

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

TERMÉSZETTUDOMÁNYI KAR

GRÁF KONVOLÚCIÓS HÁLÓZATOK ÉS ALKALMAZÁSAIK

BSc Szakdolgozat

Harsányi Benedek

Matematikus szakirány

Témavezető: Lukács András

Számítógéptudományi Tanszék



Budapest 2021

Köszönetnyilvánítás

Szeretnék köszönetet mondani a témavezetőmnek, Lukács Andrásnak az egész éves rendszeres konzultációkért és az izgalmas témajavaslatért, valamint hálás vagyok azért, hogy megismertette velem a gépi tanulás tudományát.

Köszönet illeti még Ács Juditot, aki nélkül nem jöhettek volna létre a dolgozatban található mérési eredmények. Szeretném megköszönni a segítőkészségét és a hasznos tanácsait, amik nagyban hozzájárultak a programozáshoz köthető előzetes féltelmeim leküzdésében.

Végül szeretném megköszönni a családomnak és a barátnőmnek, Horváth Hildának, azt a rengetek szeretetet és támogatást, amit az évek során kaptam tőlük.

Tartalomjegyzék

1. Gépi tanulás gráfokon	4
2. Gráfelméleti alapfogalmak	5
2.1. Laplace-mátrix	5
2.2. Spektrum meghatározása	8
2.3. Gráfkonvolúció	10
3. Gráf filterek	16
3.1. Csebisev-polinomok	16
3.2. Cayley-polinomok	23
3.3. Racionális törtfüggvények, ARMA	27
3.4. B-Spline függvények	29
4. Gráf kernelek	34
4.1. Kernelek karakterizációja	34
4.2. Support Vector Machine	37
4.3. Weisfeiler-Lehman kernel	40
5. Kísérletek	42
5.1. Képlaszifikáció rácsgráfokkal	42
5.2. Molekula osztályozás	44
6. Alkalmazások	47
6.1. Mátrix kiegészítési probléma	47
6.2. Maximális független csúcshalmaz keresése	52
Hivatkozások	56

1. Gépi tanulás gráfokon

A mai világban előforduló adatok között sokszor olyan komplex összefüggések figyelhetőek meg, amiket gráfokkal a legegyszerűbb modellezni. Például a közösségi hálók vagy a molekulák felépítése is gráfokkal modellezhető legkönnyebben. A gráfokon történő gépi tanulás nehézségét az okozza, hogy nem világos milyen matematikai eszközök segítségével reprezentálhatóak megfelelően az adatok. Megoldást a gépi látásnál alkalmazott módszerek általánosításával kaphatunk. Tekintsük egy képklasszifikációs feladatot, ahol az input egy kép, és a gép feladata felismerni, hogy mi látható a képen. A konvolúciós hálózatok megjelenésével labor körülmények között a gépek képesek közel humán teljesítményt elérni. A konvolúciós hálók úgynevezett filter függvények segítségével működnek, amelyek a pixelek térbeli függését figyelembe véve nyernek ki információt a képekből. A konvolúció műveletét többféleképpen is lehet gráfokra általánosítani, egyik lehetséges eljárás a konvolúciós tételen keresztül spektrálgráfelméleti eszközökkel történik, egy másik módszer pedig csupán a csúcsok közötti szomszédsági kapcsolatra támaszkodik.

A dolgozatban áttekintjük a spektrálgráfelmélet alapvető fogalmait, majd Fourier-analízis eszköztárához analóg módon definiáljuk a konvolúciót. A 3. fejezetben a Csebisev- és Cayley-polinomokon, majd a racionális törtfüggvényeken alapuló filterezési technikákat tekintjük végig. A Spline függvényeken keresztül mutatunk egy példát a térbeli kapcsolatokat kihasználó konvolúcióra. A 4. fejezetben a gráfokon történő tanulás egy másik lehetséges megközelítését, a gráfkerneleket járjuk körbe. Az 5. fejezetben ismertetem a mérési eredményeimet, amelyeket képklasszifikációs és a molekula osztályozási feladaton végeztem. A 6. fejezetben az ajánlórendszerek egy lehetséges gráfokon alapuló modelljét mutatjuk be, majd egy NP nehéz feladathoz konstruálunk heurisztikus függvényeket gráf konvolúciós hálók segítségével.

A gépi tanulási klasszifikációs feladatokban adott egy tanító adathalmaz $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, ahol az x_i adatponthoz y_i címke tartozik, szeretnénk az adathalmaz segítségével előállítani egy f_θ függvényt, amit egy neurális háló valósít meg, amire $f_\theta(x_i) = \hat{y}_i \approx y_i$ teljesül. Tanulás alatt az f_θ -hoz tartozó θ paraméterek megfelelő beállítását értjük. A tanulási folyamat egy L célfüggvény minimalizálásával történik általában a gradiens módszer segítségével. Az $L(\hat{y}, y)$ függvény \hat{y} -on keresztül függ θ -tól, a θ -ra vonatkozó deriváltak segítségével frissítjük a súlyokat, így y egy jobb közelítéséhez jutunk. Célunk körbejárni, hogy ha az adathalmazunk gráfokból áll, akkor milyen f_θ függvényeket érdemes alkalmazni.

2. Gráfelméleti alapfogalmak

A fejezetben található gráfelméleti fogalmak az [1] és [2] forrás mintájára, a lineáris algebrai eredmények a [3] jegyzet alapján, a Rayleigh- és a Courant-Fischer-tételek bizonyítása a [4] könyv alapján íródott.

2.1. Laplace-mátrix

Legyen $\mathcal{G} = (V, E)$ egy véges irányítatlan egyszerű gráf, ahol $V = \{1, 2, \dots, n\}$ a gráf csúcshalmaza és $E \subset V \times V$ a gráf élhalmaza jelöli, ahol $|E| = m$. Súlyozott gráfok esetén beszélhetünk csúcsokon értelmezett súlyokról $x : V \rightarrow \mathbb{R}$, a jelölés módja esetében $x \in \mathbb{R}^n$, ahol $x(i)$ -vel, vagy x_i -vel jelöljük az i . csúcs súlyát. Az éleken értelmezett súlyoknál $w : E \rightarrow \mathbb{R}_{\geq 0}$ pedig $(v_i, v_j) \in E$ él súlya $w_{ij} \geq 0$. Két csúcs távolsága alatt az őket összekötő legrövidebb út súlyozatlan hosszát értjük és $d(u, v)$ -vel jelöljük. Egy $v \in V$ csúcs szomszédok halmazára $N(v) = \{w \in V : (v, w) \in E\}$, továbbá egy csúcs szomszédainak számát $\deg(v) = |N(v)|$ -vel jelöljük.

2.1.1. Definíció. Jelölje $W \in \mathbb{R}^{n \times n}$ a gráf súlyozott adjacencia mátrixát, amelyre teljesül, hogy

$$W_{ij} = \begin{cases} w_{ij} & (i, j) \in E, \\ 0 & \text{különben.} \end{cases}$$

2.1.2. Definíció. $\mathcal{L} = D - W \in \mathbb{R}^{n \times n}$ a gráf Laplace-mátrixa, ahol D diagonálmátrix, elemei pedig $D_{ii} = \sum_{j \in N(i)} w_{ij} = w_i$.

A Laplace-mátrixra tekintetünk úgy, mint egy $\mathcal{L} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ operátor amire tetszőleges $x \in \mathbb{R}^n$ csúcsokon értelmezett súlyfüggvény esetén fennáll

$$(\mathcal{L}x)(i) = \sum_{j \in N(i)} w_{ij}[x(i) - x(j)].$$

$W \equiv \mathbf{1}$ esetén visszakapjuk a súlyozatlan Laplace-mátrixot

$$\mathcal{L}_{ij} = \begin{cases} \deg(v_i) & i = j, \\ -1 & (i, j) \in E, \\ 0 & (i, j) \notin E. \end{cases}$$

2.1.3. Definíció. $\mathcal{L}_n = D^{-\frac{1}{2}}\mathcal{L}D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ a normalizált Laplace-mátrix

$$(\mathcal{L}_n)_{ij} = \begin{cases} 1 & i = j, \\ \frac{-w_{ij}}{\sqrt{w_i w_j}} & (i, j) \in E, \\ 0 & (i, j) \notin E. \end{cases}$$

Hasonlóan $\mathcal{L}_n : \mathbb{R}^n \rightarrow \mathbb{R}^n$ operátor, melynél tetszőleges $x \in \mathbb{R}^n$ súlyfüggvény esetén fennáll

$$(\mathcal{L}_n x)(i) = \frac{1}{\sqrt{w_i}} \sum_{j \in N(i)} w_{ij} \left[\frac{x(i)}{\sqrt{w_i}} - \frac{x(j)}{\sqrt{w_j}} \right].$$

2.1.4. Megjegyzés. A Laplace-mátrix szimmetrikus, azaz $\mathcal{L} = \mathcal{L}^\top$.

2.1.5. Definíció. $\mathcal{S} \in \mathbb{R}^{n \times m}$ a gráf súlyozott incidencia mátrixa, ahol

$$\mathcal{S}_{ij} = \begin{cases} \pm\sqrt{w_{ij}} & v_i \in e_j, \\ 0 & \text{különben,} \end{cases}$$

úgy, hogy minden oszlopban pontosan egy negatív és pozitív szám szerepel. Továbbá legyen $\mathcal{S}_n = D^{-\frac{1}{2}}\mathcal{S}$ a normalizált incidencia mátrix.

2.1.6. Megjegyzés. $\mathcal{L} = \mathcal{S}\mathcal{S}^\top$ és $\mathcal{L}_n = \mathcal{S}_n\mathcal{S}_n^\top$.

2.1.7. Definíció. Az $A \in \mathbb{C}^{n \times n}$ mátrix pozitív szemidefinit, ha minden $x \in \mathbb{R}^n$ vektorra teljesül, hogy $\langle x, Ax \rangle \geq 0$.

2.1.8. Megjegyzés. A skalárszorzatot komplex értelemben tekintjük, azaz $\langle x, y \rangle = \overline{\langle y, x \rangle}$.

2.1.9. Állítás. Ha $A \in \mathbb{R}^{n \times n}$ szimmetrikus mátrix, akkor minden sajátértéke valós és a különböző sajátvektorok ortogonálisak.

Bizonyítás. Legyen $Av = \lambda v$, $v \neq 0$. Ekkor

$$\lambda \langle v, v \rangle = \langle \lambda v, v \rangle = \langle Av, v \rangle = \langle v, Av \rangle = \langle v, \lambda v \rangle = \bar{\lambda} \langle v, v \rangle.$$

Tehát $\lambda = \bar{\lambda} \in \mathbb{R}$. Továbbá legyen $Av_1 = \lambda_1 v_1$ és $Av_2 = \lambda_2 v_2$, $v_1, v_2 \neq 0$ különböző sajátértékekhez tartozó sajátvektorok. Így kapjuk, hogy

$$\lambda_1 \langle v_1, v_2 \rangle = \langle \lambda_1 v_1, v_2 \rangle = \langle Av_1, v_2 \rangle = \langle v_1, Av_2 \rangle = \langle v_1, \lambda_2 v_2 \rangle = \lambda_2 \langle v_1, v_2 \rangle.$$

Ez csak úgy lehetséges, ha $\langle v_1, v_2 \rangle = 0$. □

2.1.10. Állítás. Ha $A \in \mathbb{R}^{n \times n}$ szimmetrikus, akkor n darab különböző sajátvektora van, amelyek \mathbb{R}^n ortonormált bázisát alkotják.

Bizonyítás. Tegyük fel, hogy már megtaláltuk A -nak u_1, u_2, \dots, u_k k darab ortogonális normált sajátvektorát és a hozzá tartozó sajátértékeket $\lambda_1, \lambda_2, \dots, \lambda_k$. Legyen $U = [u_1, u_2, \dots, u_k] \in \mathbb{R}^{n \times k}$, ekkor $AU = [\lambda_1 u_1, \dots, \lambda_k u_k] \in \mathbb{R}^k$. Egészítsük ki U -t \mathbb{R}^n ortonormált bázisává: $V = [v_{k+1}, \dots, v_n] \in \mathbb{R}^{n \times (n-k)}$. Világos, hogy $U^\top V = 0$. Most legyen w sajátvektora $V^\top AV$ mátrixnak, μ sajátértékkel, azaz $V^\top AVw = \mu w$. Definiáljuk $u_{k+1} := Vw$ vektort. Megfelelő konstanssal beszorozva elérhetjük, hogy $\|u_{k+1}\| = 1$ teljesüljön. $U^\top u_{k+1} = U^\top Vw = 0w = 0$, tehát u_{k+1} ortogonális U -ra. Végül pedig, megmutatjuk, hogy u_{k+1} sajátvektora A -nak. $d = Au_{k+1} - \mu u_{k+1}$, elég belátni, hogy $d = 0$.

$$V^\top d = V^\top (Au_{k+1} - \mu u_{k+1}) = V^\top (AVw - \mu Vw) = V^\top AVw = \mu w = 0.$$

Tehát d benne van U által kifeszített altérben, azaz $d = Ur$ valamilyen $0 \neq r \in \mathbb{R}^k$ vektorra.

$$r = U^\top Ur = U^\top d = U^\top AVw - \mu U^\top Vw = 0.$$

Amiből következik, hogy $d = 0$. □

2.1.11. Állítás (SVD felbontás). Az $A \in \mathbb{R}^{n \times n}$ szimmetrikus mátrix előáll $A = UDU^\top$ alakban, ahol $U = [u_1, \dots, u_n]$ tartalmazza az A mátrix sajátvektorait és $D = \text{diag}(\lambda_i)$ diagonálmátrix diagonálelemei pedig rendre a sajátvektorokhoz tartozó sajátértékek.

Bizonyítás. A (2.1.9) állítás értelmében $U^\top U = I$, speciálisan $\lambda_i U^\top u_i = \lambda_i e_i$, ahol e_i jelöli a standard i . bázisvektort.

$$U^\top AU = U^\top A[u_1, \dots, u_n] = U^\top [\lambda_1 u_1, \dots, \lambda_n u_n] = [\lambda_1 e_1, \dots, \lambda_n e_n] = D.$$

Ebből következik, hogy $A = (U^\top)^{-1} D U^{-1} = U D U^\top$. □

2.1.12. Állítás. Ha $A \in \mathbb{R}^{n \times n}$ szimmetrikus pozitív szemidefinit, akkor minden sajátértéke nemnegatív.

Bizonyítás. $0 \leq \langle v, Av \rangle = \langle v, \lambda v \rangle = \lambda \langle v, v \rangle = \lambda \|v\|^2$. □

2.1.13. Állítás. A Laplace-mátrix pozitív szemidefinit mátrix.

Bizonyítás. $\langle x, \mathcal{L}x \rangle = \langle x, \mathcal{S}\mathcal{S}^\top x \rangle = \langle \mathcal{S}^\top x, \mathcal{S}^\top x \rangle = \|\mathcal{S}^\top x\|^2 \geq 0$. □

2.1.14. Következmény. A Laplace-mátrixnak n darab nemnegatív sajátértéke van és a hozzá tartozó sajátvektorok ortogonális bázist alkotnak.

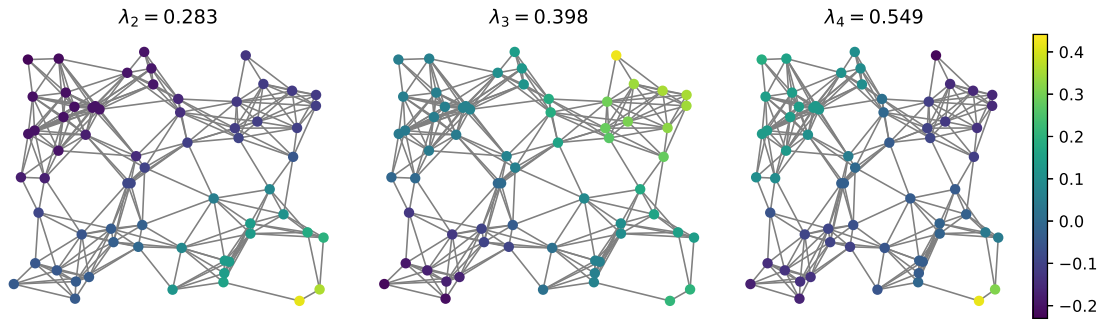
2.2. Spektrum meghatározása

2.2.1. Definíció. A $\mathcal{G} = (V, E, W)$ élsúlyozott összefüggő gráf normalizált Laplace-mátrixához tartozó sajátértékeket

$$0 \leq \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$$

gráf spektrumának nevezzük és $\sigma(\mathcal{L}_n)$ -nel jelöljük.

Mivel a Laplace-mátrix szimmetrikus pozitív szemidefinit, elkészíthetjük az SVD felbontását. A továbbiakban a következő jelölésekkel fogunk élni: $\mathcal{L}_n = \Phi \Lambda \Phi^\top$ ahol $\Phi = [\phi_0, \phi_1, \dots, \phi_{n-1}]$ sajátvektorok és $\Lambda = \text{diag}([\lambda_0, \lambda_1, \dots, \lambda_{n-1}])$ tartalmazza a sajátértékeket.



1. ábra. ϕ_2, ϕ_3, ϕ_4 sajátvektorok súlyozatlan gráfra

2.2.2. Állítás. $\lambda_0 = 0$ és $\phi_0 = [\sqrt{w_1}, \dots, \sqrt{w_n}] = D^{\frac{1}{2}} \mathbf{1}$.

2.2.3. Állítás. $\lambda_{n-1} \leq 2$, illetve $\lambda_{n-1} = 2$ akkor és csak akkor, ha \mathcal{G} páros gráf.

Bizonyítás. Először megmutatjuk, hogy az $I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ mátrix pozitív szemidefinit. Legyen $0 \neq x \in \mathbb{R}^n$,

$$x^\top (I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) x = \sum_{i \in V} x(i)^2 + \sum_{(i,j) \in E} \frac{2x(i)x(j)}{\sqrt{w_i w_j}} = \sum_{(i,j) \in E} \left[\frac{x(i)}{\sqrt{w_i}} + \frac{x(j)}{\sqrt{w_j}} \right]^2 \geq 0.$$

Ezt átrendezve és hozzáadva $x^\top x$ -et, majd $x = \phi_{n-1}$ választással élve

$$\begin{aligned} -x^\top D^{-\frac{1}{2}} W D^{-\frac{1}{2}} x &\leq x^\top x \\ x^\top I x - x^\top D^{-\frac{1}{2}} W D^{-\frac{1}{2}} x &\leq 2x^\top x \\ \frac{x^\top \mathcal{L}_n x}{x^\top x} = \frac{x^\top \lambda_n x}{x^\top x} &= \lambda_n \leq 2. \end{aligned}$$

Akkor és csak akkor áll fent mindenhol egyenlőség, ha minden élre $\sqrt{w_j}x(i) = -\sqrt{w_i}x(j)$, ami pontosan akkor teljesül, ha \mathcal{G} -nek létezik két csúcsosztálya, amik között nem halad él és az egyik osztályon x negatív, a másikon pedig pozitív, azaz a gráf páros. \square

A fent használt törtet szokás Rayleigh-hányadosnak is nevezni. Ennek segítségével iteratív módon meg tudjuk adni a többi sajátértéket is.

