Updating the Stationary Distribution Vector for an Irreducible Markov Chain

Prakash Chanchana
Carl D. Meyer
Amy N. Langville
Outline

- PageRank Problem
- Updating Problem
- Exact Aggregation
- Approximate Updating Algorithm
- Updating Algorithm with the Iterative Aggregation
- Determining the Partition
- Future Work
Solving for the dominant eigenvector of a matrix $Q$ by using the **Power Method**.

$$
\phi^T Q = \phi^T
$$
Updating Markov Chain

Updating PageRank Problem

Given $\phi^T$ and $Q$ for the original system, find $\pi^T$ of $P$ for updated system.

One way to solve this problem is to apply the Power Method (not good for a Huge matrix)
Motivation

- Working with smaller system
- Paying more attention to the updated states
### Exact Aggregation

1. Partition states into $k$ disjoint groups as $S = G_1 \cup G_2 \cup \ldots \cup G_k$ where each $P_{ii}$ is a square block.

$$P_{n \times n} = \begin{pmatrix}
P_{11} & P_{12} & \cdots & P_{1k} \\
P_{21} & P_{22} & \cdots & P_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
P_{k1} & P_{n2} & \cdots & P_{kk}
\end{pmatrix}$$

2. Find the stochastic complement $S_i$ for each block $P_{ii}$ where

$$S_i = P_{ii} + P_{i*}(I - P_i^*)^{-1}P_{*i}$$

$P_{i*}$ and $P_{*i}$ are, respectively, the $i^{th}$-row and the $i^{th}$-column of block with $P_{ii}$ removed.

$P_{i*}$ is principle submatrix of $P$ obtained by deleting the $i^{th}$-row and the $i^{th}$-column of blocks.
**Exact Aggregation**

Example:

\[ P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}, \text{then} \]

\[ S_1 = P_{11} + P_{12}(I - P_{22})^{-1}P_{21} \]

\[ S_2 = P_{22} + P_{21}(I - P_{11})^{-1}P_{12} \]

3. Form the smaller \( k \times k \) matrix called the *Aggregated Transition Matrix*.

\[
A_{k \times k} = \begin{pmatrix}
    s_1^T P_{11} e & \cdots & s_1^T P_{1k} e \\
    \vdots & \ddots & \vdots \\
    s_k^T P_{k1} e & \cdots & s_k^T P_{kk} e
\end{pmatrix}
\]

where \( s_i^T \) is the stationary probability vector of \( S_i \).
4. Find the stationary distribution of $A$: $\alpha^T = (\alpha_1, \alpha_2, \ldots, \alpha_k)$

5. By the Exact Aggregation Theorem: $\pi^T = (\alpha_1 s_1^T, \alpha_2 s_2^T, \ldots, \alpha_k s_k^T)$

**Advantage**: Get the exact value of $\pi^T$

**Disadvantage**: Very expensive for computing inverse $S_i = P_{ii} + P_{i*}(I - P_i^*)^{-1} P_{*i}$
Special Partition

\[
P = \begin{pmatrix}
G \\
\bar{G}
\end{pmatrix}
\begin{pmatrix}
P_{11} & P_{12} \\
P_{21} & P_{22}
\end{pmatrix}
\]

Partition \(P\) into \(g + 1\) levels where first \(g\) diagonal blocks are \(1 \times 1\)

\[
P = \begin{pmatrix}
p_{11} & \cdots & p_{1g} & P_{1*} \\
\vdots & \ddots & \vdots & \vdots \\
p_{g1} & \cdots & p_{gg} & P_{g*} \\
P_{1*} & \cdots & P_{g*} & P_{22}
\end{pmatrix}
\]

Advantage of this partition:

- Since \(p_{11} \ldots p_{gg}\) are \(1 \times 1\) block \(\Rightarrow\) Stochastic complement = 1 and \(s_i = 1\)
Special Partition

• There is only one stochastic complement

\[ S = P_{22} + P_{21}(I - P_{11})^{-1}P_{12} \]

• By Exact Aggregation Theorem

\[ \tilde{s}^T = (\phi_{g+1}, \ldots, \phi_n) / \sum_{i=g+1}^{n} \phi_i \]

So, the Exact Aggregated Transition Matrix associated with the new partition is

\[
\tilde{A} = \begin{pmatrix}
  p_{11} & \cdots & p_{1g} & | & P_{1*}e \\
  \vdots & \ddots & \vdots & | & \vdots \\
  p_{g1} & \cdots & p_{gg} & | & P_{g*}e \\
  \tilde{s}^T P_{*1} & \cdots & \tilde{s}^T P_{*g} & | & \tilde{s}^T P_{22}e \\
\end{pmatrix}_{(g+1) \times (g+1)}
\]
Since we use the old stationary vector $\phi^T$ to find the updated stationary vector $\pi^T$

$$\Rightarrow \pi^T \approx \phi^T$$

$$A \approx \tilde{A} = \begin{pmatrix} P_{11} & P_{12} \bar{e} \\ \bar{e}^T P_{21} & \bar{e}^T P_{22} \bar{e} \end{pmatrix}$$

$$\alpha^T \approx \tilde{\alpha}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_g, \tilde{\alpha}_{g+1})$$

$$\pi^T \approx \tilde{\pi}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_g, \tilde{\alpha}_{g+1} \bar{e}^T)$$

