Approximate Spectral Clustering via Randomized Sketching

Christos Boutsidis
Yahoo! Labs, New York

Joint work with Alex Gittens (Ebay), Anju Kambadur (IBM)
The big picture: “sketch” and solve

\[ A = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \]

\[ \tilde{A} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \]

Tradeoff: Speed (depends on the size of \( \tilde{A} \)) with accuracy (quantified by the parameter \( \varepsilon > 0 \)).
1 **Sampling:** $A \rightarrow \tilde{A}$ by picking a subset of the columns of $A$

\[
\begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}
\quad \rightarrow 
\begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}
\]

2 **Linear sketching:** $A \rightarrow \tilde{A} = AR$ for some matrix $R$.

3 **Non-linear sketching:** $A \rightarrow \tilde{A}$ (no linear relationship).
Sketching techniques (low level)

1 Sampling:
- **Importance sampling**: randomized sampling with probabilities proportional to the norms of the columns of $A$ [Frieze, Kannan, Vempala, FOCS 1998], [Drineas, Kannan, Mahoney, SISC 2006].
- **Subspace sampling**: randomized sampling with probabilities proportional to the norms of the rows of the matrix $V_k$ containing the top $k$ right singular vectors of $A$ (leverage-scores sampling) [Drineas, Mahoney, Muthukrishnan, SODA 2006].
- **Deterministic sampling**: deterministically selecting rows from $V_k$ - equivalently columns from $A$ [Batson, Spielman, Srivastava, STOC 2009], [Boutsidis, Drineas, Magdon-Ismail, FOCS 2011].

2 Linear sketching:
- **Random Projections**: post-multiply $A$ with a random gaussian matrix [Johnson, Lindenstrauss 1982].
- **Fast Random Projections**: post-multiply $A$ with an FFT-type random matrix [Ailon, Chazelle 2006].
- **Sparse Random Projections**: post-multiply $A$ with a sparse matrix [Clarkson, Woodruff STOC 2013].

3 Non-linear sketching:
- **Frequent Directions**: SVD-type transform. [Liberty, KDD ’13], [Ghashami, Phillips, SODA ’14].
- Other non-linear dimensionality reduction methods such as LLE, ISOMAP etc.
Problems

Linear Algebra:

1. **Matrix Multiplication** [Drineas, Kannan, Rudelson, Virshynin, Woodruff, Ipsen, Liberty, and others]
2. **Low-rank Matrix Approx.** [Tygert, Tropp, Clarkson, Candes, B., Despande, Vempala, and others]
3. **Element-wise Sparsification** [Achlioptas, McSherry, Kale, Drineas, Zouzias, Liberty, Karnin, and others]
4. **Least-squares** [Mahoney, Muthukrishnan, Dasgupta, Kumar, Sarlos, Rokhlin, Boutsidis, Avron, and others]
5. **Linear Equations with SDD matrices** [Spielman, Teng, Koutis, Miller, Peng, Orecchia, Kelner, and others]
6. **Determinant of SPSD matrices** [Barry, Pace, B., Zouzias and others]
7. **Trace of SPSD matrices** [Avron, Toledo, Bekas, Roosta-Khorasani, Uri Ascher, and others]

Machine Learning:

1. **Canonical Correlation Analysis** [Avron, B., Toledo, Zouzias]
2. **Kernel Learning** [Rahimi, Recht, Smola, Sindhwani and others]
3. **$k$-means Clustering** [B., Zouzias, Drineas, Magdon-Ismail, Mahoney, Feldman, and others]
4. **Spectral Clustering** [Gittens, Kambadur, Boutsidis, Strohmer and others]
5. **Spectral Graph Sparsification** [Batson, Spielman, Srivastava, Koutis, Miller, Peng, Kelner, and others]
6. **Support Vector Machines** [Paul, B., Drineas, Magdon-Ismail and others]
7. **Regularized least-squares classification** [Dasgupta, Drineas, Harb, Josifovsky, Mahoney]
What approach should we use to cluster these data?

Answer: $k$-means clustering
$k$-means optimizes the “right” metric over this space

- $\mathcal{P} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^d$. number of clusters $k$.
- $k$-partition of $\mathcal{P}$: a collection $S = \{S_1, S_2, \ldots, S_k\}$ of sets of points.
- For each set $S_j$, let $\mu_j \in \mathbb{R}^d$ be its centroid.
- $k$-means objective function: $\mathcal{F}(\mathcal{P}, S) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mu(\mathbf{x}_i)\|_2^2$
- Find the best partition:

$$S_{opt} = \arg \min_S \mathcal{F}(\mathcal{P}, S).$$
What approach should we use to cluster these data?

Answer: \( k \)-means will fail miserably. What else?
Spectral Clustering: Transform the data into a space where $k$-means would be useful.