2.2.4. Tétel (Rayleigh). $A \in \mathbb{R}^{n \times n}$ szimmetrikus, $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ sajátértékekkel és a hozzájuk tartozó ortonormált sajátvektorok rendre u_1, u_2, \dots, u_n . Továbbá jelölje $S = \text{span}(u_{i_1}, \dots, u_{i_k})$ sajátvektorok által kifeszített alteret, $1 \leq i_1 \leq \dots \leq i_k \leq n$, ekkor $0 \neq x \in \mathbb{C}^n$ vektorra teljesül hogy

$$\lambda_{i_1} = \min_{x \in S} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \min_{\substack{x \in S \\ \|x\|_2=1}} \langle x, Ax \rangle \leq \max_{x \in S} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \max_{\substack{x \in S \\ \|x\|_2=1}} \langle x, Ax \rangle = \lambda_{i_k}.$$

Továbbá $\lambda_{i_1} \leq \langle x, Ax \rangle \leq \lambda_{i_k}$ teljesül minden $x \in S_k$ egységvektorra, egyenlőség pedig akkor és csak akkor áll fent a megfelelő oldalakon, ha $Ax = \lambda_{i_1}x$, illetve ha $Ax = \lambda_{i_k}x$.

Bizonyítás. Tetszőleges $x \in S$ egységvektor felírható $x = \sum_{j=1}^k \alpha_j u_{i_j}$ alakban. Az ortogonalitást felhasználva

$$1 = \langle x, x \rangle = \sum_{j,l=1}^k \alpha_j \bar{\alpha}_l \langle u_{i_j}, u_{i_l} \rangle = \sum_{j=1}^k |\alpha_j|^2.$$

Kibontva a skalárszorzatot

$$\langle x, Ax \rangle = \left\langle \sum_{j=1}^k \alpha_j u_{i_j}, \sum_{j=1}^k \alpha_j \lambda_{i_j} u_{i_j} \right\rangle = \sum_{j=1}^k |\alpha_j|^2 \lambda_{i_j},$$

látszik, hogy az eredmény éppen a sajátértékek $|\alpha_j|^2$ skalárokkal vett konvex kombinációja. Tehát teljesül hogy

$$\lambda_{i_1} \leq \sum_{j=1}^k |\alpha_j|^2 \lambda_{i_j} \leq \lambda_{i_k}.$$

Egyenlőség jobb oldalon és csak akkor teljesül, ha $\lambda_{i_j} \neq \lambda_{i_k}$ együtthatója 0, azaz $x = u_{i_k}$ sajátvektor. Hasonlóan a bal oldali egyenlőség $x = u_{i_1}$ vektorral adódik. \square

2.2.5. Tétel (Courant-Fischer). $A \in \mathbb{R}^{n \times n}$ szimmetrikus, $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ sajátértékekkel, jelölje \mathbb{R}^n egy valódi alterét S . Ekkor $0 \neq x \in S$ vektorra teljesül, hogy

$$\lambda_k = \min_{\dim S=k} \max_{x \in S} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \max_{\dim S=n-k+1} \min_{x \in S} \frac{\langle x, Ax \rangle}{\langle x, x \rangle}.$$

Bizonyítás. Legyenek u_1, u_2, \dots, u_n sajátvektorok, melyek \mathbb{R}^n ortonormált bázisát alkotják, S_k egy k -imenziós altér, $S' = \text{span}(u_k, \dots, u_n)$. Ekkor $S \cap S' \neq \emptyset$. Kihasználva, hogy $\emptyset \neq K_1 \subset K_2$, akkor $\sup_{x \in K_2} f(x) \geq \sup_{x \in K_1} f(x) \geq \inf_{x \in K_1} f(x) \geq \inf_{x \in K_2} f(x)$, a következőt kapjuk

$$\sup_{x \in S_k} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \geq \sup_{x \in S_k \cap S'} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \geq \inf_{x \in S_k \cap S'} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \geq \inf_{x \in S'} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \min_{x \in S'} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \lambda_k.$$

Itt az utolsó két egyenlőségnél használtuk ki a Rayleigh-tételt, hogy S' -n az infimum fel is vétetik, mégpedig u_k sajátvektoron. Ezek után az egészből infimumot véve,

$$\inf_{\substack{S_k \\ \dim S_k = k}} \sup_{x \in S_k} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} \geq \lambda_k,$$

majd $S_k = \text{span}(u_1, \dots, u_k)$ és $x = u_k$ választással élve a Rayleigh-tétel szerint valójában a megfelelő minimum és maximum is felvétetik

$$\inf_{\substack{S_k \\ \dim S_k = k}} \sup_{x \in S_k} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \min_{\substack{S_k \\ \dim S_k = k}} \max_{x \in S_k} \frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \lambda_k.$$

Hasonló módon megkaphatjuk a tételben szereplő második egyenlőséget, ha $-A$ mátrixra mondjuk végig az előzőeket, mivel ha λ_i sajátértéke A -nak, akkor $-\lambda_i$ sajátértéke az ellentetjének. \square

2.3. Gráfkonvolúció

Az $f, g : \mathbb{R} \rightarrow \mathbb{R}$ függvények konvolúcióján a alábbi függvényt értjük, amennyiben az integrál értelmes

$$(f * g)(x) = \int_{\mathbb{R}} f(t)g(x-t) dt.$$

Célunk értelmezni egy \mathcal{G} gráf csúcsain adott $f, g : V \rightarrow \mathbb{R}$ függvények konvolúcióját. A klasszikus definíciót közvetlenül azonban nem tudjuk használni, ugyanis nem világos hogy a $g(x-t)$ valós számokon vett eltolás művelet alatt mit értsünk. A megoldást a konvolúciós tétel biztosítja, amely szerint két függvény konvolúciója kifejezhető a Fourier transzformáltak segítségével

$$(f * g) = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\},$$

itt $\mathcal{F} : f \rightarrow \hat{f}$, ahol $\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle$ a Fourier-transzformált, és $\mathcal{F}^{-1}\{\mathcal{F}\{f\}\} = f$ jelöli az inverz transzformált. Megjegyezzük, hogy az egy dimenziós Laplace-operátor sajátfüggvényei éppen a Fourier-transzformáltban szereplő exponenciális függvények

$$-\Delta(e^{2\pi i \xi t}) = -\frac{\partial^2}{\partial t^2} e^{2\pi i \xi t} = (2\pi \xi)^2 e^{2\pi i \xi t}.$$

Az előzőek analógiájára a Laplace-mátrix segítségével tudjuk deifinálni a gráf csúcsain értelmezett függvények Fourier-transzformáltját és konvolúcióját.

2.3.1. Definíció. A $\mathcal{G} = (V, E, W)$ gráf csúcsain értelmezett $x : V \rightarrow \mathbb{R}$ függvény Fourier-transzformáltja

$$\hat{x}(\lambda_j) = \langle x, \phi_j \rangle = \sum_{i=1}^n x(i) \phi_j(i).$$

Az inverz Fourier-transzformált pedig

$$x(i) = \sum_{j=0}^{n-1} \hat{x}(\lambda_j) \phi_j(i).$$

A Laplace-mátrix SVD felbontása segítségével kifejezve: $\hat{x} = \Phi^\top x$ és $x = \Phi \hat{x}$. Tehát egy függvényt ha kifejezünk a sajátvektorok lineáris kombinációjaként, akkor a megfelelő együtthatók adják a Fourier transzformáltat. Az $\hat{x} : \sigma(\mathcal{L}_n) \rightarrow \mathbb{R}$ leképezés az x függvény frekvencia-tartománybeli reprezentációjaként fogható fel. Az ortogonalitás miatt $\Phi \Phi^\top x = x$, tehát a Fourier-transzformált invertálható.

2.3.2. Definíció. A $\mathcal{G} = (V, E, W)$ gráf csúcsain értelmezett $x, g : V \rightarrow \mathbb{R}$ függvények konvolúcióján a következő kifejezést értjük

$$(x * g)(i) = \sum_{j=0}^{n-1} \hat{x}(\lambda_j) \hat{g}(\lambda_j) \phi_j(i).$$

Áttérve a mátrixokra és \circ -rel jelölve a vektorok koordinátánkénti szorzatát, a konvolúciós tételnek megfelelő formulát kapjuk

$$(x * g) = \Phi((\Phi^\top x) \circ (\Phi^\top g)).$$

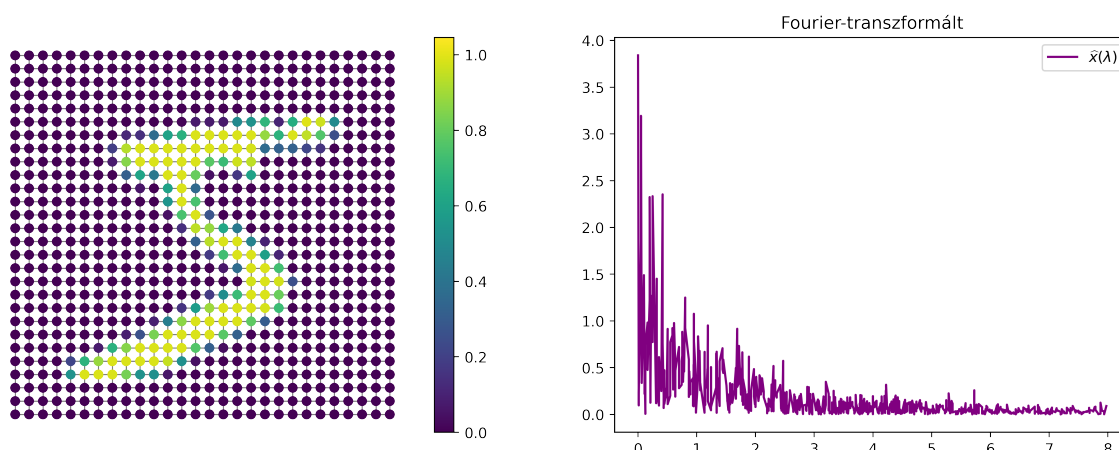
Ez azt jelenti, hogy az x függvényre alkalmazunk valamilyen g filter operációt a frekvencia tartományon, majd az inverz transzformációval újra $(x * g) : V \rightarrow \mathbb{R}$ függvényt kapunk.

2.3.3. Állítás (Parseval). $x, y : V \rightarrow \mathbb{R}$ függvényekre $\langle x, y \rangle = \langle \hat{x}, \hat{y} \rangle$ és $\sum_{i=1}^n |x(i)|^2 = \sum_{i=0}^{n-1} |\hat{x}(\lambda_i)|^2$.

Bizonyítás. Φ ortogonalitásából rögtön adódik, hogy

$$\begin{aligned} \langle \hat{x}, \hat{y} \rangle &= (\Phi^\top x)^\top \Phi^\top y = x^\top \Phi \Phi^\top y = x^\top y = \langle x, y \rangle \\ \sum_{i=1}^n |x(i)|^2 &= \|x\|^2 = \langle x, x \rangle = \langle \hat{x}, \hat{x} \rangle = \|\hat{x}\|^2 = \sum_{i=0}^{n-1} |\hat{x}(\lambda_i)|^2. \end{aligned}$$

□



2. ábra. MNIST adathalmaz egy eleme 28x28-as rácsgráfon és Fourier-transzformáltja

2.3.4. Állítás. Legyen $f, g, h : V \rightarrow \mathbb{R}$ függvények és $\alpha \in \mathbb{R}$ ekkor a konvolúcióra igazak az alábbiak

- 1) $\widehat{f * g} = \widehat{f} \widehat{g}$,
- 2) $\alpha(f * g) = (\alpha f) * g = f * (\alpha g)$,
- 3) $f * g = g * f$,
- 4) $f * (g + h) = f * g + f * h$,
- 5) $(f * g) * h = f * (g * h)$.

2.3.5. Állítás. $\delta(i) = \sum_{j=1}^{n-1} \phi_j(i)$ a konvolúció multiplikatív egysége, azaz, minden $f : V \rightarrow \mathbb{R}$ függvényre $f * \delta = f$.

Bizonyítás.

$$\widehat{\delta}(\lambda_j) = \langle \delta, \Phi_j \rangle = \left\langle \sum_{j=1}^{n-1} \phi_j, \phi_j \right\rangle = 1$$

$$(f * \delta)(i) = \sum_{j=0}^{n-1} \widehat{f}(\lambda_j) \widehat{\delta}(\lambda_j) \phi_j(i) = \sum_{j=0}^{n-1} \widehat{f}(\lambda_j) \phi_j(i) = f(i).$$

□

A továbbiakban azzal fogunk foglalkozni, hogyan érdemes a g vagy \widehat{g} függvényeket megválasztani, illetve hogyan tudnánk kiszámolni a konvolúciót. Először is, \mathcal{L}_n SVD felbontása meglehetősen drága nagy gráfok esetén, de még ha ezt ki is tudnánk számolni, a sajátvektorokkal és sajátértékekkel való mátrixszorzás művelei igénye $\mathcal{O}(n^2)$. A probléma megoldásához alakítsuk át a konvolúciós képletet. Ehhez tegyük fel, hogy g függvény hatványsorba fejthető.

2.3.6. Definíció. Legyen $f(x) = \sum_{n=0}^{\infty} a_n x^n$ hatványsorba fejthető és $A \in \mathbb{R}^{n \times n}$ négyzetes mátrix. Ekkor az $f(A)$ mátrixfüggvényt a hatványsor segítségével értelmezzük $f(A) = \sum_{n=0}^{\infty} a_n A^n$.

2.3.7. Állítás. Ha f hatványsorba fejthető és a sor konvergens $|x| < r$ -en, akkor $f(A)$ is konvergens, amennyiben $\|A\| < r$ valamilyen szubmultiplikatív ($\|AB\| \leq \|A\| \|B\|$) mátrix normára.

2.3.8. Következmény. Ha $A \in \mathbb{R}^{n \times n}$ diagonalizálható, $A = PDP^{-1}$, akkor $f(A) = P[\text{diag}(f(d_1), \dots, f(d_n))]P^{-1} = Pf(D)P^{-1}$ alakban írható fel.

Ez előzőek alapján a konvolúcióra igaz, hogy

$$(x * g) = \Phi((\Phi^\top x) \circ (\widehat{g})) = \Phi \text{diag}(\widehat{g}(\lambda_0), \dots, \widehat{g}(\lambda_{n-1}))\Phi^\top x = \Phi \widehat{g}(\Lambda)\Phi^\top x = \widehat{g}(\mathcal{L}_n)x.$$

A \widehat{g} mátrix függvényt filternek nevezzük. Az előzőeket felhasználva a klasszikus konvolúciós háló (CNN) mintájára definiáljuk a gráf-konvolúciós réteget (GCNN).

2.3.9. Definíció (Gráf Konvolúciós Réteg). Jelölje q és q' rendre az input és output vektorok számát, $x_i \in \mathbb{R}^{|V|}$ $i \in \{1, \dots, q\}$ jelöli az input vektorokat, $\widehat{x}_{i'} \in \mathbb{R}^{|V|}$ $i' \in \{1, \dots, q'\}$ jelöli az output vektorokat. $\xi : \mathbb{R} \rightarrow \mathbb{R}$ az aktivációs függvény, amelyet csúcsenként alkalmazunk. Ekkor konvolúciós rétegnek nevezzük a következő formulát

$$\widehat{x}_{i'} = \xi \left(\sum_{i=1}^{q'} g_{i'}(\mathcal{L}_n)x_i \right) \quad i' \in \{1, \dots, q'\}.$$

Tekintsük az alábbi k -ad fokú, $\Theta \in \mathbb{R}^{k+1}$ paraméterű polinomiális filtert

$$\widehat{g}(\mathcal{L}_n) = \sum_{i=0}^k \theta_i \mathcal{L}_n^i.$$

Ennek a számolása továbbra is drága mátrixszorzásokkal jár, viszont nagy előnye, hogy lokális filter, azaz egy $v \in V$ csúcs értékét pontosan a tőle k távolságra lévő csúcsok fogják befolyásolni.

2.3.10. Állítás. Legyen $\mathcal{G} = (V, E, W)$ gráf, \mathcal{L} a Laplace-mátrixa és $s > 0$ egész. Ha két $u, v \in V$ csúcsra $d(u, v) > s$, akkor $(\mathcal{L}^s)_{uv} = 0$.

