**Summary:** Power Iteration Clustering, by Frank Lin (CMU grad, co-founder of EnFind) and William W. Cohen (Professor, Machine Learning Department, CMU)

**Main Point:** Power iteration clustering uses power iteration to find a vector that is a linear combination of the $k$ eigenvectors corresponding to the $k$ smallest eigenvalues $\lambda_1, \ldots, \lambda_k$ of the normalized Laplacian matrix $D^{-1}L = I - D^{-1}W$, where $\lambda_k$ is assumed to be significantly greater than $\lambda_{k+1}$, and all $\lambda_i$ for $i$ in $\{2, \ldots, k\}$ are assumed to be sufficiently close to $\lambda_1$. The vector found by Lin and Cohen’s version of power iteration is viewed as a 1-dimensional embedding of the data points and is used to cluster data points into $k$ clusters, in a manner similar to spectral clustering.

Lin and Cohen begin with the simple observation that given a similarity matrix $S$ describing pairwise similarities $s_{ij}$ between data points $x_i$ and $x_j$, we can find the $k$ eigenvectors corresponding to the $k$ smallest eigenvalues of the row-normalized Laplacian matrix $D^{-1}L$ by finding the $k$ eigenvectors corresponding to the $k$ largest eigenvalues of $I - D^{-1}L$. This is due to the simple fact that, for any matrix $A$ and eigenvector $v$, we have: $Av = \lambda v \Rightarrow (I - A)v = v - Av = v - \lambda v = (1 - \lambda)v$, so if $\lambda$ is an eigenvalue of $A$ with eigenvector $v$, then $(1 - \lambda)$ is an eigenvalue of $(I - A)$ with the same eigenvector $v$. Note that for the unnormalized graph Laplacian $D^{-1}L = D^{-1}(D - W) = I - D^{-1}W$, so $I - D^{-1}L = D^{-1}W$. Therefore, the $k$ smallest eigenvalues of $D^{-1}L$ are the $k$ largest eigenvalues of $D^{-1}W$, with the same eigenvectors.

Recall that power iteration simply multiplies a diagonalizable matrix $A$ by a random initial vector $v$, then normalizes this product, until the difference in two iterations of this process is very small. In other words, we have at each step $v^{t+1} = \frac{Av^t}{\|Av^t\|}$ until $v^{t+1} - v^t$ is close to 0. $v^t$ converges to the principal eigenvector of $A$. To see this, write $v^0$ as a linear combination of the eigenvectors $\mu$ of $A$. Then: $Av^0 = A(c_1\mu_1 + \cdots + c_n\mu_n) = c_1\lambda_1\mu_1 + \cdots + c_n\lambda_n\mu_n$ where the $c_i$'s are constants, and without loss of generality, assume $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$. For a connected, undirected graph with unnormalized graph Laplacian matrix $L$, the eigenvalues of $I - D^{-1}L$ will have this property. In addition:

$$\|Av^0\| = \|c_1\lambda_1\mu_1 + \cdots + c_n\lambda_n\mu_n\|$$

Thus:

$$v^{t+1} = \frac{Av^t}{\|Av^t\|} = \frac{A^tv^0}{\|A^tv^0\|} = \frac{c_1\lambda_1^t\mu_1 + \cdots + c_n\lambda_n^t\mu_n}{\|c_1\lambda_1^t\mu_1 + \cdots + c_n\lambda_n^t\mu_n\|}$$

Note that

$$A^tv^0 = c_1\lambda_1^t\mu_1 + \cdots + c_n\lambda_n^t\mu_n = c_1\lambda_1^t \left( \mu_1 + c_2 \left( \frac{\lambda_2}{\lambda_1} \right)^t \mu_2 + \cdots + c_n \left( \frac{\lambda_n}{\lambda_1} \right)^t \mu_n \right)$$

So as $t \to \infty$, $A^tv^0 \to c_1\lambda_1^t\mu_1$, in other words $A^tv^0$ converges to (a multiple of) the principal eigenvector $\mu_1$ of $A$. The normalization at each step is used to
ensure that \(|v^{t+1}| = 1\) (because for any vector \(v\), \(\|Av\|/\|v\| = \sqrt{v^T A v}/\|v\| = 1\)). Thus \(1 = \|Av\|/\|v\| = |\lambda_1|\), so that we ensure that \(v^{t+1} \rightarrow \lambda_1 \mu_1\) as \(t \rightarrow \infty\), and we can be sure that the entries in \(v^t\) won’t “blow up” to become too large.

