Power Iteration Clustering

Frank Lin and William W. Cohen
Presented by Minhua Chen

March 10, 2011
1. Power Iteration Method
2. Spectral Clustering
3. Power Iteration Clustering
4. Result
Power Iteration Method

For any matrix $W$ with eigenvalue decomposition $W = E \Lambda E^{-1}$, the dominant eigenvector $e_1$ can be computed via iteration $v(t) = \eta(t)Wv(t-1)$ ($\eta(t)$ is a normalizing constant).

Proof: By induction, $v(t) = \eta(t)Wv(t-1) = \ldots = \tilde{\eta}(t)W^tv(0) \propto W^tv(0)$ where $\tilde{\eta}(t) = \prod_{l=1}^{t} \eta(l)$ is another scalar constant.

Since $E$ is a basis in $\mathbb{R}^n$, the initial value $v(0)$ can be expressed as $v(0) = \sum_{i=1}^{n} c_i e_i$. Then

$$W^tv(0) = \sum_{i=1}^{n} c_i W^te_i = \sum_{i=1}^{n} c_i \lambda_i^t e_i = c_1 \lambda_1^t \left( e_1 + \sum_{i=2}^{n} \frac{c_i}{c_1} \left( \frac{\lambda_i}{\lambda_1} \right)^t e_i \right).$$

Since $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$, hence $(\frac{\lambda_i}{\lambda_1})^t \to 0$ and $v(t) \propto e_1$ as $t \to \infty$. Thus $v(t)$ converges to the dominant eigenvector.
Spectral Clustering

1. Given the data matrix $X = [x_1, x_2, \cdots, x_n]_{p \times n}$, an affinity matrix $A \in \mathbb{R}^{n \times n}$ is defined as $A_{ij} = s(x_i, x_j)$ where $s(\cdot, \cdot)$ is a similarity function.

2. Define the normalized affinity matrix as $W = \text{diag}^{-1}(A \cdot 1)A$. Then the top eigenvectors of $W$ give an embedding of the original data $X$. Clustering analysis is further applied to the embedded data.

3. The Eigen-decomposition of $W$ is expressed as $W = E \Lambda E^{-1}$ with $E = [e_1, e_2, \cdots, e_n]$ and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ ($|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$).

4. Since $W \cdot 1 = 1$, $(\lambda_1 = 1, e_1 = 1)$ is always an eigen-pair of $W$. So what is really useful for embedding is $[e_2, e_3, \cdots, e_k]_{n \times (k-1)}$, the rows being the embedding of the original $n$ data points. The dimensionality of the data is reduced from $p$ to $k - 1$. 
Power Iteration Clustering

1. The Spectral Clustering algorithm is summarized as:
   1) Construct the normalized affinity matrix $\mathbf{W}$.
   2) Find top $k$ eigenvectors of $\mathbf{W}$ as $[e_1, e_2, \cdots, e_k]$.
   3) Cluster on $[e_2, e_3, \cdots, e_k]_{n \times (k-1)}$.

2. The dominant eigenvector $e_1 = 1$ seems to be useless for clustering. However, this paper turns the useless thing into a very useful thing.

3. What if we apply the Power Iteration Method to $\mathbf{W}$? Of course as $t \to \infty$, $v^{(t)} \propto e_1 = 1$, but if we look closer...
From the proof of the Power Iteration Method:
\[ \mathbf{v}^{(t)} \propto c_1 \lambda_1^t \left( \mathbf{e}_1 + \sum_{i=2}^{k} \frac{c_i}{c_1} \left( \frac{\lambda_i}{\lambda_1} \right)^t \mathbf{e}_i + \sum_{j=k+1}^{n} \frac{c_j}{c_1} \left( \frac{\lambda_j}{\lambda_1} \right)^t \mathbf{e}_j \right). \]

Typically, \( k \) is selected such that
\[ |\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_k| \gg |\lambda_{k+1}| \geq \cdots \geq |\lambda_n|. \]
Hence
\[ \left( \frac{\lambda_i}{\lambda_1} \right)^t \ll \left( \frac{\lambda_i}{\lambda_1} \right)^t \text{ for } i = 2, 3, \ldots, k; j = k + 1, k + 2, \ldots, n. \]
This means that components in the noise subspace \([\mathbf{e}_{k+1}, \mathbf{e}_{k+2}, \ldots, \mathbf{e}_n]\) will diminish much faster than that in the signal subspace \([\mathbf{e}_2, \mathbf{e}_3, \ldots, \mathbf{e}_k]\).

