Spectrum-Adapted Polynomial Approximation for Matrix Functions

Tiffany Fan and David Shuman
Joint work with Shashanka Ubaru and Yousef Saad
June 7, 2019
Graph Signal Processing Workshop
Minneapolis, MN
Motivation: Graph spectral filtering

Signal: $b$

Filter: $f(\lambda) = 1/(1 + 2\lambda)$

Filtered Signal: $f(A)b$

$$f(A) := \mathbf{V} f(\Lambda) \mathbf{V}^\top = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_N \end{bmatrix} \begin{bmatrix} f(\lambda_1) \\ \vdots \\ f(\lambda_N) \end{bmatrix} \begin{bmatrix} - \mathbf{v}_1 & - \\ \vdots \\ - \mathbf{v}_N & - \end{bmatrix}$$
Spectral decomposition of large matrices is prohibitively expensive

- Complexity: $O(N^3)$  \[ \mathbf{b} \quad \rightarrow \quad f(\mathbf{A})\mathbf{b} \]
- How to approximately compute this efficiently and accurately?
  - Truncated polynomial expansions (e.g. Chebyshev, Jacobi)
  - Rational approximations
  - Krylov subspace methods (e.g. Lanczos)
  - Quadrature/contour integral methods

Frommer and V. Simoncini, “Matrix functions,” in Model Order Reduction: Theory, Research Aspects and Applications, pp. 275–303. Springer, 2008.

Loukas, Simonetto, and Leus, “Distributed autoregressive moving average graph filters,” IEEE Signal Process. Lett., 2015.

Druskin and Knizhnerman, “Two polynomial methods of calculating functions of symmetric matrices,” U.S.S.R. Comput. Maths. Math. Phys., 1989.
Efficient computation with polynomial expansions

- Approximate \( f(\lambda) \approx p_K(\lambda) = c_0 + \sum_{k=1}^{K} c_k \lambda^k \)

- Iterative multiplication

\[
x^{(0)} = c_K b \\
x^{(l)} = c_{K-l} b + A x^{(l-1)}, \quad l = 1, 2, \ldots, K
\]

- Complexity: \( \mathcal{O}(KZ) \) where \( Z \) is the number of nonzero entries in \( A \); fast for large sparse graphs
example: \[ f(\lambda) = e^{-\lambda} \]

Classical methods find an order \( K \) polynomial \( p_K(\lambda) \) by minimizing the error on the interval \([\underline{\lambda}, \overline{\lambda}]\)

However, the error in \( f(A)b \) depends only on the residuals at the eigenvalues of \( A \)

\[
\min_{p \in \mathcal{P}_K} \sum_{l=1}^{N} [f(\lambda_l) - p(\lambda_l)]^2 = \max_{l=1,2,\ldots,N} |f(\lambda_l) - p_K(\lambda_l)| \leq \sup_{\lambda \in [\underline{\lambda}, \overline{\lambda}]} |f(\lambda) - p_K(\lambda)|
\]

Better approximation without computing all the eigenvalues?
Main idea

More accurately approximate the filter function where there are more eigenvalues

Step 1 Efficiently estimate the distribution of eigenvalues

Step 2 Find the best polynomial that minimizes the error on the regions with higher density of eigenvalues
Eigenvalue distribution estimation: Kernel polynomial method

- Generate evenly spaced points \( \{ \xi_i \} \) on \( [\Lambda, \bar{\lambda}] \)

- Build ideal lowpass filter to estimate the count of eigenvalues below \( \xi_i \)

\[
\eta_i = \sum_{\ell=1}^{N} \mathbb{I}\{\lambda_\ell \leq \xi_i\} \\
= \text{tr}(\Theta_{\xi_i}(A))
\]

\[
\begin{align*}
\text{tr}(\Theta_{\xi_i}(A)) &= \text{tr}(V \Theta_{\xi_i}(\Lambda) V^{-1}) = \text{tr}(\Theta_{\xi_i}(\Lambda)) \\
&= \text{tr} \left( \begin{bmatrix}
\Theta_{\xi_i}(\lambda_1) & 0 & \cdots & 0 \\
0 & \Theta_{\xi_i}(\lambda_2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \Theta_{\xi_i}(\lambda_N)
\end{bmatrix} \right) \\
&= \text{tr} \left( \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0
\end{bmatrix} \right)
\end{align*}
\]
Eigenvalue distribution estimation: Kernel polynomial method