Bizonyítás. Elvégezve a mátrixszorzást

$$(\mathcal{L}^s)_{uv} = \sum \mathcal{L}_{uk_1} \mathcal{L}_{k_1 k_2} \cdots \mathcal{L}_{k_{s-1} v},$$

ahol a szumma az a megfelelő $s - 1$ hosszú k_i sorozatokon fut végig. Tegyük fel, hogy $(\mathcal{L}^s)_{uv} \neq 0$, azaz létezik egy tag az összegben ami nem 0. Ez csak úgy lehetséges, ha létezik egy $s - 1$ hosszú sorozat amire $\mathcal{L}_{uk_1} \neq 0, \mathcal{L}_{k_1 k_2} \neq 0 \dots \mathcal{L}_{k_{s-1} v} \neq 0$, azaz ha van maximum s hosszú út u -ból v -be. Mivel a feltétel szerint ilyen nincs, ellentmondásra jutottunk. \square

2.3.11. feladat (Képrekonstrukció). Tekintsünk egy $n \times n$ pixelből álló képet és a hozzá tartozó $\mathcal{G} = (V, E, W, x)$ rácsgráfot, ahol $|V| = n^2$. A csúcsokat két koordináta határozza meg, $[i, j]$ amennyiben $i, j \leq n$. Egy előre rögzített K számra az élek súlyait a következőképpen adjuk meg

$$W_{uv} = \exp\left(-\frac{\|z_u - z_v\|_2^2}{\sigma^2}\right) \quad \text{ha } \|z_u - z_v\|_2^2 \leq K,$$

ahol z_u az u csúcs 2D koordinátáit jelöli. A rácson értelmezett képre úgy tekintünk, mint a csúcsokon $x : V \rightarrow [0, 1]$ függvény. ahol $x([i, j])$ mondja meg hogy az adott pixel mennyire sötét. Legyen $x_0 = x + \eta$, ahol $\eta([i, j]) \sim \mathcal{N}(0, \sigma^2)$ normális eloszlású zajfüggvény minden csúcsra független. A feladat rekonstruálni x -et. Legyen $\tau > 0$ konstans, keressük $f : V \rightarrow [0, 1]$ függvényt, amelyre

$$\min_f \{\|f - x_0\|_2^2 + \tau \langle f, \mathcal{L}f \rangle\}.$$

itt \mathcal{L} a rácsgráf Laplace mátrixa, $\tau \langle f, \mathcal{L}f \rangle$ pedig azért felelős, hogy a kapott f sima legyen olyan értelemben, hogy a szomszédos csúcsok között kicsi legyen f eltérése. Valóban, $\langle f, \mathcal{L}f \rangle$ minimuma éppen akkor vétetik fel ha f konstans, azaz a szomszédos csúcson a függvény eltérése 0, ugyanis

$$\langle f, \mathcal{L}f \rangle = \frac{1}{2} \sum_{i \in V} \sum_{j \in N(i)} w_{ij} (f(i) - f(j))^2.$$

Tehát a célfüggvény minimalizálása szemléletesen azt jelenti, hogy a zajos képet, kismítjük. A norma folytonossága és konvexitása miatt, valamint a Laplace-mátrix pozitív szemidefinitése miatt a célfüggvény konvex, azaz a globális minimuma ott vétetik fel, ahol az f szerinti parciális deriváltak eltűnnek.

$$0 = \nabla_f [\|f - x_0\|_2^2 + \tau \langle f, \mathcal{L}f \rangle] = 2(f - x_0) + 2\tau \mathcal{L}f$$

A Fourier-transzformáltat véve

$$0 = 2(\widehat{f}(\lambda_i) - \widehat{x}_0(\lambda_i)) + 2\tau \widehat{\mathcal{L}f}(\lambda_i).$$

Kihasználva \mathcal{L} szimmetrikus jellegét és hogy $\mathcal{L}\phi_i = \lambda_i \phi_i$ kapjuk hogy

$$\widehat{\mathcal{L}f}(\lambda_i) = \langle \mathcal{L}f, \phi_i \rangle = \langle f, \mathcal{L}\phi_i \rangle = \langle f, \lambda_i \phi_i \rangle = \lambda_i \langle f, \phi_i \rangle = \lambda_i \widehat{f}(\lambda_i).$$

Átrendezve és alkalmazva az inverz transzformációt kapjuk, hogy

$$\widehat{f}(\lambda_i) = \frac{\widehat{x}_0(\lambda_i)}{1 + \tau \lambda_i}, \quad f(i) = \sum_{j=0}^{n-1} \left[\frac{1}{1 + \tau \lambda_i} \right] \widehat{x}_0(\lambda_i) \phi_j(i).$$

Az eredeti kép rekonstrukcióját megkapjuk x_0 és g konvolúciójából, ahol $\widehat{g}(\lambda_i) = \frac{1}{1 + \tau \lambda_i}$ és $(g * x_0) = \widehat{g}(\mathcal{L})x_0$.

3. Gráf filterek

Ebben fejezetben konkrét konstrukciókat és példákat mutatunk $\hat{g}(\mathcal{L}_n)x$ gyors kiszámítására. Először a Csebisev-polinomokkal definiált filtereket tekintjük át az [5] és [6] alapján. A Csebisev-polinomok fontosabb tulajdonságainak megértéséhez a [7] könyvet használtam. A Cayley és ARMA filterekhez a [8] és [9] használtam. A 4.4.-es alfejezetben egy Spline-okon alapuló filterezési technikát mutatok be [10] alapján. Ez abban tér el az előbbiektől, hogy nem használ spektrálgráfelméletet, csupán a gráf éleinek elhelyezkedését veszi figyelembe. A Spline függvényekhez tartozó ismereteket [11] és [12] alapján írtam.

3.0.1. Definíció. Egy $\hat{g} : [\lambda_{\min}, \lambda_{\max}] \rightarrow \mathbb{R}$ függvény által definiált gráf filter alatt a

$$\sum_{i=1}^n \phi_i \hat{g}(\lambda_i) \phi_i^\top = \Phi \hat{g}(\Lambda) \Phi^\top$$

kifejezést értjük.

3.1. Csebisev-polinomok

3.1.1. Definíció. A Csebisev-polinomokat a következő rekurzióval definiáljuk

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned}$$

3.1.2. Megjegyzés. $T_n(x)$ főegyütthatója 2^{n-1} .

3.1.3. Állítás. $T_n(\cos(\theta)) = \cos(n\theta)$, azaz $T_n(x) = \cos(n \cdot \arccos(x))$, ahol $|x| \leq 1$.

Bizonyítás. Teljes indukcióval bizonyítunk. $n = 0$ -ra $T_0(x) = \cos(0) = 1$.

$$\begin{aligned} T_{n+1}(\cos \theta) &= 2 \cos(\theta) T_n(\cos \theta) - T_{n-1}(\cos \theta) \\ &= 2 \cos(\theta) \cos(n\theta) - \cos((n-1)\theta) \\ &= 2 \cos(\theta) \cos(n\theta) - \cos(n\theta) \cos(\theta) - \sin(n\theta) \sin(\theta) \\ &= 2 \cos(\theta) \cos(n\theta) - \sin(n\theta) \sin(\theta) = \cos(n\theta). \end{aligned}$$

□

3.1.4. Következmény. Amennyiben $|x| \leq 1$, akkor $T_n(x) \in [-1, 1]$ korlátos függvények.

3.1.5. Definíció. Azt mondjuk, hogy $f \in L^2(I, \omega)$, ha f értelmezési tartománya I , és

$$\|f\| = \int_I \omega(x) [f(x)]^2 dx < \infty.$$

3.1.6. Állítás. A Csebisev-polinomok ortogonálisak az $L^2([-1, 1], dy/\sqrt{1-y^2})$ térben.

Bizonyítás. Legyen $m \neq n \neq 0$. $x = \cos(\theta)$ helyettesítést használva

$$\begin{aligned} \int_{-1}^1 \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx &= \int_0^\pi \frac{\cos(n\theta)\cos(m\theta)}{\sqrt{1-\cos(\theta)^2}} \sin(\theta) d\theta = \int_0^\pi \cos(n\theta)\cos(m\theta) d\theta \\ &= \int_0^\pi \frac{1}{2} \cos((n+m)\theta) + \cos((m-n)\theta) d\theta = \frac{1}{2} \left[\frac{\sin(n+m)\theta}{n+m} + \frac{\sin(m-n)\theta}{m-n} \right]_0^\pi = 0. \end{aligned}$$

Ha $n = m \neq 0$, akkor a fenti helyettesítéssel

$$\int_{-1}^1 \frac{T_n^2(x)}{\sqrt{1-x^2}} dx = \int_0^\pi \cos^2(n\theta) d\theta = \int_0^\pi \frac{1}{2} (\cos(2n\theta) + 1) d\theta = \left[\frac{\sin(2n\theta)}{4n} + \frac{\theta}{2} \right]_0^\pi = \frac{\pi}{2}.$$

Ha $n = m = 0$, akkor

$$\int_{-1}^1 \frac{T_0^2(x)}{\sqrt{1-x^2}} dx = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} = \pi.$$

□

3.1.7. Megjegyzés. Különböző súlyfüggvények esetén különböző ortogonális polinomokhoz jutunk. Az $\omega(x) = 1$ esetén megkapjuk a $(n+1)Q_{n+1}(x) = (2n+1)xQ_n(x) - nQ_{n-1}(x)$ rekurzióval definiált Legendre-polinomokat.

3.1.8. Tétel. A Csebisev-polinomok a $[-1, 1]$ -en értelmezett polinomok bázisát alkotják.

Bizonyítás. A bizonyításhoz szükségünk a trigonometrikus függvények komplex alakjára

$$x = \cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta}).$$

Ennek segítségével felírható, hogy

$$T_n(x) = \cos(n\theta) = \frac{1}{2} (e^{in\theta} + e^{-in\theta}), \quad x^n = \frac{1}{2^n} (e^{i\theta} + e^{-i\theta})^n.$$

A binomiális tételt használva

$$x^n = \frac{1}{2^n} \sum_{k=0}^n \binom{n}{k} e^{ik\theta} e^{-i(n-k)\theta} = \frac{1}{2^n} \sum_{k=0}^n \binom{n}{k} e^{i(2k-n)\theta}.$$

Összevonva a k . és $(n-k)$. tagot

$$\binom{n}{k} e^{i(2k-n)\theta} + \binom{n}{n-k} e^{-i(2k-n)\theta} = 2 \binom{n}{k} T_{n-2k}(x).$$

Páratlan n esetén a szummában páros sok tag van, tehát

$$x^{2n+1} = \frac{1}{2^{2n}} \sum_{k=0}^n \binom{2n+1}{k} T_{2n+1-2k}(x).$$

Páros n -re a szummában páratlan sok tag van, azaz

$$x^{2n} = \frac{1}{2^{2n}} \binom{2n}{n} T_0(x) + \frac{1}{2^{2n-1}} \sum_{k=0}^{n-1} \binom{2n}{k} T_{2n-2k}(x).$$

Tehát bármely n -ed fokú p polinom előáll a következő alakban: $p(x) = \sum_{k=0}^n \alpha_k T_k(x)$. \square

3.1.9. Következmény. Legyen $f(x) = \sum_{k=0}^{\infty} a_k x^k$ egyenletesen konvergens $[-1, 1]$ -en, minden x^k tagot fel tudjuk írni Csebisev-polinomok segítségével, ily módon elkészíthetjük

$$f(x) = \frac{1}{2} \alpha_0 + \sum_{k=1}^{\infty} \alpha_k T_k(x)$$

sort, ezt az f függvény Csebisev-sorának hívjuk. Világos, hogy ez szintén egyenletesen konvergens.

3.1.10. Következmény. A Csebisev-sor mindkét oldalát megszorozva $T_s(x)$ -szel, majd integrálva és kihasználva az ortogonalitást, meghatározhatóak az α_s együtthatók is

$$\int_{-1}^1 \frac{f(x) T_s(x)}{\sqrt{1-x^2}} dx = \alpha_s \int_{-1}^1 \frac{T_s^2(x)}{\sqrt{1-x^2}} dx = \alpha_s \frac{\pi}{2} \quad s > 0,$$

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx = \frac{1}{2} \alpha_0 \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx = \alpha_0 \frac{1}{2} [\arcsin(x)]_{-1}^{+1} = \alpha_0 \frac{\pi}{4} \quad s = 0.$$

A továbbiakban arra a kérdésre fogunk válaszolni, hogy ha a Csebisev-sor helyett csak véges $p_n(x) = \sum_{i=0}^n \alpha_i T_i(x)$ közelítő összeget vezetünk, akkor mekkora lesz az

$$\|f - p_n\|^2 = \int_{-1}^1 \frac{[f(x) - p(x)]^2}{\sqrt{1-x^2}} dx$$

eltérés. Feltesszük, hogy $f \in L^2([-1, 1], dy/\sqrt{1-y^2})$.

3.1.11. Állítás. Az $\|x^{n+1} - p(x)\|$ normát a legfeljebb n -ed fokú polinomok közül a $p(x) = x^{n+1} - \frac{1}{2^{n-1}}T_{n+1}(x)$ minimalizálja.

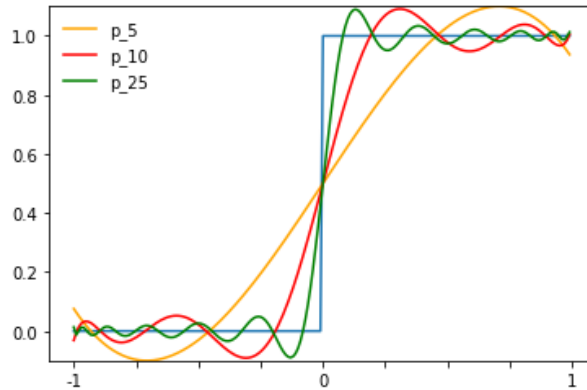
Bizonyítás. Mivel a Csebisev-polinomok $[-1, 1]$ -en értelmezett polinomok bázisát alkotják

$$x^{n+1} - p(x) = \frac{T_{n+1}(x)}{2^{n-1}} + \sum_{i=0}^n \gamma_i T_i(x).$$

Mivel a norma konvex és folytonos függvény, minimumát ott veszi fel ahol az együtthatókra vett parciális deriváltjai zérusok

$$\begin{aligned} 0 &= \frac{\partial \|x^{n+1} - p(x)\|^2}{\partial \gamma_i} = \int_{-1}^1 \frac{\partial}{\partial \gamma_i} \left[\frac{x^{n+1} - p(x)}{\sqrt{1-x^2}} \right]^2 dx = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} \frac{\partial}{\partial \gamma_i} \left[\sum_{j=0}^{n+1} \gamma_j T_j(x) \right]^2 dx \\ &= \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} 2 \left[\sum_{j=0}^{n+1} \gamma_j T_j(x) \right] T_i(x) dx = 2\gamma_i \int_{-1}^1 \frac{T_i(x)T_i(x)}{\sqrt{1-x^2}} dx = \gamma_i \pi. \end{aligned}$$

Azaz $\gamma_i = 0$ minden $i \leq n$ -re, tehát $p(x) = x^{n+1} - \frac{T_{n+1}(x)}{2^{n-1}}$. □



$f(x) = \chi_{[0,1]}(x)$ indikátorfüggvény közelítése Csebisev-polinomokkal $[-1, 1]$ intervallumon

3.1.12. Állítás. Az f függvény legjobb n -ed fokú közelítő polinomja éppen p_n .

Bizonyítás. Legyen $p = \sum_{i=0}^n a_i T_i$. Belátjuk, hogy f legjobb közelítése esetén $a_i = \alpha_i$. Mivel a norma folytonos és konvex, a minimum akkor vétetik fel, ha az a_i szerinti parciális deriváltak nullák

$$0 = \frac{\partial}{\partial a_i} \|f - p(x)\| = \int_{-1}^1 \frac{2(f(x) - p(x))T_i(x)}{\sqrt{1-x^2}} dx.$$

Kihasználva az ortogonalitást majd átrendezve kapjuk hogy

$$a_i = \frac{2}{\pi} \int_{-1}^1 \frac{f(x)T_i(x)}{\sqrt{1-x^2}} dx.$$

□

3.1.13. Megjegyzés. Az ortogonalitásból és (3.1.10)-ből következik, hogy $0 \leq k \leq n$ esetén

$$\langle f - p_n, T_k \rangle = \langle f, T_k \rangle - \langle p_n, T_k \rangle = \langle f, T_k \rangle - \alpha_k \langle T_k, T_k \rangle = \langle f, T_k \rangle - \langle f, T_k \rangle = 0.$$

3.1.14. Állítás. $f - p_n$ legalább $n + 1$ -szer vált előjelet $[-1, 1]$ belsejében.

Bizonyítás. Mivel $\langle f - p_n, T_0 \rangle = \langle f - p_n, 1 \rangle = 0$, legalább egy $x_i \in (-1, 1)$ pont ahol a különbség előjelet vált. Most tegyük, fel hogy $f - p_n$ előjelet vált $1 \leq k < n + 1$ pontban. Legyenek ezek $-1 < x_1 < x_2 < \dots < x_k < 1$. Definiáljuk a következő polinomot

$$\pi_k(x) = (x - x_1)(x - x_2) \dots (x - x_k).$$

Világos, hogy π_k éppen x_i pontokban vált előjelet. Ebből következik, hogy $\langle f - p_n, \pi_k \rangle \neq 0$. Viszont mivel minden polinom előll Csebisev-polinomok lineáris kombinációjaként, $\pi_k = \sum \beta_i T_i$, amiből következik, hogy $\langle f - p_n, \pi_k \rangle = 0$. Ez ellentmondás, tehát $n + 1 \leq k$. □

3.1.15. Állítás. $T_n(x)$ -nek n darab különböző gyöke van $[-1, 1]$ intervallumon, mégpedig $y_i = \cos\left(\frac{(2i-1)\pi}{2n}\right)$ helyeken, $i = 1, \dots, n$.

Bizonyítás. Mivel $T_n(x)$ n -ed fokú polinom, tudjuk hogy n darab komplex gyöke van, ráadásul ezek pontosan az y_i -vel jelölt számok

$$T_n(y_i) = \cos(n \arccos(y_i)) = \cos\left(n \arccos\left(\cos\left(\frac{(2i-1)\pi}{2n}\right)\right)\right) = \cos\left(\frac{(2i-1)\pi}{2}\right) = 0.$$

□

3.1.16. Következmény. A $\|(x - x_0) \dots (x - x_n)\|$ normát $x_j = y_j$ választással minimalizálhatjuk, ahol y_j számok éppen T_{n+1} gyökei.

