Power Iteration Clustering

Frank Lin and William W. Cohen
Presented by Minhua Chen

March 10, 2011

Power Iteration Method

For any matrix $\mathbf{W}$ *with eigenvalue decomposition* $\mathbf{W} = \mathbf{E}\Lambda\mathbf{E}^{-1}$, *the dominant eigenvector* $\mathbf{e}_1$ *can be computed via iteration* $\mathbf{v}^{(t)} = \eta^{(t)}\mathbf{W}\mathbf{v}^{(t-1)}$ ($\eta^{(t)}$ *is a normalizing constant).*

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

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

$$\mathbf{W}^t\mathbf{v}^{(0)} = \sum_{i=1}^{n} c_i\mathbf{W}^t\mathbf{e}_i = \sum_{i=1}^{n} c_i\lambda_i^t\mathbf{e}_i = c_1\lambda_1^t\left(\mathbf{e}_1 + \sum_{i=2}^{n}\frac{c_i}{c_1}(\frac{\lambda_i}{\lambda_1})^t\mathbf{e}_i\right).$$

Since $|\lambda|_1 > |\lambda|_2 \geq |\lambda|_3 \geq \cdots \geq |\lambda_n|$, hence $(\frac{\lambda_i}{\lambda_1})^t \to 0$ and $\mathbf{v}^{(t)} \propto \mathbf{e}_1$ as $t \to \infty$. Thus $\mathbf{v}^{(t)}$ converges to the dominant eigenvector.

Spectral Clustering

1. Given the data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]_{p \times n}$, an affinity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is defined as $A_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$ where $s(\cdot, \cdot)$ is a similarity function.
2. Define the normalized affinity matrix as $\mathbf{W} = \mathrm{diag}^{-1}(\mathbf{A} \cdot \mathbf{1})\mathbf{A}$. Then the top eigenvectors of $\mathbf{W}$ give an embedding of the original data $\mathbf{X}$. Clustering analysis is further applied to the embedded data.
3. The Eigen-decomposition of $\mathbf{W}$ is expressed as $\mathbf{W} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^{-1}$ with $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n]$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n) \quad (|\lambda|_1 > |\lambda|_2 \geq |\lambda|_3 \geq \cdots \geq |\lambda_n|)$.
4. Since $\mathbf{W} \cdot \mathbf{1} = \mathbf{1}$, $(\lambda_1 = 1, \mathbf{e}_1 = \mathbf{1})$ is always an eigen-pair of $\mathbf{W}$. So what is really useful for embedding is $[\mathbf{e}_2, \mathbf{e}_3, \cdots, \mathbf{e}_k]_{n \times (k-1)}$, 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 $[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k]$.
3)	Cluster on $[\mathbf{e}_2, \mathbf{e}_3, \cdots, \mathbf{e}_k]_{n \times (k-1)}$.

2. The dominant eigenvector $\mathbf{e}_1 = \mathbf{1}$ 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 $t \to \infty$, $\mathbf{v}^{(t)} \propto \mathbf{e}_1 = \mathbf{1}$, but if we look closer ...

Power Iteration Clustering (continued 1)

1. From the proof of the Power Iteration Method:
   $\mathbf{v}^{(t)} \propto c_1 \lambda_1^t \left(\mathbf{e}_1 + \sum_{i=2}^{k} \frac{c_i}{c_1} (\frac{\lambda_i}{\lambda_1})^t \mathbf{e}_i + \sum_{j=k+1}^{n} \frac{c_j}{c_1} (\frac{\lambda_j}{\lambda_1})^t \mathbf{e}_j \right)$.
2. 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 $(\frac{\lambda_j}{\lambda_1})^t \ll (\frac{\lambda_i}{\lambda_1})^t$ for $i = 2, 3, \cdots, k; j = k+1, k+2, \cdots, n$. This means that components in the noise subspace $[\mathbf{e}_{k+1}, \mathbf{e}_{k+2}, \cdots, \mathbf{e}_n]$ will diminish much faster than that in the signal subspace $[\mathbf{e}_2, \mathbf{e}_3, \cdots, \mathbf{e}_k]$.
3. 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 $\mathbf{v}^{(t)} \propto c_1 \lambda_1^t \left(\mathbf{e}_1 + \sum_{i=2}^{k} \frac{c_i}{c_1} (\frac{\lambda_i}{\lambda_1})^t \mathbf{e}_i \right)$ which is a weighted combination of the top k 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 $\mathbf{v}^0$