1-d representation of points from the first dataset in previous picture (this is an eigenvector from an appropriate graph).
Spectral Clustering: the graph theoretic perspective

- $n$ points \{\(x_1, x_2, \ldots, x_n\)\} in $d$-dimensional space.
- $G(V, E)$ is the corresponding graph with $n$ nodes.
- Similarity matrix $W \in \mathbb{R}^{n \times n}$ \(W_{ij} = e^{-\frac{\|x_i-x_j\|^2}{\sigma}}\) (for $i \neq j$); $W_{ii} = 0$.
- Let $k$ be the number of clusters.

**Definition**

Let $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ and $k = 2$ are given. Find subgraphs of $G$, denoted as $A$ and $B$, to minimize:

\[
\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)},
\]

where $\text{cut}(A, B) = \sum_{x_i \in A, x_j \in B} W_{ij}$; and

\[
\text{assoc}(A, V) = \sum_{x_i \in A, x_j \in V} W_{ij}; \quad \text{assoc}(B, V) = \sum_{x_i \in B, x_j \in V} W_{ij}.
\]
For any $G, A, B$ and partition vector $y \in \mathbb{R}^n$ with $+1$ to the entries corresponding to $A$ and $-1$ to the entries corresponding to $B$ it is:

$$4 \cdot \text{Ncut}(A, B) = y^T(D - W)y / (y^TDy).$$

Here, $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix of degree nodes: $D_{ii} = \sum_j W_{ij}$.

**Definition**

Given graph $G$ with $n$ nodes, adjacency matrix $W$, and degrees matrix $D$ find $y \in \mathbb{R}^n$:

$$y = \arg\min_{y \in \mathbb{R}^n, y^TD1_n} \frac{y^T(D - W)y}{y^TDy}.$$
Spectral Clustering: Algorithm for $k$-partitioning

Cluster $n$ points $\{x_1, x_2, ..., x_n\}$ into $k$ clusters

1. Construct the similarity matrix $W \in \mathbb{R}^{n \times n}$ as $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}$ (for $i \neq j$) and $W_{ii} = 0$.

2. Construct $D \in \mathbb{R}^{n \times n}$ as the diagonal matrix of degree nodes: $D_{ii} = \sum_j W_{ij}$.

3. Construct $\tilde{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$.

4. Find the largest $k$ eigenvectors of $\tilde{W}$ and assign them as columns to a matrix $Y \in \mathbb{R}^{n \times k}$.

5. Apply $k$-means clustering on the rows of $Y$, and cluster the original points accordingly.

In a nutshell, compute the top $k$ eigenvectors of $\tilde{W}$ and then apply $k$-means on the rows of the matrix containing those eigenvectors.
Cluster $n$ points $\{x_1, x_2, ..., x_n\}$ into $k$ clusters

1. Construct the similarity matrix $W \in \mathbb{R}^{n \times n}$ as $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}$ (for $i \neq j$) and $W_{ii} = 0$.

2. Construct $D \in \mathbb{R}^{n \times n}$ as the diagonal matrix of degree nodes: $D_{ii} = \sum_j W_{ij}$.

3. Construct $\tilde{W} = D^{-\frac{1}{2}} WD^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$.

4. Let $\tilde{Y} \in \mathbb{R}^{n \times k}$ contain the left singular vectors of $B = (\tilde{W}\tilde{W}^T)^p\tilde{W}S$, with $p \geq 0$, and $S \in \mathbb{R}^{n \times k}$ being a matrix with $i.i.d$ random Gaussian variables.

5. Apply $k$-means clustering on the rows of $\tilde{Y}$, and cluster the original data points accordingly.

In a nutshell, “approximate” the top $k$ eigenvectors of $\tilde{W}$ and then apply $k$-means on the rows of the matrix containing those eigenvectors.
Related work

The Nystrom method: Uniform random sampling of the similarity matrix $W$ and then compute the eigenvectors. [Fowlkes et al. 2004]

The Spielman-Teng iterative algorithm: Very strong theoretical result based on their fast solvers for SDD systems of linear equations. Complex algorithm to implement. [2009]

Spectral clustering via random projections: Reduce the dimensions of the data points before forming the similarity matrix $W$. No theoretical results are reported for this method. [Sakai and Imiya, 2009].

Power iteration clustering: Like our idea but for the $k = 2$ case. No theoretical results reported. [Lin, Cohen, ICML 2010]

Other approximation algorithms: [Yen et al. KDD 2009]; [Shamir and Tishby, AISTATS 2011]; [Wang et al. KDD 2009]
Assume that $\|Y - \tilde{Y}\|_2 \leq \varepsilon$.

For all $i = 1 : n$, let $y_i^T, \tilde{y}_i^T \in \mathbb{R}^{1 \times k}$ be the $i$th rows of $Y, \tilde{Y}$.

Then,

$$\|y_i - \tilde{y}_i\|_2 \leq \|Y - \tilde{Y}\|_2 \leq \varepsilon.$$  

Clustering the rows of $Y$ and the rows of $\tilde{Y}$ with the same method should result to the same clustering.

A distance-based algorithm such as $k$-means would lead to the same clustering as $\varepsilon \to 0$.