3.1.17. Tétel. Legyen f $(n + 1)$ -szer differenciálható és $f^{(n+1)}$ legyen folytonos $[-1, 1]$ -en. Ilyenkor létezik egy $\xi \in (-1, 1)$ konstans, hogy

$$\|f - p_n\| = \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \|(x - y_0) \dots (x - y_n)\| = \frac{\sqrt{\pi}}{2^{n+1/2}} \frac{|f^{(n+1)}(\xi)|}{(n+1)!}.$$

Bizonyítás. Mivel $f - p_n$ legalább $n + 1$ -szer előjelet vált, létezik x_i, \dots, x_n , amire $p_n(x_i) = f(x_i)$. Azaz p_n f -nek interpolációs polinomja. Vizsgáljuk a következő $G(x)$ függvényt

$$G(x) = f(x) - p_n(x) - \frac{\prod_{i=0}^n (x - x_i)}{\prod_{i=0}^n (a - x_i)} (f(a) - p_n(a)),$$

ahol $a \neq x_i$. Világos, hogy $G(x)$ -nek $n + 2$ gyöke van: a, x_0, \dots, x_n . A Rolle-tétel alapján egy differenciálható függvény két nullhely között létezik egy pont, ahol a derivált is nulla. $n + 1$ -szer alkalmazva a Rolle-tételt, G' -nek legalább $n + 1$ nullhelye van. $\dots, G^{(n+1)}$ -nek pedig legalább egy gyöke, azaz létezik $\xi_a \in [-1, 1]$, amire

$$0 = f^{(n+1)}(\xi_a) - \frac{(n+1)!(f(a) - p_n(a))}{(a - x_0) \dots (a - x_n)}$$

Átrendezve, $x = a$ helyettesítéssel élve, és normát számítva

$$\|f - p_n\| = \frac{f^{(n+1)}(\xi_a)}{(n+1)!} \|(x - x_0) \dots (x - x_n)\|.$$

Végül f -et approximálva p^* polinommal, ahol p^* alappontjai T_{n+1} gyökei, azaz y_i pontok, majd kihasználva, hogy p_n minimalizálja a normát

$$\|f - p_n\| \leq \|f - p^*\| = \frac{f^{(n+1)}(\xi)}{(n+1)!} \|(x - y_0) \dots (x - y_n)\|.$$

Végül kiszámítva az integrált

$$\|(x - y_0) \dots (x - y_n)\| = \left(\int_{-1}^1 \frac{(T_{n+1})^2}{2^n \sqrt{1 - x^2}} dx \right)^{1/2} = \frac{\sqrt{\pi}}{2^n \sqrt{2}}.$$

□

Most rátérünk $\hat{g}(\mathcal{L}_n)x$ kiszámítására Csebisev-polinomok segítségével. Mivel $\sigma(\mathcal{L}_n) \subset [0, 2]$, toljuk el a Laplace-mátrix spektrumát $[-1, 1]$ tartományba: $\overline{\mathcal{L}_n} = \mathcal{L}_n - I$. Ebből adódik a $\hat{g}(\mathcal{L}_n) = \sum_{i=0}^k \theta'_i \mathcal{L}_n^i = \sum_{i=0}^k \theta_i T_i(\overline{\mathcal{L}_n})$ formula, felhasználva T_i polinomok bázis tulajdonságát. k megfelelő választásával. A feladatunk a $\Theta \in \mathbb{R}^{k+1}$ paraméter tanítása lesz. $y_k = T_k(\overline{\mathcal{L}_n})x$ jelöléssel, rekurziót használva a következőt kapjuk

$$y_0 = x, \quad y_1 = \overline{\mathcal{L}_n}x, \quad y_k = 2\overline{\mathcal{L}_n}y_{k-1} - y_{k-2}.$$

Összerakva a számítás a következőre egyszerűsödik, műveleti igénye pedig $\mathcal{O}((k+1)|E|)$

$$x * g_\Theta = \hat{g}(\mathcal{L}_n)x = \sum_{i=0}^k \theta_i T_i(\overline{\mathcal{L}_n})x = \langle \Theta, y \rangle.$$

p darab input és q darab output channel esetén $X \in \mathbb{R}^{n \times p}$ és $X' \in \mathbb{R}^{n \times q}$

$$X' = \sum_{l=1}^k Z^{(l)} \Theta^{(l)},$$

ahol

$$Z^{(1)} = \overline{\mathcal{L}_n}, \quad Z^{(2)} = \overline{\mathcal{L}_n} X, \quad Z^{(l)} = 2\overline{\mathcal{L}_n} Z^{(l-1)} - Z^{(l-2)}.$$

Tehát a tanulandó paraméterek számára igaz hogy $\Theta^{(l)} \in \mathbb{R}^{p \times q}$, azaz egy k -ad fokú Cse-bisev filter $p \cdot q \cdot k$ paramétert tartalmaz.

Vizsgáljuk azt a speciális esetet, amikor $k = 1$. A gyakorlatban minnél kevesebb paraméterre törekszünk, ezért élünk az $\theta_0 = -\theta_1 = \theta$ választással.

$$\begin{aligned} x * g_{\Theta} = \hat{g}(\mathcal{L}_n)x &= \sum_{i=0}^1 \theta_i T_i(\overline{\mathcal{L}_n})x = \theta_0 x + \theta_1 (\mathcal{L}_n - I)x = \theta_0 x - \theta_1 (D^{-\frac{1}{2}} W D^{-\frac{1}{2}})x = \\ &= \theta (I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}})x \end{aligned}$$

Vegyük észre, hogy az itt szereplő $I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ mátrix spektruma $[0, 2]$ közé esik. Valóban, $I + D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = -\mathcal{L}_n + 2I$, továbbál $\sigma(\mathcal{L}_n) \subset [0, 2]$, akkor $\sigma(-\mathcal{L}_n + 2I) \subset [-2 + 2, 0 + 2] = [0, 2]$. A Laplace mátrix első hatványával való szorzása egy csúcshoz csak a közvetlen szomszédjaira van hatással, ezt orvosoljuk több réteg egymás utáni alkalmazásával, ami garantálja, hogy l réteg esetén egy csúcsot pontosan a tőle maximum l távolságra lévő csúcsok fognak befolyásolni. Viszont ez a fenti mátrix többszöri egymás utáni alkalmazásához vezet, amelynek spektruma $[0, 2]$ közé esik, a tapasztalat azt mutatja, hogy ez esetben a számolás numerikusan instabillá válik. Ennek orvosolása érdekében az identitás tagot olvasszuk bele a mátrixba hurokélek behúzásával, $\widehat{W} = W + I$, $\widehat{D} = \sum_{j \in N(i)} W_{ij} + 1$. Elvégezve a normalizációt, kapjuk, a $\widehat{D}^{-\frac{1}{2}} \widehat{W} \widehat{D}^{-\frac{1}{2}}$. Általánosabban, ha az input $X \in \mathbb{R}^{n \times c}$, ahol n a gráf csúcsainak száma, c pedig az egy csúcsához tartozó attribútumok száma, akkor a konvolúció a következőképpen írható fel

$$Z = \widehat{D}^{-\frac{1}{2}} \widehat{W} \widehat{D}^{-\frac{1}{2}} X \Theta,$$

ahol $\Theta \in \mathbb{R}^{c \times m}$ mátrix tartalmazza a tanulandó filter paramétereit m channel esetén, azaz $Z \in \mathbb{R}^{n \times m}$. Több réteg esetén σ -val jelölve az aktivációs függvényt és ha $H^0 = X$,

$$H^{(k+1)} = \sigma \left(\widehat{D}^{-\frac{1}{2}} \widehat{W} \widehat{D}^{-\frac{1}{2}} H^l \Theta^{(l)} \right).$$

Egy réteg kiszámításának műveleti igénye $\mathcal{O}(|E| \text{ cm})$. Egy v_i csúcs értékét egy tehát éppen a szomszédaiból kinyert információ és a háló tanítandó paraméterei határozzák meg.

$$x'_i = \Theta \sum_{j \in N(i) \cup \{i\}} \frac{e_{j,i}}{\sqrt{\widehat{d}_i \widehat{d}_j}} x_j, \quad \widehat{d}_j = 1 + \sum_{i \in N(j)} e_{j,i}.$$

ahol, $e_{i,j}$ az adott él súlya. Ezt az architektúrát szokás GCN-nek (Graph Convolutional Network) nevezni.

3.2. Cayley-polinomok

Ebben a fejezetben bemutatjuk a gráf filterek másik népszerű családját, a Cayley polinomokat. A Cayley polinomok komplex együtthatós, valós értékű polinomok, amelyek a Csebisev-polinomokhoz hasonló tulajdonsággal rendelkeznek. Jelöljük a komplex egységkört $e^{i\mathbb{R}} = \{e^{i\theta} : \theta \in \mathbb{R}\}$ -rel. A Cayley tranformáció a $C(z) = \frac{z-i}{z+i}$ képlettel definiált függvény. A komplex számsíkon értelmezett törtlineáris függvények a köröket és egyeneseket, körökbe és egyenesekbe képzik. Vegyük a valós egyenest, és azon a $\{\infty, 1, -1\}$ pontokat. Ezen pontok képe a Cayley tranzformációra rendre $\{1, -i, i\}$, tehát a Cayley tranzformáció egy folytonos bijekció \mathbb{R} és $e^{i\mathbb{R}}$ között.

3.2.1. Következmény. *Mivel a Laplace mátrix sajátértékei nemnegatív valósak, a Cayley tranformáció által nyert $C(\mathcal{L}_n) = (\mathcal{L}_n - iI)(\mathcal{L}_n + iI)^{-1}$ komplex mátrix sajátértékei rajta vannak a komplex egységkör alsó felén.*

3.2.2. Definíció. *Legyen az r -ed fokú Cayley polinom*

$$g_{c,h}(\lambda) = c_0 + 2\text{Re} \left\{ \sum_{j=1}^r c_j \left(\frac{h\lambda - i}{h\lambda + i} \right)^j \right\},$$

ahol az együtthatók $c = (c_0, c_1, \dots, c_r) \in \mathbb{R} \oplus \mathbb{C}^r$ és $h \in \mathbb{R}$ nagyító paraméter.

A nagyító paraméterrel való szorzás kinyújtja a valós egyenesen fekvő sajátértékeket, ezáltal ránagyítja a spektrum számunkra fontos részére. Ez különösen hasznos lehet közösségetektáló feladatoknál, ahol a sajátértékek sok információt hordoznak a klaszterekről. A tanulás folyamán h és a Cayley polinomok együtthatói tanulandó paraméterek.

Kihasználva, hogy $2\text{Re}\{z\} = z + \bar{z}$, és hogy az egységkörön $z^{-1} = \bar{z}$, kapjuk

$$g_{c,h}(\mathcal{L}_n) = c_0 I + \sum_{j=1}^r c_j C^j(h\mathcal{L}_n) + \bar{c}_j C^{-j}(h\mathcal{L}_n).$$

Az SVD felbontás alapján $C^j(h\mathcal{L}_n) = C^j(h\Phi\Lambda\Phi^\top) = \Phi\text{diag}([C(h\lambda_1)]^j, \dots, [C(h\lambda_n)]^j)\Phi^\top$. A továbbiakban megmutatjuk, hogy tetszőleges gráf filter meghatározható Cayley polinomok segítségével, azaz bármilyen $\lambda_i \in [-1, 1]$ és $y_i \in \mathbb{R}$ értékekre létezik $g_{c,1}$ Cayley polinom, amelyre $g(\lambda_i) = y_i$ ahol $(i = 1, \dots, n)$.

3.2.3. Állítás. *A Cayley polinomok felírhatóak trigonometrikus polinom alakban.*

Bizonyítás. Mivel a Cayley transzformáció folytonos bijekció a valós egyenes és a komplex egységkör között, létezik paraméterezés, amire $C(\lambda) = e^{i\omega}$ ($\omega \in [0, 2\pi]$). Mivel $e^{ix} = \cos(x) + i \sin(x)$, ezek alapján a Cayley polinomok a következő alakra hozhatóak

$$\begin{aligned} g_{c,1}(\lambda) &= c_0 + \sum_{j=1}^r c_j C^j(\lambda) + \bar{c}_j C^{-j}(\lambda) = c_0 + \sum_{j=1}^r c_j e^{ij\omega} + \bar{c}_j e^{-ij\omega} \\ &= c_0 + \sum_{j=1}^r c_j [\cos(j\omega) + i \sin(j\omega)] + \bar{c}_j [\cos(j\omega) - i \sin(j\omega)] \\ &= c_0 + 2 \sum_{j=1}^r \text{Re}\{c_j\} \cos(j\omega) - \text{Im}\{c_j\} \sin(j\omega). \end{aligned}$$

□

3.2.4. Állítás. *A $p(t) = a_0 + \sum_{j=1}^r [a_j \cos(tj) + b_j \sin(tj)]$ trigonometrikus polinom együtthatói megválaszthatóak úgy, hogy $p(x_i) = y_i$ teljesüljön, ahol $x_i \in [0, 2\pi]$, $y_i \in \mathbb{R}$. és $i = \{0, \dots, 2n\}$.*

Bizonyítás. Írjuk át a polinomot exponenciális alakba: $p(t) = \sum_{j=-n}^n c_j e^{ijx}$ és vegyük észre, hogy $c_0 = a_0$, $c_j = \frac{1}{2}(a_j - b_j)$ és $c_{-j} = \frac{1}{2}(a_j + b_j)$ $j \in \{1, \dots, n\}$. A Lagrange interpolációs módszert használjuk. Legyen $p(t) = \sum_{j=0}^{2n} y_k \ell_k(x)$, ahol a Lagrange-polinomok

$$\ell_k(x) = e^{-inx+inx_k} \prod_{m=0, m \neq k}^{2n} \frac{e^{ix} - e^{ix_k}}{e^{ix_k} - e^{ix_m}}.$$

Könnyen ellenőrizhető, hogy $\ell_k(x_j) = \delta_{kj}$. Az $e^{-inx+inx_k}$ faktorra azért van szükség, hogy $\ell_k(x)$ -ben e^{ix} kitevője $-n$ és n közé essen. A valós esethez hasonlóan belátható, hogy $2n + 1$ alappont esetén egyértelmű az interpolációs polinom, mivel $2n + 1$ együtthatót

választhatunk meg. Páros sok alappont esetén felvesszünk egy tetszőleges pontot és úgy végezzük el az interpolációt. \square

Megjegyezzük, hogy ha az $2n + 1$ darab alappont ekvidisztáns távolságban van, azaz $x_k = \frac{2\pi k}{2n+1}$, és a feladat egy $u(x)$ komplex 2π szerint periodikus függvény közelítése akkor az interpolációs polinom együtthatóit egy egyszerűbb összefüggés határozza meg. Legyen $p(t) = \frac{1}{\sqrt{2\pi}} \sum_{j=-n}^n c_j e^{ijx}$ az interpolációs polinom. Célunk a c_j együtthatók megválasztása, hogy $p(x_j) = u(x_j) = u_j$ teljesüljön. Legyen $u, v \in [0, 2\pi] \rightarrow \mathbb{C}$ komplex periodikus függvények diszkrét skalárszorzata $\langle u, v \rangle = \sum_{j=0}^n u_j \bar{v}_j h$.

3.2.5. Állítás. *A diszkrét skalárszorzatra nézve az $\left\{ \frac{1}{\sqrt{2\pi}} e^{ijx} \right\}_{j=-n}^n$ függvények ortonormáltak.*

Bizonyítás. Legyenek $a, b \in -n, \dots, n$. Felhasználva, hogy $h = \frac{2\pi i}{2n+1}$, valamint a mértani sorozat összegképletét, kapjuk

$$\langle e^{iax}, e^{ibx} \rangle = \sum_{j=0}^n e^{i(a-b)jh} h = \frac{1 - e^{i(a-b)2\pi}}{1 - e^{i(a-b)h}} h = \begin{cases} 0 & a = b, \\ 2\pi & a \neq b. \end{cases}$$

\square

Az $\frac{1}{\sqrt{2\pi}} \sum_{j=-n}^n c_j e^{ijx_k} = u(x_k)$ egyenlet mindkét oldalát diszkrét skalárisan megszorozva e^{ijx} függvénnyel, azonnal adódik az alábbi állítás.

3.2.6. Állítás. *Az interpolációs polinom együtthatóira teljesül, hogy $c_j = \frac{1}{\sqrt{2\pi}} \langle e^{ijx}, u \rangle$.*

A következőkben rátérünk a Cayley polinomok numerikus tulajdonságaira. Legyen adott az $x : V \rightarrow \mathbb{R}$ gráf csúcsain értelmezett függvény. Ki szeretnénk számolni $g_{c,h}(\mathcal{L}_n)x$ -et. Ehhez meg kell határozni $C^j(h\mathcal{L}_n)x$ értékeket. A Cayley transzformáció definícióját használva az alábbi rekurzív egyenletrendszer kapjuk

$$y_0 = x, \quad (h\mathcal{L}_n + iI)y_j = (h\mathcal{L}_n - iI)y_{j-1}, \quad j = 1, \dots, r.$$

Így $y_j = C^j(h\mathcal{L}_n)x$ adódik. Az egyenletrendszer megoldható a megfelelő mátrixok invertálásával $\mathcal{O}(n^3)$ lépésben vagy a Jacobi-iteráció segítségével approximálhatjuk a megoldást. Vezessük be az alábbi jelölést: $\text{off}(A) = A - \text{diag}(A)$.

3.2.7. Állítás. (Jacobi-iteráció) Ha a $J = -\text{diag}(A)^{-1} \text{off}(A)$ átmeneti mátrixra $\|J\| < 1$ valamilyen operátornormára, akkor az $Ax = b$ egyenletnek egyértelműen létezik megoldása és a megoldást megkaphatjuk tetszőleges x_0 kiindulási vektorból az alábbi rekurzióval

$$x^{(n+1)} = -\text{diag}(A)^{-1} \text{off}(A)x^{(n)} + \text{diag}(A)^{-1}b.$$

Legyen tehát $J = -\text{diag}(h\mathcal{L}_n + iI)^{-1} \text{off}(h\mathcal{L}_n + iI)$ az átmenet mátrix, ekkor K iterációs lépés esetén:

$$\begin{aligned}\widehat{y}_j^{(0)} &= b_j, \\ \widehat{y}_j^{(k+1)} &= J\widehat{y}_j^{(k)} + b_j, \\ b_j &= \text{diag}(h\mathcal{L}_n + iI)^{-1}(h\mathcal{L}_n - i, I)\widehat{y}_{j-1}, \\ \widehat{y}_j &= \widehat{y}_j^{(K)}.\end{aligned}$$

Megmutatható, hogy h megfelelő megválasztásával elérhető $\|J\| < 1$ valamilyen operátornormára. Tehát $g_{c,h}(\mathcal{L}_n)x \approx c_0\widehat{y}_0 + 2\text{Re} \sum_{j=1}^r c_j\widehat{y}_j = \widehat{g_{c,h}(\mathcal{L}_n)}x$ közelítést kapjuk.

Mivel a Cayley polinomok komplex együtthatós polinomok, ezért a backpropagation algoritmus végrehajtásakor meg kell állapodnunk, hogy mit értünk gradiens vektor alatt. Legyen L egy differenciálható veszteségfüggvény, vizsgálni szeretnénk az $F(c, h) = L(g_{c,h}(\mathcal{L}_n)x)$ összetett függvény c és h szerinti parciális deriváltjait. A $\frac{\partial F(a)}{\partial v}$ a -beli, v irányú iránymenti deriváltra teljesül

$$F(a + tv) - F(a) = \left\langle \frac{\partial F(a)}{\partial v}, v \right\rangle + o(t).$$

Az $f(z) = \bar{z}$ komplex függvény nem differenciálható, ezért a szokásos $\langle x, y \rangle = x\bar{y}$ skálárszorzat helyett mást kell választanunk. Egy \mathcal{H} komplex Hilbert tér realizációján azt a $\mathcal{H}_{\mathbb{R}}$ teret értjük, ahol a szorzás műveletet megszorítjuk a valós számokra. Ezáltal ha $\dim(\mathcal{H}) = r$, akkor $\mathcal{H}_{\mathbb{R}} \cong \mathbb{R}^{2r}$. Értelmezzük két $x, y \in \mathcal{H}_{\mathbb{R}}$ vektorok skalárszorzatát $\text{Re} \langle x, y \rangle = \text{Re}\{x\}\text{Re}\{y\} + \text{Im}\{x\}\text{Im}\{y\}$. Tekintsük r -ed fokú Cayley polinom esetén az együtthatók által meghatározott \mathbb{R}^{2r+2} realizált vektorteret. A h szerinti derivált

$$\begin{aligned}\frac{\partial g_{c,h}(y)}{\partial h} &= \frac{\partial C(hy)}{\partial(hy)} y \sum_{j=1}^r j (c_j C^{j-1}(hy) - \bar{c}_j C^{-j-1}(hy)) \\ \frac{\partial C(z)}{\partial z} &= \frac{z + i - (z - i)}{(z + i)^2} = \frac{1 - C(z)}{z + i} \\ \frac{\partial F(c, h)}{\partial h} &= L'(g_{c,h}(\mathcal{L}_n)x)C'(h\mathcal{L}_n)y \sum_{j=1}^r j (c_j C^{j-1}(h\mathcal{L}_n) - \bar{c}_j C^{-j-1}(h\mathcal{L}_n)).\end{aligned}$$

A c szerinti derivált pedig

$$\frac{g_{c+tv,h}(\mathcal{L}_n)x - g_{c,h}(\mathcal{L}_n)x}{t} = g_{t,h}(\mathcal{L}_n)x = \operatorname{Re}\{t_0x + \sum_{j=1}^r 2t_j C^j(h\mathcal{L}_n)x\}.$$

Tehát a realizált térben vett skalárszorzat alapján

$$\frac{\partial g_{c,h}(\mathcal{L}_n)x}{\partial c_j} = \begin{cases} x & j = 0, \\ 2C^j(h\mathcal{L}_n)x & j \neq 0. \end{cases}$$

3.3. Racionális törtfüggvények, ARMA

3.3.1. Definíció. K -ad fokú $ARMA_K$ (Autoregressive–moving-average) filternek nevezzük a

$$h(\lambda) = \frac{\sum_{i=0}^{K-1} a_i \lambda^i}{1 + \sum_{i=1}^K b_i \lambda^i}$$

alakú racionális törtfüggvényeket, ahol $a_i, b_i \in \mathbb{R}$.

