Agenda

- Spectral Graph for node embeddings
  - **Approximation theory** and spectral graph theory
  - **Chebyshev approximation and experiments**
    - Chevyshev fundamentals(link)
    - Chevyshev implementation (GCN v.s. GCN_cheby)
    - Parameters study by experiments
    - Other option: Lancsoz methods

Chebyshev methods

- The discussion here will mainly focus on two recent papers:
  - Defferrard et al. (**NIPS 2016**), Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering
    - **Contribution: First introduced Spectral Graph theory and Chebyshev approximation by CNN, GCN_cheby**
    - **10k nodes will burst the memory**
  - Kipf & Welling (**ICLR 2017**), Semi-Supervised Classification with Graph Convolutional Networks
    - **Contribution: Effectiveness and Efficiency Improvement GCN**
- Experimental results recap:
  - random walk + word2vec v.s. spectral graph (always better)
  - GCN(always better) v.s. GCN_cheby (**slow, and often crashes**)

What is Chebyshev Poly.

Definition 1.1 *The Chebyshev polynomial $T_n(x)$ of the first kind is a polynomial in x of degree n, defined by the relation*

$$T_n(x) = \cos n\theta \quad \text{when } x = \cos\theta.$$

Definition 1.2 *The Chebyshev polynomial $U_n(x)$ of the second kind is a polynomial of degree n in x defined by*

$$U_n(x) = \sin(n+1)\theta / \sin\theta \quad \text{when } x = \cos\theta. \tag{1.4}$$

Definition 1.3 *The Chebyshev polynomials $V_n(x)$ and $W_n(x)$ of the third and fourth kinds are polynomials of degree n in x defined respectively by*

$$V_n(x) = \cos(n+\tfrac{1}{2})\theta / \cos\tfrac{1}{2}\theta \tag{1.8}$$

and

$$W_n(x) = \sin(n+\tfrac{1}{2})\theta / \sin\tfrac{1}{2}\theta, \tag{1.9}$$

when $x = \cos\theta$.

★ $$\cos 0\theta = 1, \quad \cos 1\theta = \cos\theta, \quad \cos 2\theta = 2\cos^2\theta - 1,$$
$$\cos 3\theta = 4\cos^3\theta - 3\cos\theta, \quad \cos 4\theta = 8\cos^4\theta - 8\cos^2\theta + 1, \quad \ldots.$$

★ $$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1,$$
$$T_3(x) = 4x^3 - 3x, \quad T_4(x) = 8x^4 - 8x^2 + 1, \quad \ldots.$$

★ $$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n = 2, 3, \ldots, \qquad \langle T_i\,, T_j \rangle = 0 \quad (i \neq j).$$

Application: minimax approximation of functions by polynomials.

Good $\|f - f^*\| \leq \epsilon$

Best $\|f - f_B^*\| \leq \|f - f^*\|$

Near-Best $\|f - f_N^*\| \leq (1+\rho)\,\|f - f_B^*\|$

Theorem 3.2 (Weierstrass's theorem) *For any given f in $C[a,b]$ and for any given $\epsilon > 0$, there exists a polynomial p_n for some sufficiently large n such that $\|f - p_n\|_\infty < \epsilon$.*

Theorem 3.3 *For any given p $(1 \leq p \leq \infty)$, there exists a unique best polynomial approximation p_n to any function $f \in \mathcal{L}_p[a,b]$ in the $\mathcal{L}_p$ norm, where $w(x)$ is taken to be unity in the case $p \to \infty$.*

How to use Chebyshev Poly.

Orthogonality(think about Fourier series)

Definition 4.1 *Two functions $f(x)$ and $g(x)$ in $\mathcal{L}_2[a,b]$ are said to be* orthogonal *on the interval $[a,b]$ with respect to a given continuous and non-negative weight function $w(x)$ if*

$$\int_a^b w(x)f(x)g(x)\,\mathrm{d}x = 0. \tag{4.1}$$

$$\langle f\,,\,g\rangle = \int_a^b w(x)f(x)g(x)\,\mathrm{d}x,$$

For the 1st kind of Chebyshev Ploy.

$$w(x) = \frac{1}{\sqrt{1-x^2}}$$

$$\begin{aligned}\langle T_i\,,\,T_j\rangle &= \int_{-1}^{1}\frac{T_i(x)T_j(x)}{\sqrt{1-x^2}}\,\mathrm{d}x\\ &= \int_0^\pi \cos i\theta\,\cos j\theta\,\mathrm{d}\theta\end{aligned}$$