Lin and Cohen’s insight is the observation that while power iteration run to convergence will give us the principal eigenvector for a matrix, running power iteration on \(D^{-1}W\) to what they call *local convergence* will give us a vector that is a linear combination of the \(k\) eigenvectors corresponding to the \(k\) smallest eigenvalues \(\lambda_1, \ldots, \lambda_k\) of \(D^{-1}L\). Assuming that the gap between \(\lambda_k\) and \(\lambda_{k+1}\) is large, and that the gap between \(\lambda_1\) and \(\lambda_2, \ldots, \lambda_k\) is not very large, as has been shown to be the case for clusters that are well-separated (see Luxburg’s *Tutorial on Spectral Clustering*, for example) Lin and Cohen note that power iteration on \(D^{-1}W\) will converge “locally” before converging globally. They define local convergence as the point at which the \((k+1)\)th to \(n\)th eigenvalues are no longer influencing the term \(v^{t+1}\) in power iteration. At this point in power iteration, we have:

\[
\frac{v^{t+1}}{c_1 \lambda_1} \approx \mu_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^t \mu_2 + \cdots + \left(\frac{\lambda_k}{\lambda_1}\right)^t \mu_k
\]

In other words, we are assuming that the terms \(\left(\frac{\lambda_j}{\lambda_1}\right)^t\) for \(j > k\) are so small that they are having a negligible effect on \(v^{t+1}\).

Furthermore, Lin and Cohen assume that at this point, the vector \(v^{t+1}\) has converged “locally” to the clusters represented by an embedding into the space spanned \(\mu_1, \ldots, \mu_k\) in spectral clustering. Local convergence is detected by recognizing the point at which \(\frac{|v^{t+1} - v^t|}{|v^t - v^{t-1}|} \approx 0\). By the reasoning above, if the \(k+1\) through \(n\)th eigenvectors have been nearly cancelled out, and have very little effect on the convergence of \(v^t\), then in two successive iterations the only vectors affecting the change in \(v\) are the eigenvectors \(\mu_1, \ldots, \mu_k\). If in addition the eigenvalues \(\lambda_2, \ldots, \lambda_k\) are close to \(\lambda_1\), then convergence to \(\mu_1\) will become nearly constant. Thus, the point at which the “velocity” of the the convergence, which Lin and Cohen define as \(v^{t+1} - v^t\), becomes nearly constant, i.e. when \(\frac{|v^{t+1} - v^t|}{|v^t - v^{t-1}|} \approx 0\), is the point at which \(v^t\) has converged “locally” to the \(k\) eigenvectors corresponding to the \(k\) smallest eigenvalues of \(D^{-1}L\).

Now, to use the locally-converged \(v^t\) to cluster the data points \(x_1, \ldots, x_n\), Lin and Cohen observe that the entries in \(v^t\) should be approximately piecewise-constant, which again comes from the assumption that the clusters are well separated. Then, the eigenvectors \(\mu_1, \ldots, \mu_k\) should approximately equal multiples of the “indicator vectors” as defined by Luxburg in *A Tutorial on Spectral Clustering*. Thus, the linear combination of these vectors will give an approximately piecewise-constant vector, where each piece corresponds to one of the clusters, assuming that the multiple of the indicator vector is not the same for two different vectors. Although no proof is given that this is indeed the case,
I think that as the number of clusters $k$ grows, it becomes increasingly difficult to separate the piecewise-constant pieces, due to a greater chance that two eigenvectors will have multiples that are close together when $k$ is large (since $v$ is normalized, and all values will be between 0 and 1, it’s just common sense that the more pieces there are in a piecewise constant vector, the fewer ways to arrange them so that the gap between pieces remains larger than some value $\epsilon$). However, I have no proof either.

Lin and Cohen show that their algorithm successfully clusters real data sets of various scales, with $2 \leq k \leq 4$ clusters. PIC outperforms the Implicitly Restarted Arnoldi Method used in normalized spectral clustering with $D^{-1}L$ as the normalized Laplacian matrix, exhibiting runtimes up to 300 times faster on data sets of up to 1222 data points. On a larger scale, PIC finishes in approximately 3 seconds on a set of 100,000 vertices. In terms of normalized mutual information, purity and the rand index, PIC is consistently competitive with or better than spectral methods using either the normalized graph Laplacian $L_{rw} = D^{-1}L$ or $L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ on identical data sets. Lin and Cohen note that starting power iteration with a normalized “degree vector” where $v_i = \sum_j W_{ij} \cdot \text{Vol}(W)$ results in faster local convergence than a random initial vector. They note that this may be because this normalization will give more weight to high-degree nodes in the graph, “which means that, in the average view, values will be distributed more evenly and quickly, leading to faster local convergence.” Lin and Cohen also note that when $k$ is “sufficiently large”, multiple random $v_0$’s are used to compute $\alpha$ different solutions, and $k$ means is then applied to cluster these solutions in an $\alpha$-dimensional space.