(not bad approximation)
Approximate Aggregation Algorithm

1. Partition states of updated as $S = G \cup \bar{G}$ and reorder $P =$

$$
\begin{pmatrix}
G & \bar{G} \\
G & P_{11} \\
\bar{G} & P_{12} \\
\end{pmatrix}
$$

2. Set $\bar{\phi}^T = (\phi_{g+1}, \phi_{g+2}, \ldots, \phi_n)$

3. Set $\tilde{s}^T = \bar{\phi}^T / \bar{\phi}^T e$ where $e$ is a column of ones

4. Set $\tilde{A} = \begin{pmatrix}
P_{11} & P_{12}e \\
\tilde{s}^T P_{21} & 1 - \tilde{s}^T P_{21} \\
\end{pmatrix}_{(g+1) \times (g+1)}$

5. Find $\tilde{\alpha}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_g, \tilde{\alpha}_{g+1})$

6. Set $\tilde{\pi}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_g | \bar{\phi}^T)$
Initiation

- Partition the states of the updated chain as $S = G \cup \bar{G}$
- $\bar{\phi}^T \leftarrow$ the component from $\phi^T$ corresponding to the states in $\bar{G}$
- $s^T \leftarrow \bar{\phi}^T / (\bar{\phi}^T e)$

Iterate until convergence

1. $\tilde{A} \leftarrow \begin{pmatrix} P_{11} & P_{12}e \\ \tilde{s}^T P_{21} & 1 - \tilde{s}^T P_{21} \end{pmatrix}$
2. $\alpha^T \leftarrow (\alpha_1, \alpha_2, \ldots, \alpha_g, \alpha_{g+1})$
3. $\chi^T \leftarrow (\alpha_1, \alpha_2, \ldots, \alpha_g | \alpha_{g+1}s^T)$
4. $\psi^T \leftarrow \chi^T P$ (Move the iterates off the fixed point $\rightarrow$ the convergence.)
5. If $\|\psi^T - \chi^T\| < \tau$ for a given tolerance $\tau$, then quit
   else $s^T \leftarrow \psi^T / \psi^T e$ go back to step 1.
How to determine the partition for the updating problem?

Rules of the partition

- Put the updated states into G.
- Put the nearest states to the update into G.
- Put the states which have a large stationary probability into $G$. (slow converging states shown by Kamvar et al.)
### Experiments

<table>
<thead>
<tr>
<th>Topic</th>
<th>Number of Nodes</th>
<th>Number of Links</th>
<th>Total number of nodes</th>
<th>Total number of links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetics</td>
<td>2,952</td>
<td>6,485</td>
<td>3,200</td>
<td>6,900</td>
</tr>
<tr>
<td>California</td>
<td>9,664</td>
<td>16,150</td>
<td>10,300</td>
<td>17,000</td>
</tr>
</tbody>
</table>

- Adding and deleting some nodes
- Adding and deleting some links
- Stopping criterion $\|\psi^T - \chi^T\|_1 < 10^{-10}$
Iterative aggregation algorithm is about 5 times faster than Power method.

<table>
<thead>
<tr>
<th></th>
<th>随着时间</th>
<th>时长（秒）</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>163</td>
<td>2.16</td>
</tr>
<tr>
<td>50</td>
<td>19</td>
<td>0.483</td>
</tr>
<tr>
<td>100</td>
<td>19</td>
<td>0.456</td>
</tr>
<tr>
<td>250</td>
<td>17</td>
<td>0.276</td>
</tr>
<tr>
<td>500</td>
<td>9</td>
<td>0.313</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>0.319</td>
</tr>
</tbody>
</table>

*Power Method* | 165 | 1.45
Iterative aggregation algorithm is about 6 times faster than Power method.

| $g = |G|$ | Iterations | Time (sec) |
|--------|------------|------------|
| 10     | 170        | 7.75       |
| 50     | 75         | 3.56       |
| 100    | 57         | 3.75       |
| 250    | 51         | 2.59       |
| 500    | 34         | 2.01       |
| 1000   | 19         | 1.03       |
| 2000   | 10         | .997       |
| 3000   | 7          | 1.17       |
| 4000   | 7          | 1.22       |
| 5000   | 7          | 1.56       |

**Power Method**

176  5.87
• Size of G near optimal stay around the right of the bend.
**Conclusion And Future Work**

**Conclusion**

Appropriate partitions can greatly speed up Stationary vector computation.

**Future work**

- Combine the Iterative Aggregation with other methods to increase the speed of entire process.
- Find a concrete way to partition $S = G \cup \bar{G}$.
- Investigate the relationship between $|\lambda_2|$ of the S and $|G|$.
- Work with real life data.