Best approx. property

Corollary 4.1B *The best $\mathcal{L}_2$ polynomial approximation p_n^B of degree n to f may be expressed in terms of the orthogonal polynomial family $\{\phi_i\}$ in the form*

$$p_n^B = \sum_{i=0}^{n} c_i\phi_i,$$

where

$$c_i = \frac{\langle f\,,\,\phi_i\rangle}{\langle \phi_i\,,\,\phi_i\rangle}.$$

Example

EXAMPLE 4.1: To illustrate Corollary 4.1B, suppose that we wish to determine the best $\mathcal{L}_2$ linear approximation p_1^B to $f(x) = 1-x^2$ on $[-1,1]$, with respect to the weight $w(x) = (1-x^2)^{-\frac12}$. In this case $\{T_i(x)\}$ is the appropriate orthogonal system and hence

$$p_1^B = c_0T_0(x) + c_1T_1(x)$$

where, by (4.17),

$$\begin{aligned} c_0 &= \frac{\langle f\,,\,T_0\rangle}{\langle T_0\,,\,T_0\rangle} = \frac{\int_{-1}^{1}(1-x^2)^{-\frac12}(1-x^2)\,\mathrm{d}x}{\pi},\\ c_1 &= \frac{\langle f\,,\,T_0\rangle}{\cancel{\langle T_0\,,\,T_0\rangle}} = \frac{\int_{-1}^{1}(1-x^2)^{-\frac12}(1-x^2)x\,\mathrm{d}x}{\frac12\pi}.\end{aligned}$$

<T1,T1>

Substituting $x = \cos\theta$,

$$\begin{aligned} c_0 &= \frac{1}{\pi}\int_0^\pi \sin^2\theta\,\mathrm{d}\theta = \frac{1}{2\pi}\int_0^\pi (1-\cos 2\theta)\,\mathrm{d}\theta = \tfrac12,\\ c_1 &= \frac{2}{\pi}\int_0^\pi \sin^2\theta\,\cos\theta\,\mathrm{d}\theta = \frac{2}{\pi}\left[\tfrac13\sin^3\theta\right]_0^\pi = 0\end{aligned}$$

and therefore

$$p_1^B = \tfrac12 T_0(x) + 0\,T_1(x) = \tfrac12,$$

so that the linear approximation reduces to a constant in this case.

How to use Chebyshev Poly.

Best approx. of Chebyshev Poly.

$$f(x) \sim \sum_{i=0}^{\infty} c_i \phi_i(x)$$

$$c_i = \langle f\,,\,\phi_i \rangle / \langle \phi_i\,,\,\phi_i \rangle$$

$$\phi_i(x) = T_i(x),\ U_i(x),\ V_i(x) \text{ or } W_i(x)$$

$$f(x) \sim {\sum_{i=0}^{\infty}}' c_i T_i(x) = \tfrac{1}{2} c_0 T_0(x) + c_1 T_1(x) + c_2 T_2(x) + \cdots$$

where

$$c_i = \frac{2}{\pi} \int_{-1}^{1} (1-x^2)^{-\frac{1}{2}} f(x) T_i(x)\,\mathrm{d}x,$$

Theorem 3.1. Chebyshev series. *If f is Lipschitz continuous on $[-1,1]$, it has a unique representation as a Chebyshev series,*

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x), \tag{3.11}$$

which is absolutely and uniformly convergent, and the coefficients are given for $k \geq 1$ by the formula

$$a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx, \tag{3.12}$$

and for $k = 0$ by the same formula with the factor $2/\pi$ changed to $1/\pi$.

Error bound

Theorem 5.14 *If the function $f(x)$ has $m+1$ continuous derivatives on $[-1,1]$, then $\left|f(x) - S_n^T f(x)\right| = O(n^{-m})$ for all x in $[-1,1]$.*

Example

Example 5.1: Expansion of $f(x) = \sqrt{1-x^2}$.

Here

$$\begin{aligned}
\frac{\pi}{2} c_i &= \int_{-1}^{1} T_i(x)\,\mathrm{d}x = \int_0^{\pi} \cos i\theta \sin\theta\,\mathrm{d}\theta \\
&= \tfrac{1}{2} \int_0^{\pi} [\sin(i+1)\theta - \sin(i-1)\theta]\,\mathrm{d}\theta \\
&= \tfrac{1}{2} \left[\frac{\cos(i-1)\theta}{i-1} - \frac{\cos(i+1)\theta}{i+1} \right]_0^{\pi} \quad (i \geq 1) \\
&= \tfrac{1}{2} \left(\frac{(-1)^{i-1} - 1}{i-1} - \frac{(-1)^{i+1} - 1}{i+1} \right)
\end{aligned}$$

and thus

$$c_{2k} = -\frac{4}{\pi(4k^2-1)}, \quad c_{2k-1} = 0 \quad (k = 1, 2, \ldots).$$

Also

$$c_0 = 4/\pi.$$

Hence,

$$\begin{aligned}
\sqrt{1-x^2} &\sim -\frac{4}{\pi} {\sum_{k=0}^{\infty}}' \frac{T_{2k}(x)}{4k^2-1} \\
&= \frac{4}{\pi} \left(\tfrac{1}{2} T_0(x) - \tfrac{1}{3} T_2(x) - \tfrac{1}{15} T_4(x) - \tfrac{1}{35} T_6(x) - \cdots \right)
\end{aligned}$$