- Build ideal lowpass filter to estimate the count of eigenvalues below $\xi_i$

$$\eta_i = \sum_{\ell=1}^{N} \mathbb{I}\{\lambda_\ell \leq \xi_i\}$$

$$= \text{tr} \left( \Theta_{\xi_i}(A) \right) \quad \text{stochastic trace estimator}$$

$$= \mathbb{E} \left[ x^T \Theta_{\xi_i}(A) x \right]$$

$$\approx \frac{1}{J} \sum_{j=1}^{J} x^{(j)}^T \Theta_{\xi_i}(A) x^{(j)}$$

approximate with polynomial filter

Lin et al., “Approximating spectral densities of large matrices,” SIAM Review, 2016.
Eigenvalue distribution estimation: Kernel polynomial method

- Interpolate a monotonic piecewise cubic function $\tilde{P}_\lambda(z)$
- Compute spectral PDF $\tilde{p}_\lambda(z)$ and inverse CDF $\tilde{P}_\lambda^{-1}(z)$

Lin et al., “Approximating spectral densities of large matrices,” SIAM Review, 2016.
Eigenvalue distribution estimation: Examples

- **G(500,0.2)**
- **Minnesota**
- **cage9**
- **net25**
Step 1 Efficiently estimate the distribution of eigenvalues

Step 2 Find the best polynomial that minimizes the error on the regions with higher density of eigenvalues

- Spectrum-adapted interpolation

- Spectrum-adapted weighted least squares regression
Spectrum-adapted interpolation

- Generate $K+1$ Chebyshev nodes $\{y_i\}$ between 0 and 1
- Warp them via $x_i = \tilde{P}_\lambda^{-1}(y_i)$
- Interpolate a unique order $K$ polynomial $p_K(\lambda)$ through points $\{(x_i, f(x_i))\}$
- Compute $p_K(A)b$ recursively
Step 1 Efficiently estimate the distribution of eigenvalues

Step 2 Find the best polynomial that minimizes the error on the regions with higher density of eigenvalues

- Spectrum-adapted interpolation

- Spectrum-adapted weighted least squares regression
Spectrum-adapted weighted least squares regression

- Generate M points \( \{ x_m \} \) on \( [\underline{\lambda}, \bar{\lambda}] \)
- Compute \( \{ w_m \} = \tilde{p}_\lambda(\{ x_m \}) \) as weights
Spectrum-adapted weighted least squares regression

- Generate $M$ points $\{x_m\}$ on $[\underline{\lambda}, \bar{\lambda}]$
- Compute $\{w_m\} = \tilde{p}_\lambda(\{x_m\})$ as weights
- Find a unique order $K$ polynomial $p_K(\lambda)$ via weighted least squares

$$p_K(\lambda) = \min_{p \in \mathcal{P}_K} \sum_{m=1}^{M} w_m [f(x_m) - p(x_m)]^2$$

- Compute $p_K(A)\mathbf{b}$ recursively
Alternative interpretation: Expansion in spectrum-adapted orthogonal polynomials

- **Goal:** represent \( f \) as a linear combination of \( \{ \pi_{k,M}(\lambda) \} \), polynomials orthogonal with respect to the discrete measure \( \{ w_m \} = \tilde{p}_\lambda(\{ x_m \}) \).

\[
f(\lambda) \approx p_K(\lambda) = \sum_{k=0}^{K} \gamma_k \pi_{k,M}(\lambda)
\]

- **Compute** \( p_K(A)b \) as a linear combination of \( \{ \pi_{k,M}(A)b \} \).
Numerical Experiments
Comparison of resulting degree 10 approximations
Comparison of resulting degree 10 approximations
Comparison of different graphs

Spectral Distribution

Relative Error

\[ \sum_{\ell=1}^{N} \frac{(f(\lambda_\ell) - p_K(\lambda_\ell))^2}{\sum_{\ell=1}^{N} f(\lambda_\ell)^2} \]
Summary

- Overall complexity of the proposed approximation methods is linear in the number of edges in the graph.
- The proposed spectrum-adapted methods are amenable to efficient distributed computation.
- Spectrum-adapted interpolation method often works well for low degree approximations \((K<10)\), but is not very stable at higher orders due to ill-conditioning.
- Spectrum-adapted weighted least squares method tends to outperform standard methods (e.g., truncated Chebyshev expansion, the Lanczos method) for matrices with a large number of distinct interior eigenvalues and a smaller spectral width.