repeat

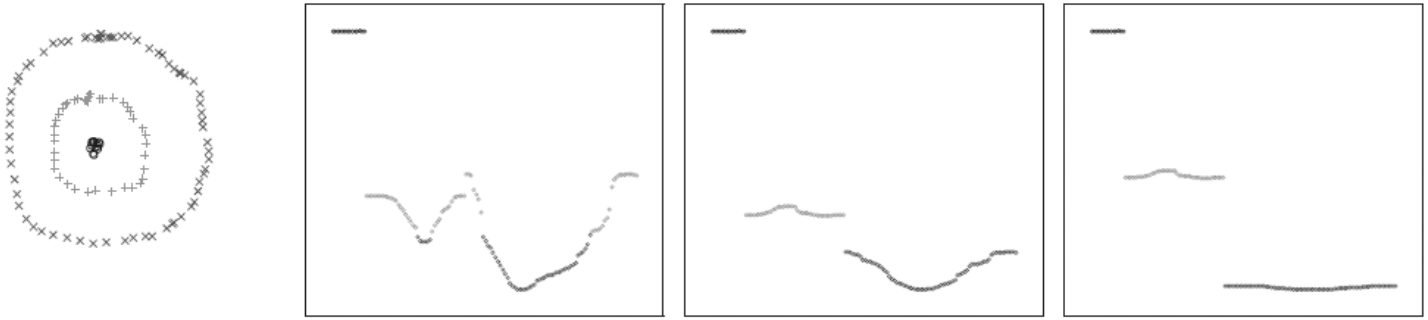
Set $\mathbf{v}^{t+1} \leftarrow \frac{W\mathbf{v}^t}{\|W\mathbf{v}^t\|_1}$ and $\delta^{t+1} \leftarrow |\mathbf{v}^{t+1} - \mathbf{v}^t|$.

Increment t

until $|\delta^t - \delta^{t-1}| \simeq \mathbf{0}$

Use k-means to cluster points on $\mathbf{v}^t$

Output: Clusters $C_1, C_2, ..., C_k$



(a) 3Circles PIC result (b) $t = 50$, scale $= 0.01708$ (c) $t = 400$, scale $= 0.01066$ (d) $t = 1000$, scale $= 0.00786$

Figure 1. Clustering result and the embedding provided by $\mathbf{v}^t$ for the 3Circles dataset. In (b) through (d), the value of each component of $\mathbf{v}^t$ 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.

Power Iteration Clustering (continued 3)

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.

$$\mathrm{pic}^{(t)}(a,b) \triangleq |v^{(t)}(a) - v^{(t)}(b)| \propto |c_1\lambda_1^t| \cdot |(e_1(a) - e_1(b)) +$$

$$\sum_{i=2}^{k} \frac{c_i}{c_1}(\frac{\lambda_i}{\lambda_1})^t(e_i(a) - e_i(b)) + \sum_{j=k+1}^{n} \frac{c_j}{c_1}(\frac{\lambda_j}{\lambda_1})^t(e_j(a) - e_j(b))|$$

$$\propto |c_1\lambda_1^t| \cdot |\sum_{i=2}^{k} \frac{c_i}{c_1}(\frac{\lambda_i}{\lambda_1})^t(e_i(a) - e_i(b))|$$

This expression is similar to the pair-wise distance of Spectral Clustering: $\mathrm{spec}(a,b) = \sqrt{\sum_{i=2}^{k}(e_i(a) - e_i(b))^2}$.
The weighting factor $(\frac{\lambda_i}{\lambda_1})^t$ in the proposed approach is reasonable and improves performance for spectral methods.

Result

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.

		NCut			NJW			PIC		
Dataset	**k**	**Purity**	**NMI**	**RI**	**Purity**	**NMI**	**RI**	**Purity**	**NMI**	**RI**
Iris	3	0.6733	0.7235	0.7779	0.7667	0.6083	0.7978	**0.9800**	**0.9306**	**0.9741**
PenDigits01	2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
PenDigits17	2	**0.7550**	**0.2066**	**0.6301**	**0.7550**	0.2043	**0.6301**	**0.7550**	**0.2066**	**0.6301**
PolBooks	3	0.8476	0.5745	0.8447	0.8286	0.5422	0.8329	**0.8667**	**0.6234**	**0.8603**
UBMCBlog	2	**0.9530**	**0.7488**	**0.9104**	**0.9530**	0.7375	**0.9104**	0.9480	0.7193	**0.9014**
AGBlog	2	0.5205	0.0060	0.5006	0.5205	0.0006	0.5007	**0.9574**	**0.7465**	**0.9185**
20ngA	2	**0.9600**	**0.7594**	**0.9232**	**0.9600**	**0.7594**	**0.9232**	**0.9600**	**0.7594**	**0.9232**
20ngB	2	0.5050	0.0096	0.5001	0.5525	0.0842	0.5055	**0.8700**	**0.5230**	**0.7738**
20ngC	3	0.6183	0.3295	0.6750	0.6317	0.3488	0.6860	**0.6933**	**0.4450**	**0.7363**
20ngD	4	0.4750	0.2385	0.6312	0.5150	0.2959	0.6820	**0.5825**	**0.3133**	**0.7149**
Average		0.7308	0.4596	0.7393	0.7483	0.4581	0.7469	**0.8613**	**0.6267**	**0.8433**