Chebyshev methods

- Defferrard et al. (**NIPS 2016**), Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, GCN_cheby
- INPUT:
  - s: raw nodes features
  - L: graph Laplacian
- OUTPUT:
  - s': new embeddings of s

Chebyshev methods

2.2 Graph filters

In the classic setting, applying a filter to a signal is performed by convolution, which corresponds to point-wise multiplication in the spectral domain. Similarly, filtering a graph-signal is peformed by multiplication with a filter in the graph Fourier domain. A graph filter is defined by a continuous function $g : \mathbb{R}_+ \to \mathbb{R}$. To obtain its discrete coefficients, this function is evaluated at each eigenvalue: $g(\lambda_\ell)$ for $\ell = 0, \ldots, N-1$. The filtering operation then corresponds to $\hat{s}'(\ell) = g(\lambda_\ell) \cdot \hat{s}(\ell)$, where s' is the filtered signal. Equivalently, using matrix notation, we have

$s \in \mathbb{R}^n$ assigning a value to each vertex

$U = [u_0, \ldots, u_{n-1}] \in \mathbb{R}^{n \times n}$

graph Fourier transform $\hat{s} = U^* s \in \mathbb{R}^n$

$$s' = U g(\Lambda) U^* s, \qquad (1)$$

inverse $s = U\hat{s}$.

where $g(\Lambda)$ is the diagonal matrix containing the coefficients $g(\lambda_\ell)$ on the diagonal. In terms of matrix functions [11], the relation (1) can be compactly expressed as $s' = g(\mathcal{L})s$ with $g(\mathcal{L}) := U g(\Lambda) U^*$.

$g_\theta(\Lambda) = \mathrm{diag}(\theta) \quad \theta \in \mathbb{R}^n$

$g_\theta(\Lambda)\, \hat{s} = \theta \odot \hat{s}$

Chebyshev methods

$$g_{\theta'} \star s = s' = U g(\Lambda) U^* s$$

$\mathcal{O}(N^3)$

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$$

$$(U \Lambda U^\top)^k = U \Lambda^k U^\top$$

$$g_{\theta'} \star s \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})$$

$$s' = U g(\Lambda) U^* s = \left(a_0 I + a_1 \mathcal{L} + \cdots + a_M \mathcal{L}^M\right) s$$

Chebyshev methods

- **GCN_cheby**
  - **calculate normalized graph Laplacian** $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^\top$
  - **calculate eigenvalues of normalized graph Laplacian (time consuming)**
  - **shift the normalized Laplacian to [-1,1]** $\tilde{L} = \frac{2}{\lambda_{\max}} L - I_N$
  - **apply Chebyshev polynomial** $g_{\theta'} \star x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L}) x$
- **GCN**
  - **calculate normalized graph Laplacian** $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$
  - **assume lambda_max=2, shift the L** $\tilde{L} = \quad L - I_N$
  - **apply Chebyshev polynomial** $g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 \left(L - I_N\right) x = \theta'_0 x - \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x$
  - $\theta = \theta'_0 = -\theta'_1$ $\quad g_\theta \star x \approx \theta \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x$

renormalization trick: $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$

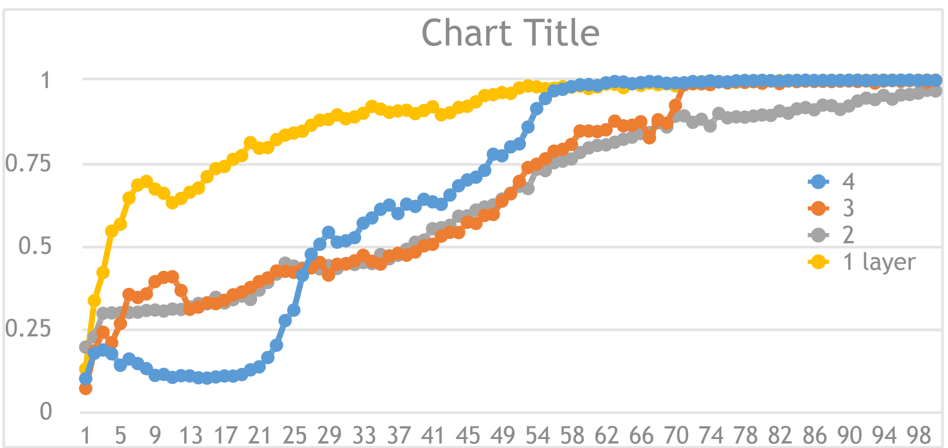
$\tilde{A} = A + I_N$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

$g_\theta \star x \approx \theta \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \right) x$