Ha $x : V \rightarrow \mathbb{R}$, akkor

$$h(\mathcal{L}_n)x = \Phi h(\Lambda) \Phi^\top x = \Phi \left(I + \sum_{i=1}^K b_i \Lambda^i \right)^{-1} \Phi \Phi^\top \left(\sum_{i=0}^{K-1} a_i \Lambda^i \right) \Phi x = \left(I + \sum_{i=1}^K b_i \mathcal{L}_n^i \right)^{-1} \left(\sum_{i=0}^{K-1} a_i \mathcal{L}_n^i \right) x^{(0)}.$$

Vizsgáljuk meg a következő rekurziót, $x = x^{(0)}$ jelöléssel

$$x^{(t+1)} = a(I - \mathcal{L}_n)x^{(t)} + bx^{(0)}.$$

3.3.2. Lemma. Ha $A \in \mathbb{R}^{n \times n}$ és minden sajátére $|\lambda| < 1$, akkor $\lim_{k \rightarrow \infty} A^k = 0$.

3.3.3. Állítás. Ha $|a| < 1$, tetszőleges $x : V \rightarrow \mathbb{R}$ kezdeti értékekre a fenti rekurzió $ARMA_1$ filterhez tartozó x' -höz konvergál.

Bizonyítás. Mivel \mathcal{L}_n sajátértékei $[0, 2]$ intervallumba esnek és $|a| < 1$, az $a(I - \mathcal{L}_n)$ mátrix sajátértékei a $(-1, 1)$ intervallumba esnek, tehát a lemma használható az első tagra.

$$x' = \lim_{t \rightarrow \infty} \left[a(I - \mathcal{L}_n)^t x + b \sum_{i=0}^t [a(I - \mathcal{L}_n)]^i x \right] = 0 + b \sum_{i=0}^{\infty} [a(I - \mathcal{L}_n)]^i x$$

A második tag egy geometriai sor, mivel $a(I - \mathcal{L}_n)$ spektrálsugara kisebb, mint 1, konvergens a sor, a határérték pedig $b(I - a(I - \mathcal{L}_n))^{-1}$. Ennek a mátrixnak a sajátértékei $\frac{b}{1-a\lambda_i}$ alakúak, tehát a rekurzióhoz tartozó ARMA₁ filter $h(\lambda) = \frac{b}{1-a\lambda}$. \square

K darab ARMA₁ filter összege éppen egy K -ad fokú racionális törtfüggvény, azaz ARMA _{K} alakú, visszafele, ha komplex együtthatókat is megengednénk, akkor a parciális törtekre bontás tétele értelmében minden komplex ARMA _{K} filter felírható ARMA₁ filterek összegéként. Ha az ARMA₁ filterek rendre $h_i(\lambda) = \frac{b_k}{1-a_k\lambda}$ alakúak, akkor a hozzájuk tartozó ARMA _{K} filter $h(\lambda) = \sum h_i(\lambda) = \sum_{k=1}^K \frac{b_k}{1-a_k\lambda}$. Implementáció szempontjából kényelmesebb az iteráció számát egy fix T számnak rögzíteni. A tanítási folyamat alatt aktivációs függvényeket használunk az ARMA₁ filterek alkalmazása után is. Ha tehát $X \in \mathbb{R}^{n \times IN}$ és $W, V \in \mathbb{R}^{IN \times OUT}$ mátrixok tartalmazzák a tanulandó paramétereket, akkor az ARMA _{K} filter az imént vizsgált rekurzió mintájára az alábbi alakban írható fel.

$$X_k^{(t+1)} = \sigma((I - \mathcal{L}_n)X_k^{(t)}W + XV),$$

$$X' = \frac{1}{K} \sum_{k=1}^K X_k^{(T)}.$$

3.3.4. Definíció. $A \in \mathbb{R}^{n \times n}$ mátrix spektrálsugara $\rho(A) = \{|\lambda| : \lambda \text{ sajátérték}\}$.

3.3.5. Definíció. $A \in \mathbb{R}^{n \times n}$ mátrix Euklideszi-normája vagy spektrál-normája

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\rho(A^*A)}.$$

továbbá ha $A = A^*$ akkor $\|A\|_2 = \rho(A)$.

A fenti egyenlőség teljesül, mivel $\|A\|_2^2 = \max_{\|x\|_2=1} \|Ax\|_2^2 = \max_{\|x\|_2=1} \langle Ax, Ax \rangle = \max_{\|x\|_2=1} \langle A^*Ax, x \rangle = \lambda_{\max}$, ahol az utolsó egyenlőség a Rayleigh-tétel miatt teljesül. Ha A önadjungált, akkor $AA^* = A^2$, azaz ha A legnagyobb sajátértéke μ_{\max} , akkor A^2 -é μ_{\max}^2 , azaz $\|A\|_2 = \sqrt{\mu_{\max}^2} = \mu_{\max} = \rho(A)$.

3.3.6. Állítás. Ha $\|W\|_2 < 1$, és σ 1-Lipschitz, akkor $X^{(t)}$ konvergens.

Bizonyítás. Legyenek $X_1, X_2 \in \mathbb{R}^{n \times IN}$ két különböző kezdeti jel. Megmutatjuk, hogy $g(X) = \sigma((I - \mathcal{L}_n)XW + XV)$ kontrakció. Ekkor a Banach-fixpont tétel értelmében

$X^{(t)}$ konvergencia, tetszőleges kezdeti értékre.

$$\begin{aligned}
& \left\| g(X_1^{(t)}) - g(X_2^{(t)}) \right\|_2 = \left\| X_1^{(t+1)} - X_2^{(t+1)} \right\|_2 \\
& = \left\| \sigma((I - \mathcal{L}_n)X_1^{(t)}W + XV) - \sigma((I - \mathcal{L}_n)X_2^{(t)}W + XV) \right\|_2 \\
& \leq \left\| (I - \mathcal{L}_n)X_1^{(t)}W + XV - (I - \mathcal{L}_n)X_2^{(t)}W + XV \right\|_2 = \left\| (I - \mathcal{L}_n)X_1^{(t)}W - (I - \mathcal{L}_n)X_2^{(t)}W \right\|_2 \\
& \leq \|I - \mathcal{L}_n\|_2 \|W\|_2 \left\| X_1^{(t)} - X_2^{(t)} \right\|
\end{aligned}$$

Mivel $\|W\|_2 < 1$ és $I - \mathcal{L}_n$ legnagyobb sajátértékének abszolút értéke legfeljebb 1, ezért g kontrakció. \square

Többek között 1-Lipschitz a $\text{ReLU}(x) = \max(x, 0)$ vagy a sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$ függvény is.

3.4. B-Spline függvények

3.4.1. Definíció. *Kardinális B-spline függvényeknek nevezzük az alábbi rekurzióval definiált függvényeket*

$$B_0(t) = \chi_{[0,1]}, \quad B_n(t) = (B_{n-1} * B_0)(t) = \int_{\mathbb{R}} B_{n-1}(t-x)B_0(x) dx.$$

A B-spline-ok néhány tulajdonságának bizonyítása teljes indukcióval történik, az indukció elindításához számoljuk ki B_1 -et.

$$B_1(t) = \int_{\mathbb{R}} B_0(t-x)B_0(x) dx = \int_0^1 B_0(t-x) dx = \int_{\max(0,t-1)}^{\min(1,t)} dx = \begin{cases} t & \text{ha } t \in [0, 1], \\ 2-t & \text{ha } t \in [1, 2]. \end{cases}$$

3.4.2. Állítás. B_n tartója $[0, n+1]$ intervallum, továbbá ezen az intervallumon $B_n \geq 0$.

Bizonyítás. Indukcióval bizonyítunk. B_0 tartója $[0, 1]$ intervallum és nemnegatív. Világos, hogy nemnegatív függvények konvolúciója nemnegatív. Tegyük fel hogy B_{n-1} tartója $[0, n]$, ekkor $B_n(t) = \int_{\mathbb{R}} B_{n-1}(t-x)B_0(x) dx = \int_{t-1}^t B_{n-1}(x) dx$. Ha $t \notin [0, n+1]$ akkor az integrálási tartományon az integrandus 0. \square

3.4.3. Állítás. B_n $n-1$ -szer folytonosan differenciálható függvény $(0, n+1)$ -en, továbbá teljesül a $B_n'(t) = B_{n-1}(t) - B_{n-1}(t-1)$ azonosság.

Bizonyítás. $B'_n(t) = \frac{d}{dt} \int_0^1 B_{n-1}(t-x) dx = \frac{d}{dt} \int_{t-1}^t B_{n-1}(x) dx = B_{n-1}(t) - B_{n-1}(t-1)$. \square

3.4.4. Állítás. *A B-spline-ok szimmetrikus függvények, azaz $B_n(t) = B_n(n+1-t)$.*

Bizonyítás. Teljes indukcióval bizonyítunk. $B_n(t) = \int_{t-1}^t B_{n-1}(x) dx$ és $B_n(n+1-t) = \int_{n-t}^{n-(t-1)} B_{n-1}(x) dx$. A feltevésünk szerint B_{n-1} -re teljesül a szimmetria tulajdonság, így a függvény integráljai megegyezik $[t-1, t]$ és $[n-t, n-(t-1)]$ intervallumon, azaz B_n -re is teljesül a szimmetria tulajdonság. \square

3.4.5. Tétel (Cox-Deboor). $B_n(t) = \frac{t}{n} B_{n-1}(t) + \frac{n+1-t}{n} B_{n-1}(t-1)$.

Bizonyítás. Teljes indukcióval bizonyítunk. $n=1$ -re $B_1(t) = tB_0(t) + (2-t)B_0(t) = t\chi_{[0,1]} + (2-t)\chi_{[1,2]}$ teljesül a formula. Világos, hogy $B_n(0) = 0$ minden n -re és $B_n(n+1) = 0$ a szimmetria tulajdonság miatt. Legyen $t \in (0, n+1)$. Itt $B_n, n \geq 2$ esetén differenciálható, a rekurziót a deriváltakra fogjuk belátni. A deriváltakra vonatkozó állítás alapján $B'_n(t) = B_{n-1}(t) - B_{n-1}(t-1)$. Most deriváljuk a rekurzió jobb oldalát

$$\begin{aligned} & \left(\frac{t}{n} B_{n-1}(t) \right)' + \left(\frac{n+1-t}{n} B_{n-1}(t-1) \right)' = \frac{1}{n} (B_{n-1}(t) - B_{n-1}(t-1)) \\ & + \left[\frac{t}{n} (B_{n-2}(t) - B_{n-2}(t-1)) + \frac{n+1-t}{n} (B_{n-2}(t-1) - B_{n-2}(t-2)) \right] \end{aligned}$$

Használva az indukciós feltevést a kapcsos zárójelben szereplő tagra

$$\begin{aligned} & \frac{n-1}{n} \left[\left(\frac{t}{n-1} B_{n-2}(t) + \frac{n-t}{n-1} B_{n-2}(t-1) \right) - \left(\frac{t-1}{n-1} B_{n-2}(t-1) + \frac{n-(t-1)}{n-1} B_{n-2}(t-2) \right) \right] \\ & = \frac{n-1}{n} (B_{n-1}(t) - B_{n-1}(t-1)) \end{aligned}$$

Végül visszahelyettesítve ugyanazt kifejezést kapjuk, amit imént számoltunk ki $B'_n(t)$ -re. Tehát azt kaptuk, hogy a rekurzió teljesül a deriváltakra. Mivel a rekurzió mindkét oldalán folytonos függvények vannak és a deriváltak megegyeznek $(0, n+1)$ -en és $t=0$ helyen a bal és jobb oldal is egyenlő, következik hogy a rekurzió B_n -re is igaz. \square

3.4.6. Következmény. *A $[j, j+1] \subset [0, n+1]$, $j \in \{0, \dots, n\}$ intervallumokon B_n éppen egy n -ed fokú polinommal egyenlő.*

Az imént definiált B-spline függvények tartója $[0, n+1]$, további paraméterek bevezetésével tetszőleges intervallumra ki tudjuk terjeszteni a definíciót. Legyen $B_n^{i,h}(t) = B_n\left(\frac{t}{h} - i\right)$.

Ekkor $B_n^{i,h}$ tartója $[i, i + n + 1]h$ intervallum. Továbbá ha tekintjük a $t_i = hi$ sorozatot, ahol $i \in \mathbb{Z}$, akkor a Cox-Deboor tétel a következő alakban írható fel.

$$B_n^{i,h}(t) = \frac{t - t_i}{t_{i+n} - t_i} B_{n-1}^{i,h}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} B_{n-1}^{i+1,h}(t).$$

A $(t_i)_{i \in \mathbb{Z}}$ sorozatot a B-splineekhoz tartozó csomóvektoroknak nevezzük. Az n -ed fokú B-splineek definiálhatóak egy véges $[a, b]$ intervallumon a Cox-Deboor tétel és a csomóvektorok megfelelő megválasztásával. Legyen ugyanis $(t_i)_{i=1}^{n+p+1}$ egyenletes felosztása az $[a - nh, b + nh]$ intervallumnak, azaz $t_i = (a - (n + 1)h + ih)$, $i \in \{1, \dots, n + p + 1\}$, $b = t_{p+1} = a - (n + 1)h + (p + 1)h$, amiből $h = \frac{b-a}{p-n}$. A nulladfokú B-spline-ok a $B_0^{i,h} = \chi_{[t_{n+i}, t_{n+i+1}]}$ alakú függvények. Ezekből rekurzív módon a Cox-Deboor-tétel segítségével definiálhatóak a n -ed fokú B-spline-ok $\{B_n^{1,h}, \dots, B_n^{p,h}\}$. Ennél fogva $B_n^{i,h}$ tartója $[t_i, t_{n+1+i}] = [a + h(i - n - 1), a + ih]$ intervallum lesz. A csomóvektorokról nem kötelező feltenni, hogy ekvidisztánsan helyezkedjenek el, tetszőleges $(t_i)_{i \in \mathbb{Z}}$ sorozatra definiálhatóak a B-splineok.

3.4.7. Állítás (Marsden). *Legyen $m_{i,n}(y) = \prod_{j=i+1}^{i+n} (y - t_j)$. Ekkor teljesül a $(y - x)^n = \sum_{i=m-n}^n m_{i,n}(y) B_n^{i,h}(x)$ azonosság, ha $t \in [t_m, t_{m+1}]$, ahol $n + 1 \leq m \leq p$.*

Bizonyítás. Először belátjuk, hogy $m_{i,n}(y)$ -ra teljesül az alábbi összefüggés

$$(y - x)m_{i,n-1}(y) = \frac{x - t_i}{t_{n+i} - t_i} m_{i,n}(y) + \frac{t_{n+i} - x}{t_{n+i} - t_i} m_{i-1,n}(y).$$

Világos, hogy $y - x = \frac{x - t_i}{t_{n+i} - t_i} (y - t_{n+i}) + \frac{t_{n+i} - x}{t_{n+i} - t_i} (y - t_i)$ igaz. Az egyenlet mindkét oldalát megszorozva $m_{i,n-1}(y)$ -nal éppen a bizonyítandót kapjuk. Ennek segítségével már teljes indukcióval bebizonyíthatjuk a Marsden azonosságot. $n = 0$ -ra teljesül az azonosság, tegyük fel hogy $n - 1$ -re igaz, ekkor

$$\begin{aligned} (y - x)^n &= (y - x)(y - x)^{n-1} = (y - x) \sum_{i=m-n+1}^m m_{i,n-1}(y) B_{n-1}^{i,h}(x) \\ &= \sum_{i=m-n+1}^m \left(\frac{x - t_i}{t_{n+i} - t_i} m_{i,n}(y) + \frac{t_{n+i} - x}{t_{n+i} - t_i} m_{i-1,n}(y) \right) B_{n-1}^{i,h}(x). \end{aligned}$$

Ehhez adjunk hozzá $\frac{x - t_{m-n}}{t_m - t_{m-n}} B_{n-1}^{m-n,h}(x) + \frac{x - t_{m+1}}{t_{m+1+n} - t_{m+1}} B_{n-1}^{m+1,h}(x)$ -et és indexeljük át a szumát. Mivel a hozzáadott függvény értéke 0 a $[t_m, t_{m+1}]$ intervallumon, a függvény értéke

nem változik. Végül használjuk a Cox-Deboor tételt.

$$\begin{aligned} &= \sum_{i=m-n}^m \left(\frac{x-t_i}{t_{n+i}-t_i} B_{n-1}^{i,h}(x) + \frac{t_{i+n+1}-x}{t_{n+i+1}-t_{i+1}} B_{n-1}^{i+1,h}(x) \right) m_{i,n}(y) \\ &= \sum_{i=m-n}^m m_{i,n}(y) B_n^{i,h}(x). \end{aligned}$$

□

3.4.8. Állítás. $1 = \sum_{i=m-n}^m B_n^{i,h}(t)$, ha $x \in [t_m, t_{m+1}]$.