We can stop the iteration early, when the noise subspace vanishes while the signal subspace does not yet.
If \( t \) is chosen properly in this way, then
\[ \mathbf{v}^{(t)} \propto c_1 \lambda_1^t \left( \mathbf{e}_1 + \sum_{i=2}^{k} \frac{c_i}{c_1} \left( \frac{\lambda_i}{\lambda_1} \right)^t \mathbf{e}_i \right) \text{ which is a weighted combination of the top } k \text{ eigenvectors.} \]
Power Iteration Clustering (continued 2)

**Algorithm 1** The PIC algorithm

*Input*: A row-normalized affinity matrix $W$ and the number of clusters $k$  
Pick an initial vector $v^0$  
repeat
  $v^{t+1} \leftarrow \frac{Wv^t}{\|Wv^t\|_1}$ and $\delta^{t+1} \leftarrow |v^{t+1} - v^t|$.  
  Increment $t$  
until $|\delta^t - \delta^{t-1}| \approx 0$

Use $k$-means to cluster points on $v^t$  
*Output*: Clusters $C_1, C_2, ..., C_k$

---

**Figure 1.** Clustering result and the embedding provided by $v^t$ for the 3Circles dataset. In (b) through (d), the value of each component of $v^t$ is plotted against its index. Plots (b) through (d) are re-scaled so the largest value is always at the very top and the minimum value at the very bottom, and *scale* is the maximum value minus the minimum value.
Question: Why this vector \( \mathbf{v}^{(t)} \) works for clustering?

Answer: Pair-wise distance of the embedding is preserved in a similar way as that in spectral clustering.

\[
\text{pic}^{(t)}(a, b) \triangleq |\mathbf{v}^{(t)}(a) - \mathbf{v}^{(t)}(b)| \propto |c_1 \lambda_1^t| \cdot |(e_1(a) - e_1(b)) + \\
\sum_{i=2}^{k} c_i (\frac{\lambda_i}{\lambda_1})^t (e_i(a) - e_i(b)) + \\
\sum_{j=k+1}^{n} c_j (\frac{\lambda_j}{\lambda_1})^t (e_j(a) - e_j(b))| \\
\propto |c_1 \lambda_1^t| \cdot |\sum_{i=2}^{k} c_i (\frac{\lambda_i}{\lambda_1})^t (e_i(a) - e_i(b))|
\]

This expression is similar to the pair-wise distance of Spectral Clustering:

\[
\text{spec}(a, b) = \sqrt{\sum_{i=2}^{k} (e_i(a) - e_i(b))^2}.
\]

The weighting factor \((\frac{\lambda_i}{\lambda_1})^t\) in the proposed approach is reasonable and improves performance for spectral methods.
Table 1. Clustering performance of PIC and spectral clustering algorithms on several real datasets. For all measures a higher number means better clustering. Bold numbers are the highest in its row.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>Purity</th>
<th>NMI</th>
<th>RI</th>
<th>Purity</th>
<th>NMI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3</td>
<td>0.6733</td>
<td>0.7235</td>
<td>0.7779</td>
<td>0.7667</td>
<td>0.6083</td>
<td>0.7978</td>
</tr>
<tr>
<td>PenDigits01</td>
<td>2</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>PenDigits17</td>
<td>2</td>
<td>0.7550</td>
<td>0.2066</td>
<td>0.6301</td>
<td>0.7550</td>
<td>0.2043</td>
<td>0.6301</td>
</tr>
<tr>
<td>PolBooks</td>
<td>3</td>
<td>0.8476</td>
<td>0.5745</td>
<td>0.8447</td>
<td>0.8286</td>
<td>0.5422</td>
<td>0.8329</td>
</tr>
<tr>
<td>UBMCBlog</td>
<td>2</td>
<td>0.9530</td>
<td>0.7488</td>
<td>0.9104</td>
<td>0.9530</td>
<td>0.7375</td>
<td>0.9104</td>
</tr>
<tr>
<td>AGBlog</td>
<td>2</td>
<td>0.5205</td>
<td>0.0060</td>
<td>0.5006</td>
<td>0.5205</td>
<td>0.0006</td>
<td>0.5007</td>
</tr>
<tr>
<td>20ngA</td>
<td>2</td>
<td>0.9600</td>
<td>0.7594</td>
<td>0.9232</td>
<td>0.9600</td>
<td>0.7594</td>
<td>0.9232</td>
</tr>
<tr>
<td>20ngB</td>
<td>2</td>
<td>0.5050</td>
<td>0.0096</td>
<td>0.5001</td>
<td>0.5525</td>
<td>0.0842</td>
<td>0.5055</td>
</tr>
<tr>
<td>20ngC</td>
<td>3</td>
<td>0.6183</td>
<td>0.3295</td>
<td>0.6750</td>
<td>0.6317</td>
<td>0.3488</td>
<td>0.6860</td>
</tr>
<tr>
<td>20ngD</td>
<td>4</td>
<td>0.4750</td>
<td>0.2385</td>
<td>0.6312</td>
<td>0.5150</td>
<td>0.2959</td>
<td>0.6820</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.7308</td>
<td>0.4596</td>
<td>0.7393</td>
<td>0.7483</td>
<td>0.4581</td>
<td>0.7469</td>
</tr>
</tbody>
</table>