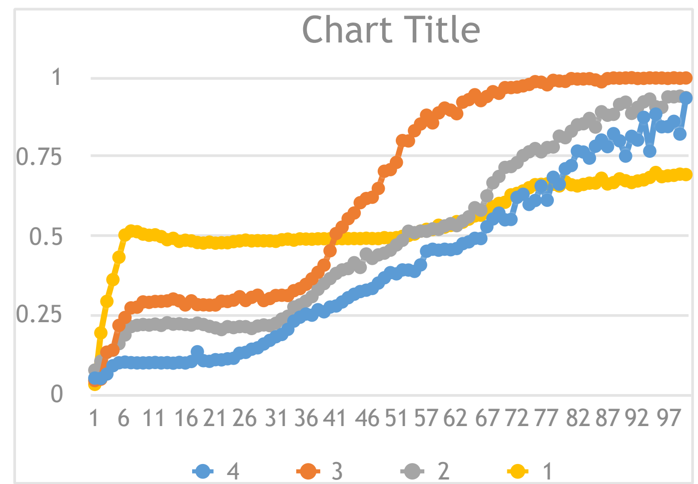
Chebyshev experiments

Performance under difference numbers of layers(GCN)

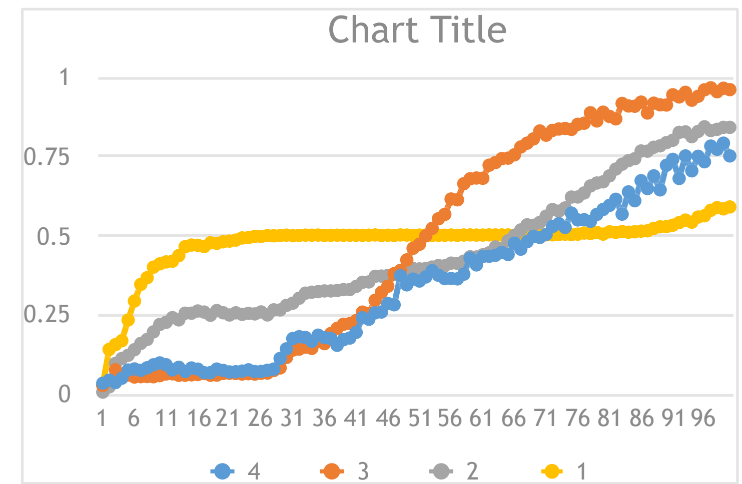
10k dp	1k feat	10class



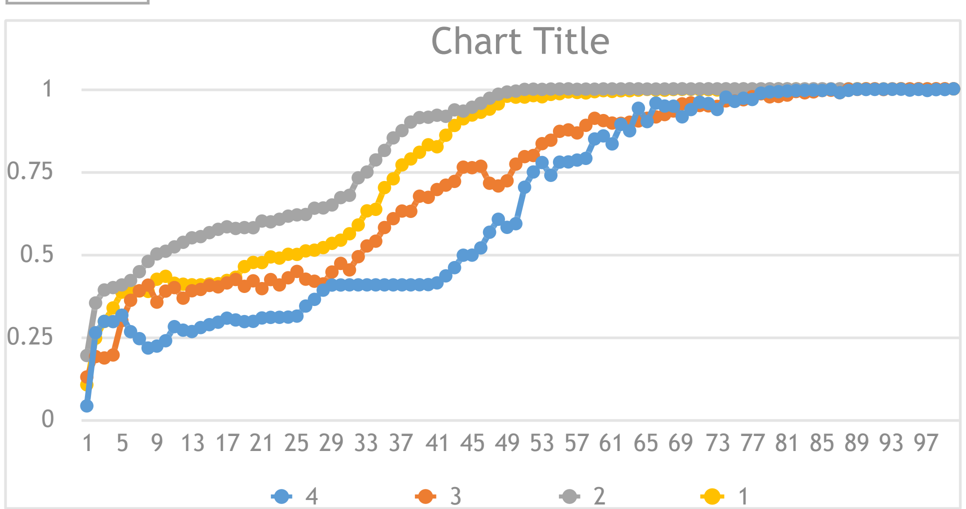
20class



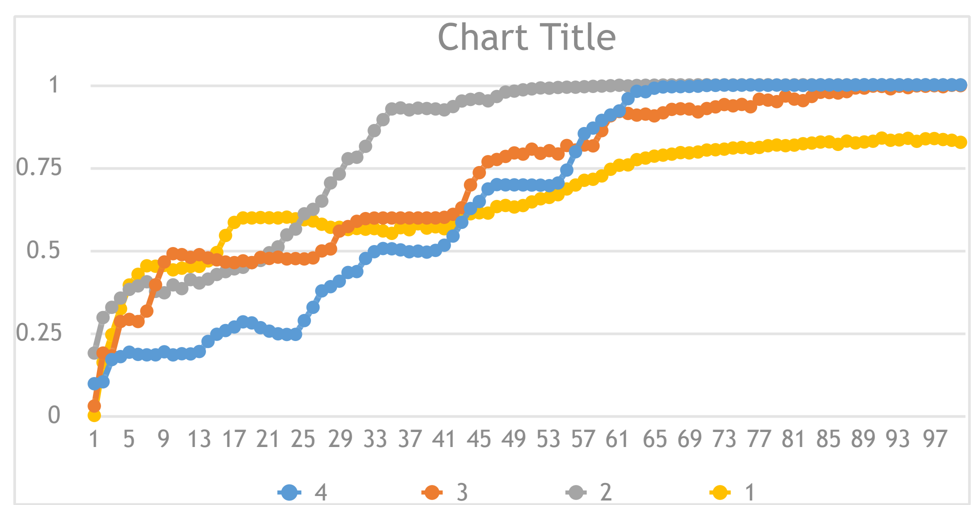
50class



100feat



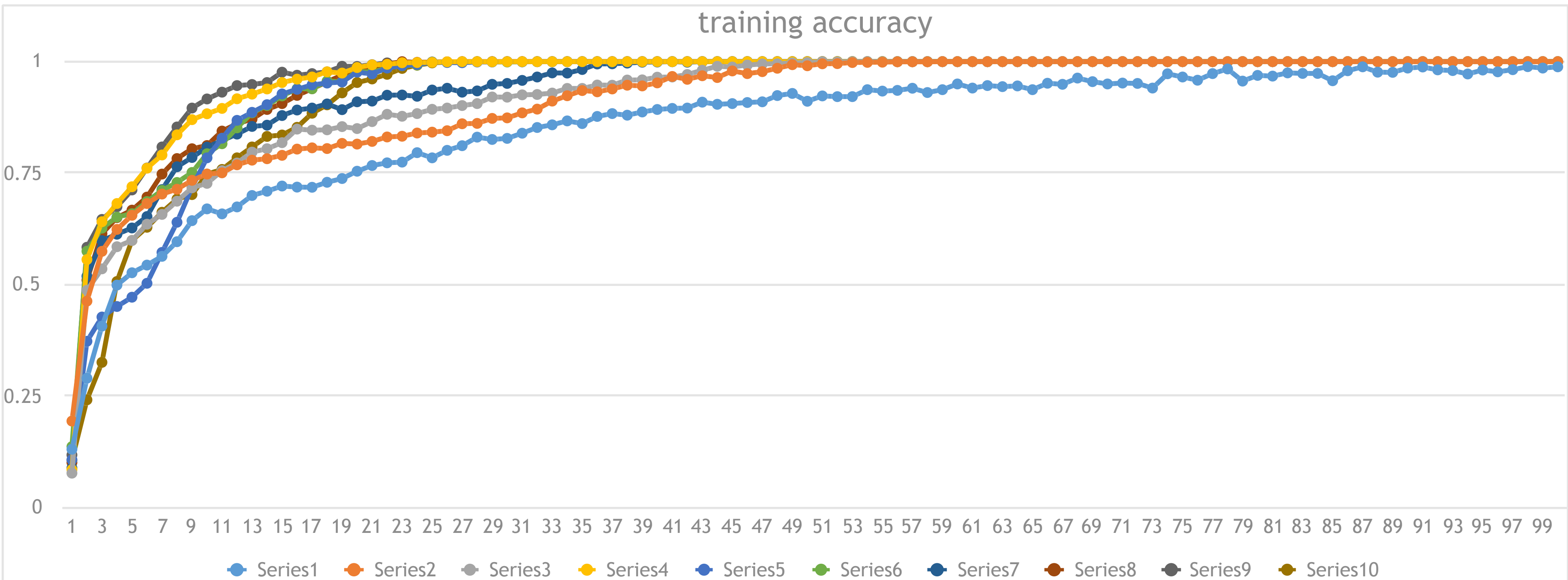
20k dp

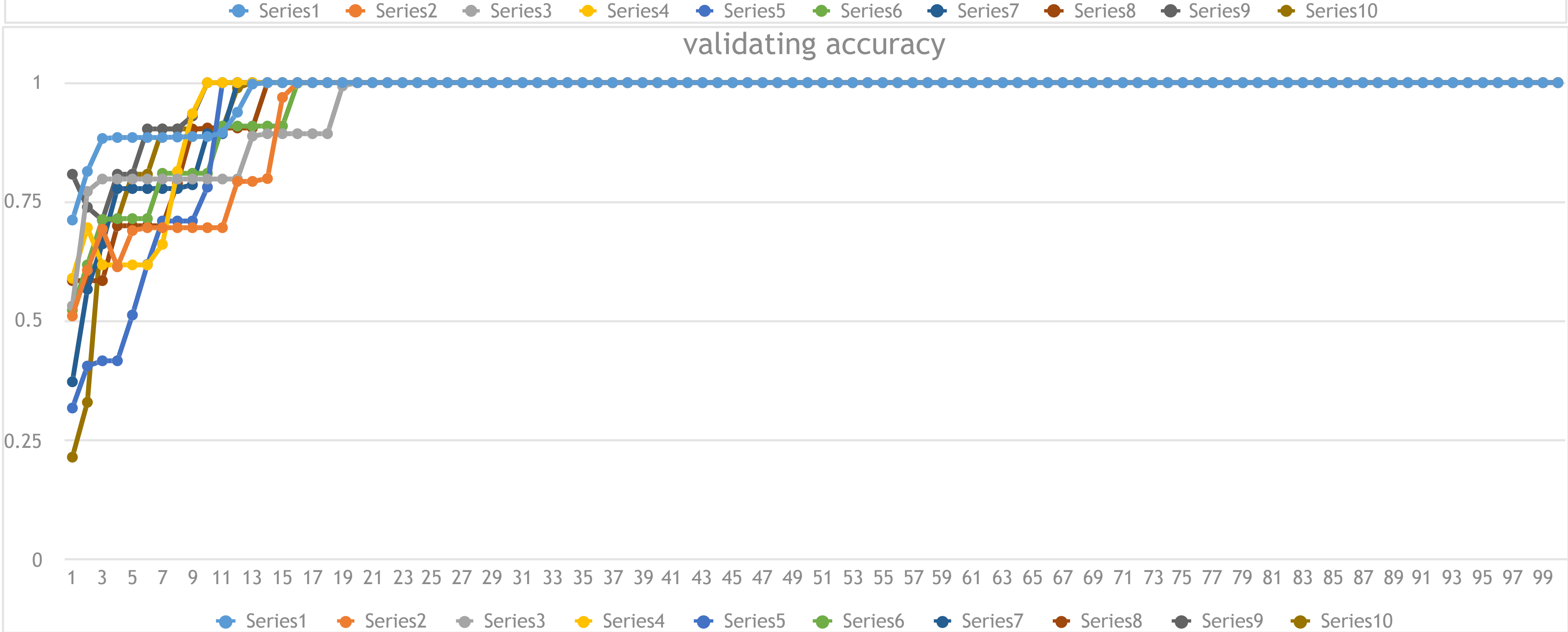


Chebyshev experiments

Performance under difference numbers of Chebyshev orders

5k dp	0.5k feat	10class

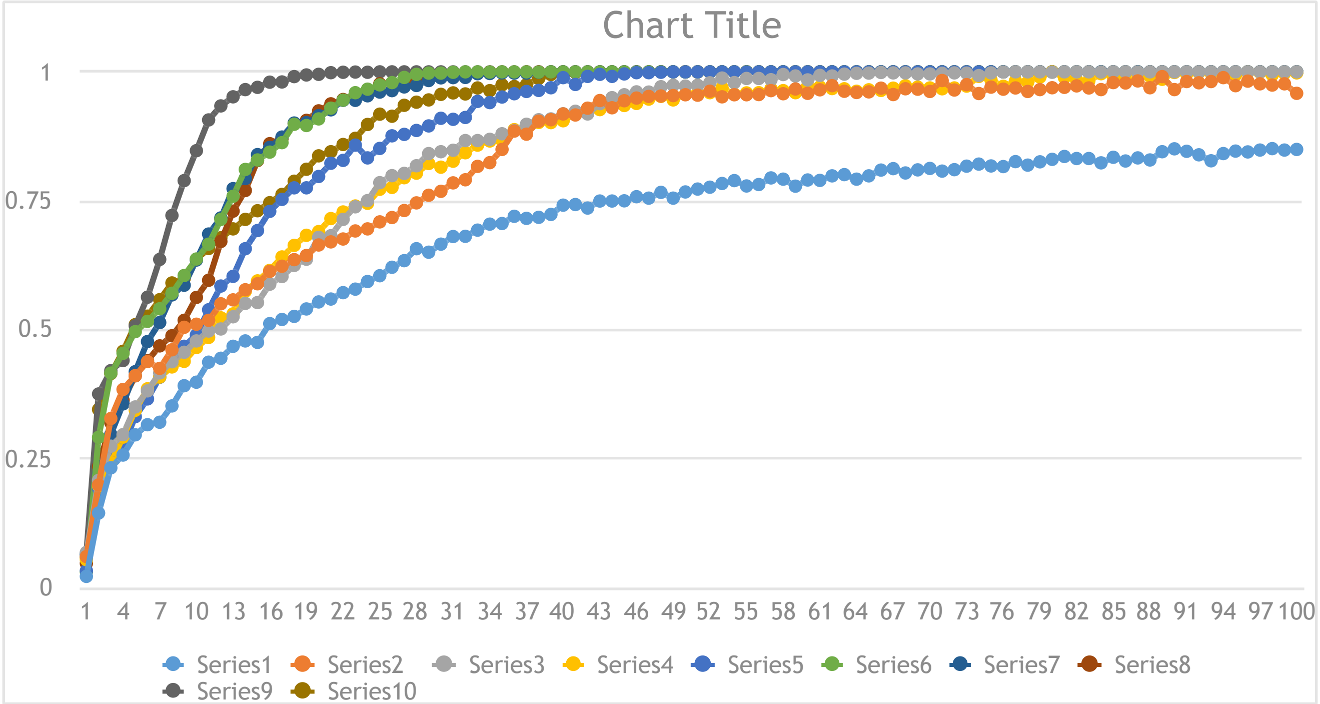