Bizonyítás. $n - k$ -szor deriválva a Marsden azonosságot y -ra nézve, kapjuk hogy

$$\frac{n!}{k!} (y-x)^k = \sum_{i=m-n}^m \left(\frac{\partial^{n-k} m_{i,n}(y)}{\partial y^{n-k}} \right) B_n^{i,h}(x)$$

$y = 0, k = 0$ választással élve, kihasználva hogy $\frac{\partial^{n-k} m_{i,n}(y)}{\partial y^{n-k}} = n!$, adódik az állítás. □

3.4.9. Állítás (lokális függetlenség). Az $[t_m, t_{m+1}]$ intervallumon értelmezett $(B_n^{i,h})_{m-n \leq i \leq m}$ B-splineok lineárisan függetlenek.

Bizonyítás. Az előző állításban láttuk, hogy $k \leq n$ -re x^k kifejezhető B_n lineáris kombinációjaként. A legfeljebb n -ed fokú polinomok tere $n+1$ dimenziós és ezt a teret $(B_n^{i,h})_{m-n \leq i \leq m}$ vektorok generálják, azaz lineárisan függetlenek. □

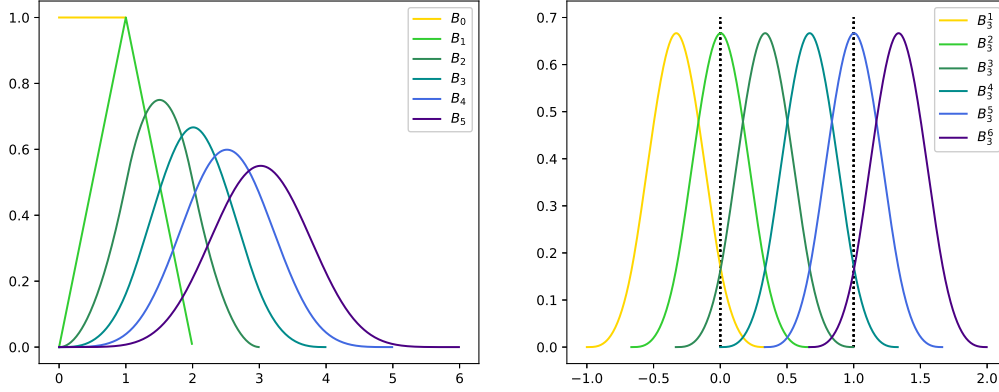
3.4.10. Definíció. Az $[a, b]$ intervallumon értelmezett n -ed fokú spline-ok terén a

$$\mathcal{S}_{n,h}([a, b]) = \{s : [a, b] \rightarrow \mathbb{R} : s = \sum_{j=1}^p c_j B_n^{j,h}, c_j \in \mathbb{R}\}$$

halmazt értjük.

3.4.11. Állítás. $\mathcal{S}_{n,h}([a, b])$ p dimenziós vektortér.

Bizonyítás. Megmutatjuk, hogy ha $\sum_{j=1}^p c_j B_n^{j,h} = 0$, akkor $c_j = 0$ minden j -re. Legyen $[t_m, t_{m+1}] \subset [a, b]$, ekkor a lokális függetlenség alapján. $\sum_{j=m-n}^n c_j B_n^{j,h} = 0$, amiből következik, hogy $c_{m-n} = \dots = c_m = 0$. Ez igaz $m = n+1, \dots, n+p+1$ -re, azaz minden i -re $c_i = 0$ teljesül. □



4. ábra. kardinális B-spline-ok és $\mathcal{S}_{3, \frac{1}{3}}([0, 1])$ bázisfüggvényei

Most definiálunk a B-spline függvények segítségével egy újabb gráfkonvolúciós műveletet. Az eddigiekkel ellentétben ez a fajta konvolúció a csúcsok szomszédságát és térbeli elhelyezkedését veszi figyelembe. Legyen adott egy $\mathcal{G} = (V, E, U)$ gráf, ahol $U \in [0, 1]^{n \times n \times d}$ tartalmazza az élek pszeudokoordinátáit. Például egy síkgráf esetén vehetjük a sík szokásos koordinátázását, $u(i, j) = (x, y)$, ahol $(i, j) \in E$ és az i csúcsot tekintjük originak. Vegyünk most d darab n -ed fokú p_i dimenziós B-spline bázist $[0, 1]$ -en. $\mathcal{B}_i^{p_i} = \{B_n^{1, h_i}, \dots, B_n^{p_i, h_i}\}$. Továbbá legyen $\mathcal{P} = \times_{i=1}^d \mathcal{B}_i^{p_i}$. Ekkor a $h_\Theta : [0, 1]^d \rightarrow \mathbb{R}$ konvolúciós kernelt definiáljuk a d dimenziós B-spline bázisok segítségével

$$h_\Theta(u) = \sum_{p \in \mathcal{P}} \theta_p B_p(u), \quad B_p(u) = \prod_{i=1}^d B_n^{p_i, h_i}(u_i).$$

képletekkel. A konvolúciós kernel mérete alatt a $k = (p_1, \dots, p_d)$ vektort értjük. A tanulandó θ_p paraméterek száma $\prod_{i=1}^d p_i$. Legyen adott egy $f : V \rightarrow \mathbb{R}$ függvény, ekkor f és egy h_Θ kernel függvény konvolúcióját

$$(f \star h_\Theta)(i) = \frac{1}{|N(i)|} \sum_{j \in N(i)} f(j) h_\Theta(u(i, j))$$

adja meg, ahol $u(i, j)$ az $(i, j) \in E$ él pszeudokoordinátája. Több channel esetén több h_Θ^i függvényre szummázzuk a konvolúciót.

4. Gráf kernelek

Ebben a fejezetben definiáljuk a kerneleket, megvizsgáljuk néhány tulajdonságukat, és mutatunk egy módszert, a kernelek és a Support Vector Machine segítségével klasszifikációs feladatokat megoldására [13]. Végül ismertetjük a Weisfeiler-Lehamn algoritmuson alapuló gráfkernelt [14].

4.1. Kernelek karakterizációja

4.1.1. Definíció. Egy $k : X \times X \rightarrow \mathbb{R}$ függvényt kernel függvénynek nevezünk, ha minden $x, y \in X$ -re teljesül $k(x, y) = \langle \phi(x), \phi(y) \rangle$, ahol $\phi : X \rightarrow F$ az úgynevezett feature leképezés, F pedig valamilyen skalárszorozattal ellátott feature tér.

4.1.2. Definíció. Az $\{x_1, x_2, \dots, x_n\} \subset X$ pontokhoz és a k kernelhez tartozó $K_{ij} = k(x_i, x_j)$ mátrixot, Gram mátrixnak nevezzük.

4.1.3. Definíció. Egy szimmetrikus $k : X \times X \rightarrow \mathbb{R}$ függvényt pozitív szemidefinit függvénynek nevezünk, ha a függvényhez tartozó bármely Gram mátrix pozitív szemidefinit, azaz ha bármely $y \in \mathbb{R}^n$ és bármely $x_1, \dots, x_n \in X$ -re $\sum_{i,j=1}^n k(x_i, x_j) y_i y_j \geq 0$.

4.1.4. Állítás. Egy k kernelhez tartozó Gram mátrix pozitív szemidefinit.

Bizonyítás. Legyen $K_{ij} = k(x_i, x_j)$ és $y \in \mathbb{R}^n$ tetszőleges vektor.

$$\begin{aligned} \sum_{i,j=1}^n k(x_i, x_j) y_i y_j &= \sum_{i,j=1}^n \langle \phi(x_i), \phi(x_j) \rangle y_i y_j = \\ &= \left\langle \sum_{i=1}^n y_i \phi(x_i), \sum_{j=1}^n y_j \phi(x_j) \right\rangle = \left\| \sum_{i=1}^n y_i \phi(x_i) \right\|^2 \geq 0. \end{aligned}$$

□

4.1.5. Tétel. Egy $k : X \times X \rightarrow \mathbb{R}$ folytonos függvény akkor és csak akkor pozitív szemidefinit, ha létezik ϕ feature leképezés és egy \mathcal{H} Hilbert-tér, amire $k(x, y) = \langle \phi(x), \phi(y) \rangle$.

Bizonyítás. Az előző állításban beláttuk, hogy egy kernel mindig pozitív szemidefinit. Visszafelé, tegyük fel, hogy adott egy pozitív szemidefinit függvényünk. Célunk mutatni, egy ϕ leképezést és egy \mathcal{H} Hilbert-teret, amiben k -re teljesül a kernel tulajdonság.

$$\mathcal{H}_0 = \left\{ f(t) : f(t) = \sum_{i=1}^n \alpha_i k(x_i, t), x_i \in X, \alpha_i \in \mathbb{R} \right\}$$

\mathcal{H}_0 nyilvánvalóan vektortér, mivel összeadásra és skalárral való szorzásra zárt. Legyen $f(t) = \sum_{i=1}^n \alpha_i k(x_i, t)$ és $g(t) = \sum_{i=1}^l \beta_i k(y_i, t)$, definiáljuk a \mathcal{H} -n való skalárszorzást a következőképpen

$$\langle f, g \rangle = \sum_{i=1}^n \sum_{j=1}^l \alpha_i \beta_j k(x_i, y_j) = \sum_{i=1}^n \alpha_i g(x_i) = \sum_{j=1}^l \beta_j f(y_j).$$

Ez a definíció valóban félskalárszorzást ad meg \mathcal{H}_0 -n, mivel szimmetrikus, bilineáris és k pozitív szemidefinitását kihasználva kapjuk, hogy $\langle f, f \rangle \geq 0$, mivel $k(x_i, x_i) \geq 0$. Legyen $f(t) = \sum_{i=1}^n \alpha_i k(x_i, t)$ és $g(t) = k(x, t)$ valamilyen $x \in X$ -re, ekkor reprodukciós tulajdonságnak hívjuk a következő összefüggést

$$\langle f, g \rangle = \langle f, k(x, \cdot) \rangle = \sum_{i=1}^n \alpha_i k(x_i, x) = f(x).$$

Ezt felhasználva kapjuk, hogy ha $\|f\| = 0$, akkor

$$f(x) = \langle f, \phi(x) \rangle \leq \|f\| \|\phi(x)\| = 0$$

azaz, $f = 0$, tehát $\langle \cdot, \cdot \rangle$ nemcsak félskalárszorzat, hanem skalárszorzat. Végül teljessé tesszük \mathcal{H}_0 , ahol már minden Cauchy-sorozat konvergens. Legyen $x \in X$ és $(f_n)_{n=1}^{\infty}$ Cauchy-sorozat. Ekkor használva a reprodukciós tulajdonságot és a Cauchy-Schwarz egyenlőtlenséget

$$(f_n(x) - f_m(x))^2 = \langle f_n - f_m, k(x, t) \rangle \leq \|f_n - f_m\|^2 k(x, x).$$

Tehát $f_n(x)$ \mathbb{R} -beli Cauchy sorozat, azaz konvergens. Legyen $h(x) = \lim_{n \rightarrow \infty} f_n(x)$. Bővítsük ki \mathcal{H}_0 -t ezen h függvényekkel, így \mathcal{H} Hilbert-teret kapjuk. Most rátérünk a ϕ leképezésre. Legyen $\phi(x) = k(x, \cdot) \in \mathcal{H}$. Felírva a reprodukciós tulajdonságot $\phi(x)$ és $\phi(y)$ függvényekre, kapjuk hogy $\langle \phi(y), \phi(x) \rangle = \langle k(y, \cdot), k(x, \cdot) \rangle = \phi(y)(x) = k(x, y)$, ami éppen a kernel tulajdonság. \square

4.1.6. Állítás. \mathcal{H} Hilbert-téren egy szimmetrikus $k(\cdot, \cdot)$ függvényre akkor és csak akkor teljesül a reprodukciós tulajdonság, azaz $\langle f, k(x, \cdot) \rangle = f(x)$ minden $f \in \mathcal{H}$ -ra, ha k pozitív szemidefinit.

Bizonyítás. Ha k pozitív szemidefinit, akkor az előző tétel értelmében teljesül a reprodukciós tulajdonság. Visszafelé pedig

$$\begin{aligned} \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j) &= \sum_{i,j=1}^n \alpha_i \alpha_j \langle k(x_i, \cdot), k(x_j, \cdot) \rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i k(x_i, \cdot), \sum_{j=1}^n \alpha_j k(x_j, \cdot) \right\rangle = \left\| \sum_{i=1}^n \alpha_i k(x_i, \cdot) \right\|^2 \geq 0. \end{aligned}$$

□

4.1.7. Állítás. Legyen X metrikus tér. Ha $k : X \times X \rightarrow \mathbb{R}$ folytonos, akkor $\phi : X \rightarrow \mathcal{H}$ folytonos.

Bizonyítás. Legyen $(x_n)_{n=1}^\infty \subset X$ és $x_n \rightarrow x \in X$.

$$\begin{aligned} \|\phi(x_n) - \phi(x)\|^2 &= \langle k(x_n, \cdot), k(x_n, \cdot) \rangle - 2 \langle k(x_n, \cdot), k(x, \cdot) \rangle + \langle k(x, \cdot), k(x, \cdot) \rangle \\ &= k(x_n, x_n) - 2k(x_n, x) + k(x, x) \rightarrow 0. \end{aligned}$$

□

4.1.8. Definíció. Egy (X, d) metrikus tér szeparábilis, ha létezik benne megszámlálható sűrű részhalmaz, azaz minden ε -hoz létezik $(v_i)_{i=1}^\infty \subset X$ úgy, hogy minden $v \in X$ -re $\min d(v, v_i) < \varepsilon$.

4.1.9. Lemma. Legyen (X, d) kompakt metrikus tér, ekkor

1) (X, d) szeparábilis.

2) Ha (X', d') metrikus-tér és $f : X \rightarrow X'$ folytonos, akkor $f(X)$ is kompakt.

4.1.10. Tétel (Mercer). Legyen $X \subset \mathbb{R}^n$ kompakt részhalmaza és $k : X \times X \rightarrow \mathbb{R}$ folytonos, szimmetrikus függvény, amire az

$$(T_k f)(\cdot) = \int_X k(x, \cdot) f(x) \, dx$$

$T_k : L_2(X) \rightarrow L_2(X)$ operátor pozitív, azaz minden $f \in L_2(X)$ -re

$$\int_{X \times X} k(x, y) f(x) f(y) dx dy \geq 0.$$

Ekkor léteznek $(\phi_j)_{j=1}^\infty \subset L_2(X)$ ortonormált sorozat, amire $k(x, y) = \sum_{j=1}^\infty \phi_j(x) \phi_j(y)$ és $\sum_{j=1}^\infty \|\phi_j\|_{L_2(X)}^2 < \infty$.

Bizonyítás. Belátjuk, hogy k pozitív szemidefinit. Legyen $f(x) = \sum_{i=1}^l \alpha_i \mathbf{1}_{x_i}(x) \in L_2(X)$. Ekkor k pozitívitése éppen a véges értelemben vett pozitív szemidefinitiséget adja vissza. Mivel k pozitív szemidefinit, így létezik egy \mathcal{H} Hilbert-tér, ahol teljesül a reprodukciós tulajdonság. Mivel X kompakt, ezért szeparábilis. Továbbá $\phi : X \rightarrow \mathcal{H}$ leképezés folytonos, azaz $\phi(X)$ is kompakt. Ebből már következik, hogy $\overline{\text{span } \phi(X)}$ is szeparábilis. Ismert, hogy szeparábilis Hilbert-téren létezik teljes ortonormált sorozat, ez legyen $(\phi_i)_{i=1}^\infty$, tehát k -t Fourier-sorba fejthetjük ez alapján. A sorbafejtésnél használva a reprodukciós tulajdonságok, kapjuk, hogy

$$k(x, y) = \sum_{i=1}^\infty \langle k(x, \cdot), \phi_i(\cdot) \rangle \phi_i(y) = \sum_{i=1}^\infty \phi_i(x) \phi_i(y).$$

Végül a monoton konvergenciatételt használva,

$$\begin{aligned} \infty > \int_X k(x, x) dx &= \int_X \lim_{n \rightarrow \infty} \sum_{i=1}^n \phi_i(x) \phi_i(x) dx = \lim_{n \rightarrow \infty} \sum_{i=1}^n \int_X \phi_i(x) \phi_i(x) dx \\ &= \lim_{n \rightarrow \infty} \sum_{i=1}^n \|\phi_i\|_{L_2(X)}^2. \end{aligned}$$

□

4.2. Support Vector Machine

Legyen $\mathcal{S} = \{(x_1, y_1), \dots, (x_l, y_l)\}$ egy klasszifikációs probléma, ahol $x_i \in \mathbb{R}^n$ magyarázó adatok és $y_i \in \{-1, 1\}$ az adatokhoz tartozó kategóriák. Célunk egy szeparáló hipersíkot találni, ami elválasztja egymástól a két osztályt. Ehhez az adathalmazra először egy $\phi : \mathbb{R}^n \rightarrow \mathcal{H}$ leképezést alkalmazunk, ezt nevezzük kernel trükknek. Legyen $g(x) = \langle w, \phi(x_i) \rangle + b$ lineáris függvény. Azt mondjuk, hogy x_i és x_j pontokat szeparálja a $g(x) = 0$ hipersík, ha $g(x_i)$ és $g(x_j)$ előjele különböző. Keressünk tehát olyan $\gamma \in \mathbb{R}$ margót, amire $\xi_i = (\gamma - y_i g(x_i))_+ = 0$ ha $1 \leq i \leq l$. $\xi_i = 0$ pontosan akkor teljesül, ha

$y_i g(x_i) \geq \gamma$, tehát ha $\phi(x_i)$ legalább γ távolságra van $g(x) = 0$ hipersíktól. Ha \mathcal{H} -ban $\phi(x_i)$ pontok szeparálhatóak, akkor w együtthatókat, b eltolásvektort és γ értékét kell meghatározni. Azonban általában ez nem érhető el, így a célfüggvénybe a ξ_i slack változókkal büntetjük a rosszul klasszifikált adatpontokat. Továbbá C hiperparaméterrel súlyozzuk a büntetés mértékét. A következő feltételes optimalizálás feladat fogalmazható meg

$$\min\{-\gamma + C \sum_{i=1}^l \xi_i : (w, b, \gamma, \xi) \in D \subset \mathbb{R}^{2l+2}\}$$

$$D = \{(w, b, \gamma, \xi) \in \mathbb{R}^{2l+2} : y_i(\langle w, \phi(x_i) \rangle + b) \geq \gamma - \xi_i, \xi_i \geq 0, \|w\|^2 = 1, 1 \leq i \leq l\}.$$

4.2.1. Tétel (Karush–Kuhn–Tucker). *Legyenek f , $(g_i)_{i=1}^n$, $(h_j)_{j=1}^m$ differenciálható függvények. Az alábbi feltételes optimalizálási problémára*

$$\begin{aligned} \min\{f(x) : x \in X\} \\ X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, h_j(x) = 0\} \end{aligned}$$

teljesülnek a KKT feltételek, ha

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^n \alpha_i^* \nabla g_i(x^*) + \sum_{j=1}^m \beta_j^* \nabla h_j(x^*) &= 0, \\ g_i(x^*) &\leq 0, \\ h_j(x^*) &= 0, \\ \alpha_i^* &\leq 0, \\ \alpha_i^* g_i(x^*) &= 0. \end{aligned}$$

