scale bipartite networks. In *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1, (pp. 50–57).

Liu, X., & Murata, T. (2009c). How does label propagation algorithm work in bipartite networks. In *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technologies*, (pp. 5–8).

Liu, X., & Murata, T. (2011). Detecting communities in k-partite k-uniform (hyper)networks. *Journal of Computer Science and Technology*, *26*(5), 778–791.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), 026113.

Pang, C., Shao, F., Sun, R., & Li, S. (2009a). Detecting community structure in networks by propagating labels of nodes. In *Proceedings of the International Symposium on Neural Networks*, (pp. 839–846).

Pang, S., Chen, C., & Wei, T. (2009b). A realtime clique detection algorithm: Time-based incremental label propagation. In *Proceedings of the International Conference on Intelligent Information Technology Application*, vol. 3, (pp. 459–462).

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*(3), 036106.

Rees, B. S., & Gallagher, K. B. (2012a). Detecting overlapping communities in complex networks using swarm intelligence for multi-threaded label propagation. In *Proceedings of the International Workshop on Complex Networks*.

Rees, B. S., & Gallagher, K. B. (2012b). Overlapping community detection using a community optimized graph swarm. *Social Network Analysis and Mining*.

Ronhovde, P., & Nussinov, Z. (2010). Local resolution-limit-free potts model for community detection. *Physical Review E*, *81*(4), 046114.

Soman, J., & Narang, A. (2011). Fast community detection algorithm with GPUs and multicore architectures. In *Proceedings of the IEEE International Parallel Distributed Processing Symposium*, (pp. 568 –579).

Šubelj, L., & Bajec, M. (2010). Unfolding network communities by combining defensive and offensive label propagation. In *Proceedings of the ECML PKDD Workshop on the Analysis of Complex Networks*, (pp. 87–104). Barcelona, Spain.

Šubelj, L., & Bajec, M. (2011a). Generalized network community detection. In *Proceedings of the ECML PKDD Workshop on Finding Patterns of Human Behaviors in Network and Mobility Data*, (pp. 66–84). Athens, Greece.

Šubelj, L., & Bajec, M. (2011b). Robust network community detection using balanced propagation. *European Physical Journal B*, *81*(3), 353–362.

Šubelj, L., & Bajec, M. (2011c). Unfolding communities in large complex networks: Combining defensive and offensive label propagation for core extraction. *Physical Review E*, *83*(3), 036103.

Šubelj, L., & Bajec, M. (2012a). Clustering assortativity, communities and functional modules in real-world networks. (p. 21).

Šubelj, L., & Bajec, M. (2012b). Ubiquitousness of link-density and link-pattern communities in real-world networks. *European Physical Journal B*, *85*(1), 32.

Tibély, G., & Kertész, J. (2008). On the equivalence of the label propagation method of community detection and a potts model approach. *Physica A: Statistical Mechanics and its Applications*, *387*(19-20), 4982–4984.

Wu, Z.-H., Lin, Y.-F., Gregory, S., Wan, H.-Y., & Tian, S.-F. (2012). Balanced multi-label propagation for overlapping community detection in social networks. *Journal of Computer Science and Technology*, *27*(3), 468–479.

Xie, J., & Szymanski, B. K. (2011). Community detection using a neighborhood strength driven label propagation algorithm. In *Proceedings of the IEEE International Workshop on Network Science*.

Xie, J., & Szymanski, B. K. (2012). Towards linear time overlapping community detection in social networks. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Xie, J., Szymanski, B. K., & Liu, X. (2011). SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Proceedings of the ICDM Workshop on Data Mining Technologies for Computational Collective Intelligence*.