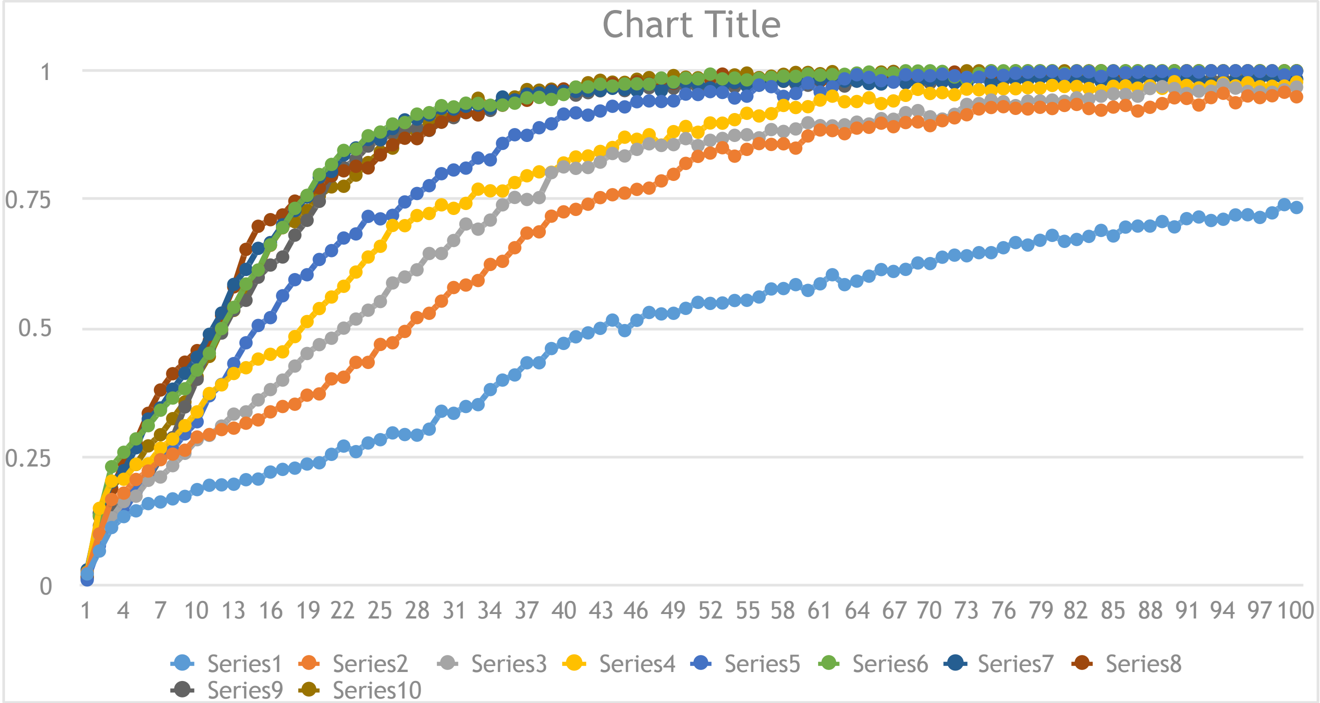
Chebyshev experiments

Performance under difference numbers of Chebyshev orders

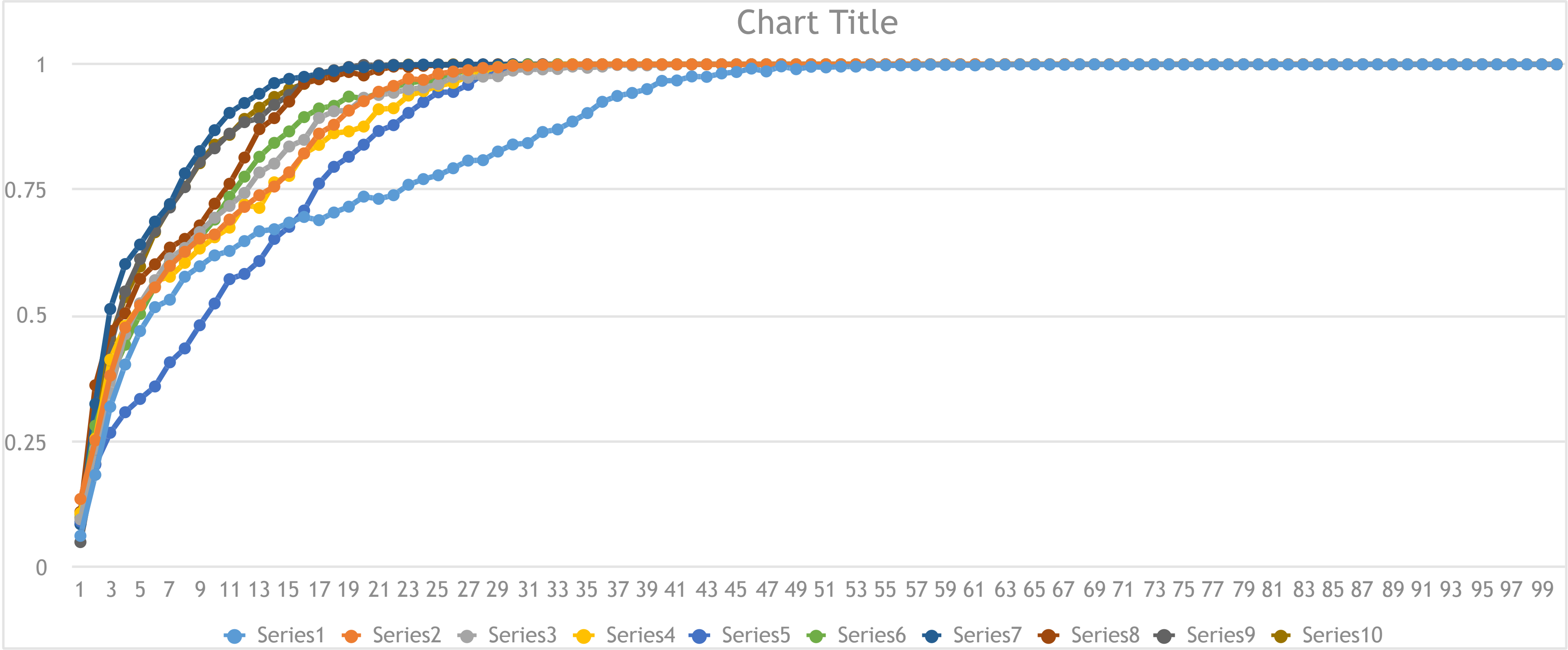
20class



50class



100feat



Any option for Chebyshev?

- 2nd, 3rd and 4th Chebyshev polynomials

Definition 1.2 *The Chebyshev polynomial $U_n(x)$ of the second kind is a polynomial of degree n in x defined by*

$$U_n(x) = \sin(n+1)\theta / \sin\theta \quad \text{when } x = \cos\theta. \tag{1.4}$$

Definition 1.3 *The Chebyshev polynomials $V_n(x)$ and $W_n(x)$ of the third and fourth kinds are polynomials of degree n in x defined respectively by*