Konvex optimalizálási problémákra, amelyre teljesül az erős dualitás tétel ($\min f(x) = \max L(\alpha, \beta)$), a KKT feltételek pontosan akkor teljesülnek, ha x^ primál és (α^*, β^*) duál optimum.*

Bizonyítás nélkül állítjuk, hogy az SVM problémára teljesül az erős dualitás tétel és a

feladat konvex. Oldjuk meg a duális feladatot, ahol $\alpha_i \geq 0$, $\beta_i \geq 0$ és λ duális változók.

$$L = -\gamma + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i [y_i (\langle w, \phi(x_i) \rangle + b) - \gamma + \xi_i] - \sum_{i=1}^l \beta_i \xi_i + \lambda (\|w\|^2 - 1)$$

$$\frac{\partial L}{\partial w} = 2\lambda w - \sum_{i=1}^l y_i \alpha_i \phi(x_i) = 0, \quad \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^l y_i \alpha_i = 0, \quad \frac{\partial L}{\partial \gamma} = 1 - \sum_{i=1}^l \alpha_i = 0.$$

Visszahelyettesítve L -be,

$$L = -\gamma + C \sum_{i=1}^l \xi_i (C - \alpha_i - \beta_i) - \sum_{i=1}^l \alpha_i y_i \langle w, \phi(x_i) \rangle - b \sum_{i=1}^l y_i \alpha_i + \gamma \sum_{i=1}^l \alpha_i + \lambda (\|w\|^2 - 1)$$

$$- \sum_{i=1}^l \alpha_i y_i \langle w, \phi(x_i) \rangle + \lambda \|w\|^2 - \lambda = \left(\frac{1}{4\lambda} - \frac{1}{2\lambda} \right) \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \lambda.$$

Ebből kiszámolva λ -ra vonatkozó parciális deriváltat kapjuk

$$\lambda^* = \sqrt{\frac{1}{2} \left(\sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j k(x_i, x_j) \right)}.$$

Behelyettesítve kapjuk, hogy

$$\max \left\{ L(\alpha) = -\sqrt{\sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j)} : \alpha \in K \right\}$$

$$K = \left\{ \alpha \in \mathbb{R}^l : \sum_{i=1}^l y_i \alpha_i = 0, \sum_{i=1}^l \alpha_i = 1, 0 \leq \alpha_i \leq C \right\}.$$

A KKT feltételekből következik, hogy $\alpha_i [y_i (\langle w, \phi(x_i) \rangle + b) - \gamma + \xi_i] = 0$ és $\xi_i (\alpha_i - C) = 0$. A nemnulla ξ_i slack változókhoz tartozó indexre $\alpha_i = C$. Ha $\xi_i = 0$, akkor csak azokhoz az x_i vektorokhoz tartozó α_i duális együtthatók nem nullák, amik a legközelebb vannak a szeparáló hipersíkhöz. Ezeket a vektorokat hívjuk szupport vektoroknak. Ha már megvan α^* , vegyünk x_i és x_j pontokat különböző osztályból, amikre $0 \leq \alpha_i, \alpha_j \leq C$. Ezekre a

változókra $\xi_i = 0$, tehát

$$\begin{aligned} y_i(\langle w^*, \phi(x_i) \rangle + b^*) - \gamma^* &= 0 = y_j(\langle w^*, \phi(x_j) \rangle + b^*) - \gamma^* \\ b^* &= -\frac{1}{2}(\langle w^*, \phi(x_i) \rangle + \langle w^*, \phi(x_j) \rangle) \\ \gamma^* &= \langle w^*, \phi(x_j) \rangle + b^* \\ w^* &= \frac{1}{2\lambda^*} \sum_{i=1}^l y_i \alpha_i \phi(x_i). \end{aligned}$$

Tehát új adatpontok klasszifikálására az $f(\cdot) = \text{sgn}\left(\frac{1}{2\lambda^*} \sum_{i=1}^l \alpha_i^* y_i k(x_i, \cdot) + b^*\right)$ függvényt használhatjuk. A duál optimalizálási probléma a kernel pozitív szemidefinitiségének köszönhetően konvex, gradiens módszer segítségével megoldható.

4.3. Weisfeiler-Lehman kernel

Ebben a szakaszban bemutatjuk az úgynevezett Weisfeiler-Lehman kernelt, amely két gráf közötti hasonlóságokat hivatott számszerűsíteni.

Legyen adott egy kellően nagy Σ ábécé ami a lehetséges csúcscímkeket tartalmazza. Ekkor a Weisfeiler-Lehman algoritmus kiindulva egy $l_0 \in \Sigma^{|V|}$ címkézésből újabb címkézéseket gyárt le, ahol az egyes csúcsokhoz tartozó új címkék a szomszédos csúcsok előző iterációbeli címkéitől függenek. Az új címkézés úgy jön létre, hogy az előző értékeket konkatenáljuk, ezt jelöljük $s(v)$ -vel, és egy olyan f függvényt alkalmazunk amire igaz, hogy $f(s(v)) = f(s(w))$ akkor és csak akkor teljesül, ha $s(v) = s(w)$. Ezzel biztosítva, hogy ha két csúcsnak különbözik a szomszédosága, akkor az algoritmus különböző címkékkel fogja ellátni őket.

Algorithm 1: Weisfeiler-Lehman

Input : $\{\mathcal{G}_i, l_i^{(0)}\}_{i=1}^d$, : gráfok és kezdeti címkézéseik,
 Σ : ábécé, $f : \Sigma^* \rightarrow \Sigma$, N : iterációk száma

Output: $\{l_i^{(N)}\}_{i=1}^d$

```

1 for  $j = 1$  to  $N$  do
2   | Minden  $\mathcal{G}_i$  gráfra és  $v$  csúcsra:
3   |  $s_i^{(j)}(v) \leftarrow \text{concat}(l_i^{(j-1)}(v), \text{sort}(\{l_i^{(j-1)}(u) : u \in N(v)\}))$ ;
4   |  $l_i^{(j)}(v) \leftarrow f(s_i^{(j)}(v))$ 
5 end

```

A WL-algoritmust szokás gráf izomorfia tesztelésére használni. Legyen az inputunk két gráf, és kezdeti címkézésnek ugyanolyan értékeket adjunk meg minden csúcshoz. Ekkor ha egy iteráció után a $\{l^{(i)}(v) : v \in V\} \neq \{l'^{(i)}(v) : v' \in V'\}$ halmazokra, akkor a két gráf nem izomorf. Azonban mutathatóak olyan gráfok, amiket a WL-algoritmus nem tud megkülönböztetni, viszont mégsem izomorfak, ilyenek például az ugyanolyan csúcshatárú k reguláris gráfok. Egy iteráció futási ideje a rendezéstől függ, mivel $|E| = m$, ezért minden lerendezendő halmaz mérete legfeljebb m akkor számláló rendezéssel ez $\mathcal{O}(m)$ műveletet igényel, N iteráció után $\mathcal{O}(Nm)$, több gráf esetén az f függvény kiszámításához szükséges idő $\mathcal{O}(Ndm)$.

4.3.1. Definíció. Legyen k az alapkernel és jelölje $\{\mathcal{G}_0, \dots, \mathcal{G}_N\} = \{(V, E, l_0), \dots, (V, E, l_N)\}$ a Weisfeiler-Lehman algoritmus által megkapott gráfsorozatot. Ekkor a WL-kernel alatt a $k_{WL}^{(N)} = \sum_{i=0}^N k(\mathcal{G}_i, \mathcal{G}'_i)$ kifejezést értjük.

4.3.2. Állítás. Ha a k alapkernel pozitív szemidefinit, akkor $k_{WL}^{(N)}$ pozitív szemidefinit.

4.3.3. Definíció. Legyenek \mathcal{G} és \mathcal{G}' gráfok és jelölje $\Sigma_i \subset \Sigma$ azokat a címkéket, amik a WL-algoritmus i . iterációja után előfordulnak $\{l^{(i)}\} \cup \{l'^{(i)}\}$ halmazban. Tegyük fel, hogy $\Sigma_i = \{\sigma_{i1}, \dots, \sigma_{i|\Sigma_i|}\}$ halmazok páronként diszjunktak és rendezettek. Jelölje $c_i(\mathcal{G}, \sigma_{ij})$ azt, hogy az i . iteráció után σ_{ij} hányszor fordult elő \mathcal{G} -ben. Ekkor WL Subtree kernelnek nevezzük a $k_{Subtree}^{(N)}(\mathcal{G}, \mathcal{G}') = \langle \phi^{(N)}(\mathcal{G}), \phi^{(N)}(\mathcal{G}') \rangle$, ahol $\phi^{(N)}(\mathcal{G}) = (r_1, \dots, r_n)$ és $r_i = (c_i(\mathcal{G}, \sigma_{i1}), \dots, c_i(\mathcal{G}, \sigma_{i|\Sigma_i|}))$.

4.3.4. Állítás. Legyen $k(\mathcal{G}, \mathcal{G}') = \sum_{v \in V} \sum_{v' \in V'} \delta(l(v), l'(v'))$, ahol δ értéke 1 ha $l(v) = l'(v')$, különben 0. Ekkor $k_{WL}^{(N)} = k_{Subtree}^{(N)}$, ha $k_{WL}^{(N)}$ alapkernel k .

Bizonyítás. Vegyük észre, hogy az i . iteráció után $k(\mathcal{G}, \mathcal{G}') = \sum_{j=1}^{|\Sigma_i|} c_i(\mathcal{G}, \sigma_{ij}) c_i(\mathcal{G}', \sigma_{ij})$. Tehát mindkét esetben azt számoltuk meg, hogy hány egyező címképar szerepel. Minden iterációra összegezve adódik az állítás. \square

A $\phi_{WL}^{(N)}$ kiszámításához el kell végezni a WL-algoritmust d darab gráfra, majd összeszámolni az egyes címkék számát, ez $\mathcal{O}(Ndm)$ műveletet vesz igénybe. Ahhoz hogy megkapjuk a $d \times d$ kernel mátrixát, ki kell számolni d gráfra a páronkénti értékeket, ez $\mathcal{O}(d^2 Nn)$ művelet. Tehát a WL Subtree kernel futási ideje összesen $\mathcal{O}(dNm + d^2 Nn)$.

5. Kísérletek

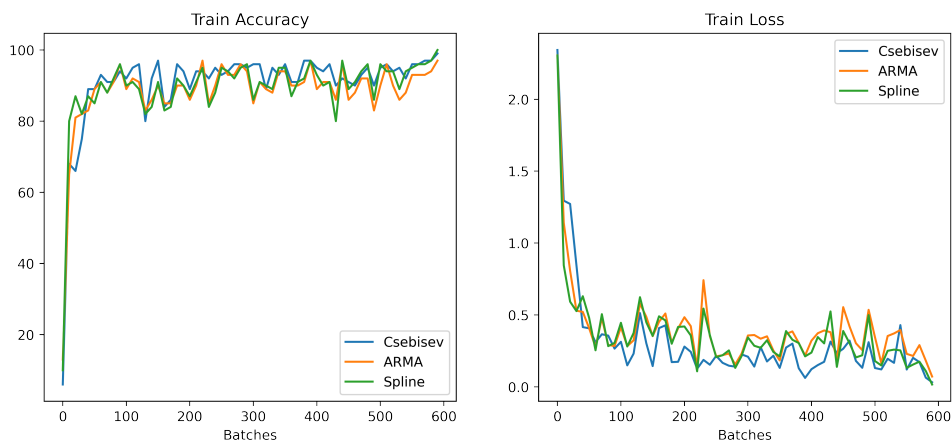
A fejezetben a gráfkonvolúciós modellek alkalmazásait vizsgáljuk meg. A képklasszifikációs feladatnál a [5] cikk eredményeinek reprodukálása volt a cél, a molekula felismerés a [15] cikkben alapszik. A futtatásokat az ELTE szerverén végeztem, amely Ubuntu 20.04.2 LTS Linux operációs rendszert használ. A gépekben GeForce RTX 2080 Ti grafikus processzor található. A konvolúciós hálók implementációja Python nyelven történt, a Pytorch és a Pytorch Geometric [16] könyvtárak segítségével, a gráfkernel futtatásához a grakel [17] könyvtárat vettem segítségül. A kísérletkhez tartozó kódok elérhetőek a <https://github.com/aielte-research/gcn-harsanyi> linken.

5.1. Képlaszifikáció rácsgráfokkal

Első alkalmazásként tekintsünk egy képklasszifikációs feladatot az MNIST adathalmazon. Az adathalmaz 60000 darab fényképet tartalmaz, mindegyik fényképen egy egyjegyű számjegyet látható. Egy képet egy $A \in [0, 1]^{28 \times 28}$ mátrix reprezentál, ahol a mátrix elemei határozzák meg, hogy a képen az adott pixel mennyire sötét. Kódoljuk el a képet egy $\mathcal{G} = (V, E, x)$ rácsgráfon. A gráfnak $|V| = 28 \times 28$ csúcsa van, ahol minden csúcsot $v_{ij} = [i, j]$ koordinátapárokkal adhatunk meg. két csúcs között behúzzunk élet, ha $\|v - w\|_2 = 1$, ami éppen egy rácsgráfot eredményez, vízszintes és függőleges élekkel. A képet a csúcsokon értelmezett $x : V \rightarrow [0, 1]$ függvény határozza meg. Az adathalmazunk tehát egy $\{G_i, y_i\}_{i=1}^{60000}$ halmaz, ahol a G_i gráfhoz tartozó y_i címke mondja meg, hogy a képen hányas számjegy látható. Ezekhez a címkékhez rendeljük egy y'_i one-hot encoding 10 elemű vektort, amely ha a számjegyünk i , akkor az $i - 1$. eleme 1, a többi pedig 0. A feladat egy olyan függvény keresése, amire $f_{\Theta}(G_i) = y_i$. Az f_{Θ} függvény egy gráf konvolúciós rétegből és egy sűrű rétegből tevődik össze, ahol a gráf konvolúciós rétegben Csebisev, ARMA filtereket és a B-Spline konvolúciót alkalmaztam. Az output csatornák számát mindhárom esetben 10-nek választottam. A konvolúciós réteg után ReLu nemlinearitásokat használtam, majd egy $A \in \mathbb{R}^{(28 \times 28 \times 10) \times 10}$ sűrű réteggel projektáltam a eredményt, végül a kapott $z \in \mathbb{R}^{10}$ vektort egy $\hat{y}_i = \text{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{i=1}^{10} e^{z_i}}$ függvénnyel valószínűségi eloszlássá alakítom. A neurális hálózat becslése tehát az $\text{argmax} \hat{y}$ érték lesz. A veszteségfüggvény a többkategóriás kereszt-entrópia függvény, amelyet a $L(x, y) = - \sum_{j=1}^{10} y_j \log(\hat{y}_j)$ képlet ír le.

Model	Hiperparam.	Paraméterek száma	Futási idő (sec)	Pontosság (teszt)	Loss (teszt)
Csebisev	K=25	260+78,410	19.3	96.91 ± 0.05	0.09 ± 0.00
ARMA	K=3, T=3	810+78,410	24.3	90.42 ± 0.75	0.33 ± 0.00
Spline	kernel=(5,5) n=1	270+78,410	14.6	91.52 ± 8.56	0.28 ± 0.01

A gradiensek frissítéséhez az ADAM algoritmust használtam 0.03-as learning rate-tel, a tanítást 50000 darab képen végeztem 100-as batch mérettel és 1 epoch-kal. A hálót 10000 darab képen teszteltem a tanítási folyamat után. A Csebisev polinomok fokszámát $K=25$ választottam, Az ARMA filterek esetén a racionális törtfüggvény számlálója és nevezője is egyaránt $K = 3$ lett választva és $T = 3$ jelöli az $ARMA_1$ filter közelítéséhez használt rekurzió iterációs számát. A Spline függvények esetén a rácsgráfhoz 2 dimenziós $n = 1$ fokú bázisfüggvényeket vettünk (5,5)-ös kernel mérettel. Az élekhez tartozó u pszeudokordinátáknak a csúcsok relatív Descartes koordinátáit választottam, ez azt eredményezi, hogy a Spline konvolúció rácsgráfokon éppen visszaadja a klasszikus (CCN) konvolúciót. A Spline (5,5) kernele éppen egy 5×5 -ös mátrix kernelnek felel meg a normalizációtól eltekintve. A kísérleteket 7 alkalommal végeztem el, a futási időkből átlagot számoltam, hasonlóan a pontosság és a loss függvény esetén a táblázatban megtalálható értékek az átlagokat jelzik korrigált tapasztalati szórásokkal.



A grafikonokon a jól olvashatóság érdekében csak minden 10. batch eredménye lett ábrázolva, a háló körülbelül 100 batch után éri el a 90%-os eredményt.

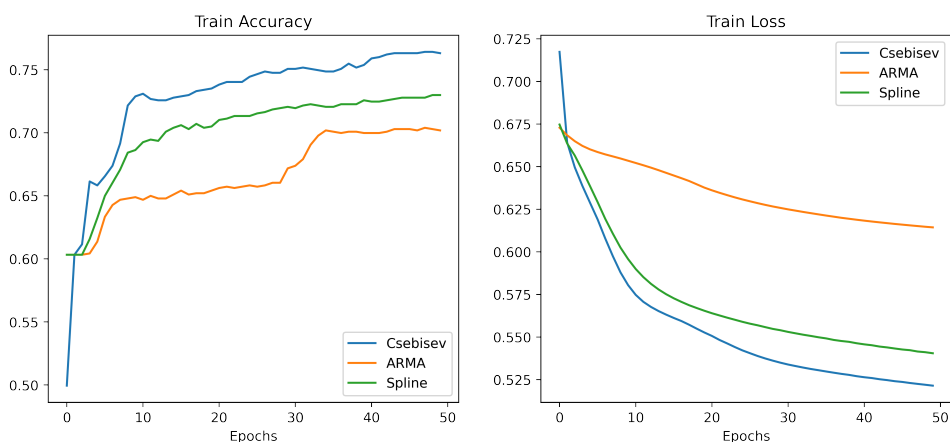
5.2. Molekula osztályozás

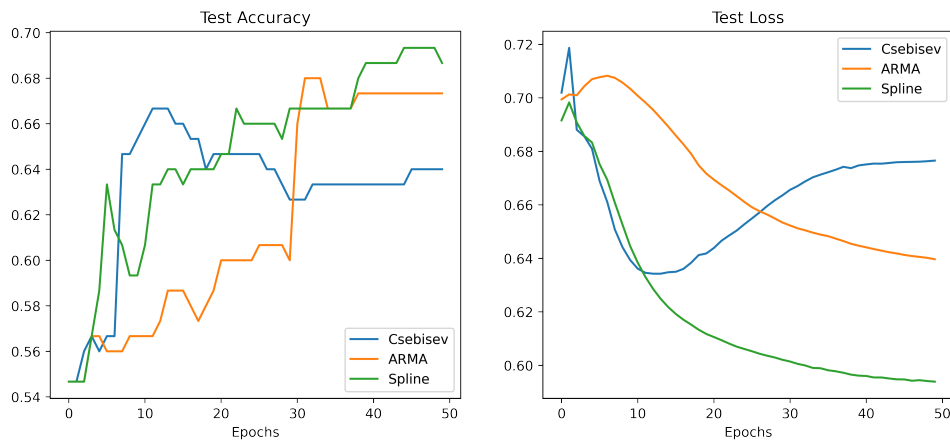
A PROTEINS [15] adathalmaz 1113 darab fehérje molekulát tartalmaz. A feladat a molekulák osztályozása, aszerint, hogy az adott fehérje enzim-e. A fehérjék gráfként vannak reprezentálva, a gráf csúcsai a másodlagos szerkezeti elemeknek felelnek meg. A csúcsokon egy három dimenziós one-hot vektoron tároljuk el, hogy az adott polipeptidlánc térbeli elhelyezkedését mi jellemzi. Ez lehet véletlenszerű felcsavarodás (hurok) vagy szabályos szerkezetű α -hélix, illetve β -redő. Egy élet két csúcs között akkor húzunk be, ha az aminosav-szekvencia mentén szomszédosak.