This is equivalent to saying that $k$-means is robust to small perturbations to the input.
Approximation Framework for Spectral Clustering

The rows of $\tilde{Y}$ and $\tilde{Y}Q$, where $Q$ is some square orthonormal matrix, are clustered identically.

**Definition (Closeness of Approximation)**

$Y$ and $\tilde{Y}$ are close for “clustering purposes” if there exists a square orthonormal $Q$ such that

$$\|Y - \tilde{Y}Q\|_2 \leq \varepsilon.$$
This is really a problem of bounding subspaces

Lemma

There is an orthonormal matrix $Q \in \mathbb{R}^{n \times k}$ ($Q^T Q = I_k$) such that:

$$\|Y - \tilde{Y}Q\|_2^2 \leq 2k\|YY^T - \tilde{Y}\tilde{Y}^T\|_2^2.$$

- $\|YY^T - \tilde{Y}\tilde{Y}^T\|_2^2$ corresponds to the cosine of the principal angle between $\text{span}(Y)$ and $\text{span}(\tilde{Y})$.

- $Q$ is the solution of the following “Procrustes Problem”:

$$\min_Q \|Y - \tilde{Y}Q\|_F$$
The Singular Value Decomposition (SVD)

Let $A$ be an $m \times n$ matrix with $\text{rank}(A) = \rho$ and $k \leq \rho$.

$$A = U_A \Sigma_A V_A^T = \left( \begin{array}{cc} U_k & U_{\rho-k} \end{array} \right) \left( \begin{array}{cc} \Sigma_k & 0 \\ 0 & \Sigma_{\rho-k} \end{array} \right) \left( \begin{array}{c} V_k^T \\ V_{\rho-k}^T \end{array} \right).$$

- $U_k$: $m \times k$ matrix of the top-$k$ left singular vectors of $A$.
- $V_k$: $n \times k$ matrix of the top-$k$ right singular vectors of $A$.
- $\Sigma_k$: $k \times k$ diagonal matrix of the top-$k$ singular values of $A$. 
A “structural” result

**Theorem**

*Given* $A \in \mathbb{R}^{m \times n}$, let $S \in \mathbb{R}^{n \times k}$ *be such that*

$$\text{rank}(A_k S) = k$$

*and*

$$\text{rank}(V_k^T S) = k.$$  

*Let* $p \geq 0$ *be an integer and let*

$$\gamma_p = \|\sum_{\rho-k}^{2p+1} V_{\rho-k}^T S (V_k^T S)^{-1} \Sigma_k^{-(2p+1)}\|_2.$$  

*Then, for* $\Omega_1 = (AA^T)^p AS$, *and* $\Omega_2 = A_k$, *we obtain*

$$\|\Omega_1 \Omega_1^+ - \Omega_2 \Omega_2^+\|_2^2 = \frac{\gamma_p^2}{1 + \gamma_p^2}.$$
Some derivations lead to final result

\[
\gamma_p \leq \| \Sigma \|_{2}^{2p+1} \| V_{\rho-k}^T S \|_{2} \| (V_k^T S)^{-1} \|_{2} \| \Sigma^{-}(2p+1) \|_{2}
\]

\[
= \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \frac{\sigma_{\max} \left( V_{\rho-k}^T S \right)}{\sigma_{\min} \left( V_k^T S \right)}
\]

\[
\leq \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \frac{\sigma_{\max} \left( V^T S \right)}{\sigma_{\min} \left( V_k^T S \right)}
\]

\[
\leq \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \frac{4 \sqrt{n-k}}{\delta / \sqrt{k}}
\]

\[
= \left( \frac{\sigma_{k+1}}{\sigma_k} \right)^{2p+1} \cdot 4 \delta^{-1} \sqrt{k(n-k)}.
\]
Lemma (The norm of a random Gaussian Matrix)

Let $A \in \mathbb{R}^{n \times m}$ be a matrix with i.i.d. standard Gaussian random variables, where $n \geq m$. Then, for every $t \geq 4$,

$$
P\{\sigma_1(A) \geq tn^{\frac{1}{2}}\} \geq e^{-nt^2/8}.
$$

Lemma (Invertibility of a random Gaussian Matrix)

Let $A \in \mathbb{R}^{n \times n}$ be a matrix with i.i.d. standard Gaussian random variables. Then, for any $\delta > 0$

$$
P\{\sigma_n(A) \leq \delta n^{-\frac{1}{2}}\} \leq 2.35\delta.
$$
Main Theorem

If for some $\varepsilon > 0$ and $\delta > 0$ we choose

$$p \geq \frac{1}{2} \ln(4\varepsilon^{-1} \delta^{-1} \sqrt{k(n-k)}) \ln^{-1} \left( \frac{\sigma_k \left( \frac{\tilde{W}}{\tilde{W}} \right)}{\sigma_{k+1} \left( \frac{\tilde{W}}{\tilde{W}} \right)} \right),$$

then with probability at least $1 - e^{-n} - 2.35 \delta$,

$$\|Y - \tilde{Y}Q\|_2^2 \leq \frac{\varepsilon^2}{1 + \varepsilon^2} = O(\varepsilon^2).$$