$$V_n(x) = \cos(n+\tfrac{1}{2})\theta / \cos\tfrac{1}{2}\theta \tag{1.8}$$

and

$$W_n(x) = \sin(n+\tfrac{1}{2})\theta / \sin\tfrac{1}{2}\theta, \tag{1.9}$$

when $x = \cos\theta$.

- Fourier series equal to Chebyshev series when g is even

$$g(\theta) \sim \tfrac{1}{2}a_0 + \sum_{k=1}^{\infty}(a_k \cos k\theta + b_k \sin k\theta) \tag{5.41}$$

where

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} g(\theta)\cos k\theta \, d\theta, \quad b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} g(\theta)\sin k\theta \, d\theta, \quad (k = 0, 1, 2, \ldots). \tag{5.42}$$

$\Rightarrow$

$$g(\theta) \sim {\sum_{k=0}^{\infty}}' a_k \cos k\theta$$

where

$$a_k = \frac{2}{\pi}\int_0^{\pi} g(\theta)\cos k\theta \, d\theta.$$

$\Rightarrow$

$$f(x) \sim {\sum_{k=0}^{\infty}}' a_k T_k(x)$$

where

$$a_k = \frac{2}{\pi}\int_{-1}^{1}(1-x^2)^{-\frac{1}{2}} f(x) T_k(x) \, dx.$$

- Lanczos method (mentioned in **GCN_cheby**):
  - Complexity comparison with GCN: $\mathcal{O}(K|\mathcal{E}|)$ v.s. $\mathcal{O}(M \cdot |\mathcal{E}|)$

Given a continuous function $g : [0, \lambda_{\max}] \to \mathbb{R}$ and a vector s, the following approximation to $g(\mathcal{L})s$ was proposed by Gallopoulos and Saad in [6]:

$$g(\mathcal{L})s \approx \|s\|_2 V_M g(H_M) e_1 := g_M, \tag{4}$$

  - Lanczos approximation can be expected to perform significantly better because of its ability to adapt to the eigenvalues of L. (This phenomenon is well-understood for Krylov subspace approximations to solutions of linear system.)

Susnjara, Ana, et al. "Accelerated filtering on graphs using lanczos method." *arXiv preprint arXiv:1509.04537* (2015).