Model	Hiperparam.	Paraméterek száma	Futási idő (sec)	Pontosság (teszt)	Loss (teszt)
Csebisev	K=10	930+60+62	0.19	0.79 ± 0.01	0.71 ± 0.00
ARMA	K=3, T=1	3300+60+62	0.15	0.78 ± 0.02	0.72 ± 0.00
Spline	kernel=(10), n=1	1020+60+62	0.15	0.79 ± 0.02	0.74 ± 0.00

Hasonlóan a képklasszifikációs feladathoz, a célunk egy $f_{\Theta}(\mathcal{G}_i) = y_i$ függvény konstruálása, ami az adott molekulagráfra megmondja, hogy enzim-e. A háló architektúrája egy gráf konvolúciós rétegből, 3 input és 10 output channel-lel, egy batch normalization rétegből, global mean pool rétegből és végül egy sűrű rétegből tevődik össze. A konvolúciós rétegek után ReLu aktivációs függvényt használtam. Az előző feladathoz hasonlóan a három futtatást végeztem, Csebisev, ARMA és Spline konvolúcióval. A batch normalization réteg az adott batchben szereplő gráfok csúcsattribútumainak szórásának és várható értékének manipulálására szolgál két újabb tanuló paraméter bevezetésével. Ha egy batch a $\{(\mathcal{G}_i, x_i)\}_{i=1}^d$ adatokból áll, akkor a $\frac{x_i - \mu}{\sqrt{\sigma - \epsilon}}\gamma_j + \beta_j$ transzformációt hajtjuk végre, azaz minden output channelre különböző (β_j, γ_j) normalizációs együtthatókat alkalmazunk. A pooling réteg pedig a megtanult reprezentáció tömörítésére szolgál. Ha a j . channel outputja x'_j , akkor a pooling layer $r_j = \frac{1}{|V_i|} \sum_{i=1}^{|V_i|} x_j(i)$ műveletet végzi el, végül az $r = (r_1, \dots, r_m)$ reprezentációt kapva m channel esetén. Ennek a műveletnek az az előnye, hogy a megtanult belső reprezentáció mérete nem függ az input gráf méretétől, a molekulagráfok mérete különböző, viszont a megtanult reprezentáció channel méretű vektor lesz minden esetben. Ezt a fix méretű vektort pedig egy sűrű réteggel projektálni tudjuk a klasszifikációs feladatnak megfelelően 2 dimenzióra. Hasonlóan az előző feladathoz a veszteségfüggvény két kategóriás kereszt-entrópia, amely előtt még egy Softmax függvényt alkalmazunk.

A futtatások során az output csatornák száma 30-nak lett választva, így a klasszifikáció előtti reprezentáció 30 dimenziós lett. Ez azt jelenti, hogy a batch normalization layerben 60 paraméter, míg a sűrű réteg $30 \cdot 2 + 2$ paramétert tartalmaz. A gradiensek frissítésére megint az ADAM algoritmust használtam 0.003-mas learning rate-tel. Az adathalmazt két részre bontottam, 150 molekulán végeztem a tesztek és $1113 - 150 = 963$ molekulát használtam tanításra, amelyet 50 epochon keresztül 100-as batch méretekkel végeztem el. Összesen 7 futást végeztem mindhárom esetben, minden futás során különböző train-test vágásokat hajtottam végre az adathalmazon. A futások eredményei a fenti táblázatban láthatóak. A futási időnél átlagot számoltam, pontosság és a loss esetén az 50 epoch közötti legjobb futások átlagát vettem, továbbá leolvashatóak a korrigált tapasztalati szórások is. A Spline konvolúció esetén a pszeudokoordinátáknak a csúcsok relatív fokszámát választottam, azaz $u(e_{ij}) = \frac{\text{deg}(v_j)}{\max_{v \in V}(\text{deg}(v))}$, ami 1 dimenziós kernelt eredményezett. Láthatóan a futási idők lényegesen gyorsabbak mint az MNIST esetén, ez az adathalmaz méretével magyarázható, míg itt batchenként 100 gráfot adunk át inputként, egy gráfnak átlagosan 39 csúcsa és 145 éle van, az MNIST esetén egy gráfnak 784 csúcsa volt, és a gráfok száma 60000, az itteni 1000-res nagyságrenddel szemben.





A Csebisev és ARMA filterek esetén túltanulás figyelhető meg. Ennek az lehet az oka hogy az adathalmaz különböző struktúrájú gráfokból áll, amihez a spektrálfilterek nem tudnak jól alkalmazkodni. A különböző train-test vágásokon az eredmények jelentősen eltértek.

A neurális hálókön túl kipróbáltam a Weisfeiler-Lehman kernelen alapuló klasszifikációt is. A WL-algoritmust 5 iterácival hajtottam végre a gráfokon, majd a kapott címkesorozatok segítségével kiszámoltam a WL Subtree kernel értékeit a tanuló adathalmazon, majd a kernelizált Support Vector Machine segítségével klasszifikáltam az adatpontokat. Az előzőekhez hasonlóan 150 adatpontot választottam teszt adathalmaznak, ezeken a pontokon az klasszifikáció eredménye 75.33% lett.

6. Alkalmazások

Az ajánlórendszerekről írottakhoz a [18] cikket vettem alapul és a [19] alapján készült a független csúcshalmaz keresési problémáról szóló alfejezet.

6.1. Mátrix kiegészítési probléma

Ebben a szakaszban az ajánló rendszereket (Recommendation Systems) járjuk körül. A feladat az, hogy megfelelő termékeket ajánljunk a felhasználóknak (például lehetséges ismerősöket a közösségi média platformokon). Adottnak tekintünk egy felhasználói bázist és közöttük egy kapcsolati rendszert, ezt $\mathcal{G}_u = (V_u, E_u, W_u)$ (users) élsúlyozott gráffal jelöljük, továbbá adott egy hasonló struktúra a termékekre is, ahol az élek és a rajtuk szereplő súlyok a termékek hasonlóságát hivatottak leírni, ezt $\mathcal{G}_p = (V_p, E_p, W_p)$ (products) gráffal jelöljük. A felhasználókat és a termékeket egy $X \in \mathbb{R}^{|V_u| \times |V_p|}$ mátrixként foghatjuk fel, ahol a megfelelő x_{ij} elem azt mondja meg, hogy az i . felhasználónak mennyira ajánlanánk a j . terméket. Adottnak tekintjük továbbá a mátrixnak néhány elemét, tehát ismerjük néhány felhasználó és termék kapcsolatát. Azaz adott egy $\Omega = \{(i, j) : i \in \{1, \dots, |V_u|\}, j \in \{1, \dots, |V_p|\}\}$ részhalmaz és $x_{ij} = y_{ij}$ elemek, ahol $(i, j) \in \Omega$. Omegára gondolhatunk úgy, mint egy mátrixra: $\Omega_m \in \{0, 1\}^{|V_u| \times |V_p|}$, ahol $\omega_{ij} = 1$ ha $(i, j) \in \Omega$. 2009-ben a Netflix 1.000.000\$-os díjazású versenyt hirdetett, ahol a cél olyan algoritmus kreálása volt ami meglévő filmértékelések alapján ajánlott új filmeket a felhasználóknak. A mátrix mérete 18.000×480.000 , az ismert értékek pedig körülbelül ennek 0.011%-a volt.

A matematika nyelvén a feladatot egy rang minimalizálási problémaként szokás felfogni. A rang minimalizálása szemléletesen azt jelenti, hogy a hasonló ízléssel rendelkező felhasználók hasonló termékeket fognak kedvelni. Jelölje X a felhasználók és termékek által meghatározott mátrixot és y_{ij} az ismert elemeket:

$$\min_X \text{rank}(X), \quad \text{feltéve hogy: } x_{ij} = y_{ij} \quad (i, j) \in \Omega.$$

Mivel a rang minimalizálási probléma NP-nehéz, az eredeti feladat helyett egy relaxált változat megoldását szokás keresni. Ehhez vezessünk be néhány fogalmat.

6.1.1. Definíció. Egy $X \in \mathbb{R}^{n \times m}$ mátrix nyom-normája

$$\|X\|_* = \text{tr}\left(\sqrt{X^\top X}\right) = \sum_{i=1}^{\min\{m,n\}} \lambda_i,$$

ahol λ_i a mátrix sajátértékei.

6.1.2. Definíció. Egy $X \in \mathbb{R}^{n \times m}$ mátrix Frobenius-normája

$$\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |x_{ij}|^2} = \sqrt{\text{tr}(X^\top X)} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \lambda_i^2},$$

ahol λ_i a mátrix sajátértékei.

Tekintsük az alábbi feladatot

$$\min_X \|X\|_* + \frac{\gamma}{2} \|\Omega_m \circ (X - Y)\|_F^2.$$

Megmutatható, hogy ez az optimalizálási feladat konvex, ahol elegendő bemeneti érték mellett az optimum közel van az eredeti feladat megoldásához. $\Omega_m \circ (X - Y)$ kifejezés azt ellenőrzi, hogy az eredetileg adott mátrixelemek minnél kevésbé változzanak meg, γ választott konstans. Láthatóan, ez a modell nem használja a felhasználók és termékek közötti kapcsolati gráfokat. A 2.3.11-ös példához hasonlóan értelmezhető egy mennyiség, ami a termékek és felhasználók kapcsolatát hivatott szabályozni.

6.1.3. Definíció. Egy $X \in \mathbb{R}^{n \times m}$ mátrix Dirichlet-normája egy $\mathcal{G} = (V, E, W)$ gráfra nézve, ahol $|V| = n$,

$$\|X\|_{\mathcal{G}} = \text{tr}(X^\top \mathcal{L}_n X).$$

Ezáltal a feladat egy másik megközelítése a következő

$$\min_X \|X\|_{\mathcal{G}_u}^2 + \|X\|_{\mathcal{G}_p}^2 + \frac{\gamma}{2} \|\Omega \circ (X - Y)\|_F^2.$$

Tehát a $\|\cdot\|_{\mathcal{G}}$ kifejezés azért felelős, hogy a hasonló érdeklődésű felhasználóknak hasonló termékeket ajánljon. Ez azt jelenti, \mathcal{G}_u gráfon egy adott terméket a csúcsokon értelmezett függvényként felfogva $y_j \in \mathbb{R}^{|V_u|}$ megmutatja, hogy a felhasználóknak mennyire ajánljuk a j . terméket. Természetes elvárás, hogy ez a függvény sima legyen, azaz a közeli csúcsokon

hasonló értékeket vegyen fel. Hasonlóan \mathcal{G}_p gráfon egy adott felhasználót a csúcsokon értelmezett függvényként felfogva $z_i \in \mathbb{R}^{|V_p|}$ megmutatja, hogy az i . felhasználónak mennyire ajánljuk az egyes termékeket. Ugyancsak természetes elvárás, hogy a hasonló termékeket közeli értékekkel látjuk el, azaz a függvény sima legyen.

Multi-Gráf Konvolúció

Kiterjesztjük az 2.3.1 definíciót több gráf esetére. Ha adott \mathcal{G}_u és \mathcal{G}_p , akkor $X \in \mathbb{R}^{|V_u| \times |V_p|}$ mátrix 2 dimenziós gráf Fourier transzformáltja az 1 dimenziós formulák egymás utáni alkalmazása $\widehat{X} = \Phi_p^\top (\Phi_u^\top X)^\top = \Phi_u^\top X \Phi_p$. Ebből adódik a konvolúcióra vonatkozó formula

$$X * Y = \Phi_p(\widehat{X} \circ \widehat{Y})\Phi_u^\top = \Phi_u[\widehat{g}(\Lambda_u)\Phi_u^\top X \Phi_p \widehat{g}(\Lambda_p)]\Phi_p^\top = \widehat{g}(\mathcal{L}_{u,n})X\widehat{g}(\mathcal{L}_{p,n}).$$

Az \widehat{Y} mátrix tartalmazza a tanulandó filtereket. $\widehat{Y} = \widehat{g}(\Lambda_u, \Lambda_p)$. Összevonva a tényezőket a konvolúció kifejezhető a két Laplace-mátrix függvényeként. \widehat{g} kiszámítására használjunk Csebisev-polinomokat, ekkor

$$\widehat{g}_\Theta(\bar{\lambda}_a, \bar{\lambda}_b) = \sum_{j,j'=1}^r \theta_{j,j'} T_j(\bar{\lambda}_a) T_{j'}(\bar{\lambda}_b)$$

a filterhez tartozó paramétereket Θ -val jelöljük. Az egy dimenziós esethez hasonlóan lehetséges több különböző konvolúciót is végrehajtani egy adott bementeten. Ezt hivatott formalizálni a következő definíció.

6.1.4. Definíció (MGC(q, q')). *Legyenek $\mathcal{G}_1 = (V_1, E_1, W_1)$ és $\mathcal{G}_2 = (V_2, E_2, W_2)$ gráfok. $X \in \mathbb{R}^{|V_1| \times |V_2| \times q}$ jelöli az input tenzort, $\widehat{X} \in \mathbb{R}^{|V_1| \times |V_2| \times q'}$ jelöli az output tenzort. $\widehat{g}(\mathcal{L}_{u,n})_{i,j}$ és $\widehat{g}(\mathcal{L}_{p,n})_{i,j}$ tartalmazzák a tanulandó filtereket. $\xi : \mathbb{R} \rightarrow \mathbb{R}$ aktivációs függvényt elemenként alkalmazzuk. Ekkor Multi-Gráf Konvolúciónak nevezzük a következő réteget :*

$$\widehat{X}_j = \xi \left(\sum_{i=1}^q \widehat{g}(\mathcal{L}_{u,n})_{i,j} X_i \widehat{g}(\mathcal{L}_{p,n})_{i,j} \right), \quad j \in \{1, \dots, q'\}.$$

A háló betanítása az

$$L(\bar{X}_\Theta) = \|\bar{X}_\Theta\|_{\mathcal{G}_u}^2 + \|\bar{X}_\Theta\|_{\mathcal{G}_p}^2 + \frac{\gamma}{2} \|\Omega \circ (\bar{X}_\Theta - Y)\|_F^2.$$

hibafüggvény minimalizálásával történik. Egy előre rögzített T hiperparaméter jelöli az iterációk számát. Egy iterációban kiszámoljuk az \bar{X} mátrixot. Majd a hibafüggvény gradiensének segítségével frissítjük a háléhoz tartozó Θ súlyokat. A gradiens kiszámítása a

backpropagation algoritmussal történik, ennek részletes tárgyalásától eltekintünk. A méresek azt mutatják, hogy a rétegek számának növelésével a hibafüggvény értéke tovább csökkenthető, viszont a paraméterek száma jelentősen megnő. Ennek kiküszöbölésére használhatjuk, az úgynevezett LSTM réteget. Intuitíven a réteg egy memóriaegység, amelynek köszönhetően egy darab konvolúciós réteg használatával hasonló eredményt érünk el mint ha többet használnánk. A tapasztalat azt mutatja, hogy az LSTM-et használó háló képes kezelni az adatok időbeli függését.

6.1.5. Definíció (Long-Short Term Memory). *A t . időpillanatban $x^{(t)} \in \mathbb{R}^n$ jelöli az input vektort, $h^{(t-1)} \in \mathbb{R}^n$ a memóriát. $W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c \in \mathbb{R}^{n \times n}$ súlymátrixok, $b_f, b_i, b_o, b_c, b_{out} \in \mathbb{R}^n$ vektorok és $W_{out} \in \mathbb{R}^{n \times k}$, $b_{out} \in \mathbb{R}^k$ tartalmazzák a tanulandó paramétereket. $\sigma, \tanh : \mathbb{R} \rightarrow \mathbb{R}$ aktivációs függvényeket elemenként alkalmazzuk. $x_{out} \in \mathbb{R}^k$. A következő architektúrát nevezzük LSTM-blokknak:*

$$\begin{aligned}
 f^{(t)} &= \sigma(x^{(t)}W_f + h^{(t-1)}U_f + b_f) \\
 i^{(t)} &= \sigma(x^{(t)}W_i + h^{(t-1)}U_i + b_i) \\
 o^{(t)} &= \sigma(x^{(t)}W_o + h^{(t-1)}U_o + b_o) \\
 \tilde{c}^{(t)} &= \sigma(x^{(t)}W_c + h^{(t-1)}U_c + b_c) \\
 c^{(t)} &= f_t \cdot c^{(t-1)} + i^{(t)} \cdot \tilde{c}^{(t)} \\
 h^{(t)} &= o^{(t)} \cdot \sigma(c^{(t)}) \\
 x_{out} &= \tanh(o^{(t)}W_{out} + b_{out})
 \end{aligned}$$

Az $f^{(t)}$ "forget gate"-nek, $i^{(t)}$ "input gate"-nek, $o^{(t)}$ "output gate"-nek szokás nevezni.

Algorithm 2: Recurrent Multi-Graph Convolutional Neural Network(RMGCC)

Input : $X \in \mathbb{R}^{|V_u| \times |V_p|}$, T : iterációk száma, R : rekurzió iterációjának száma,
 q : MGC outputjainak száma, Θ : MGC paraméterei,
 σ : LSTM paraméterei

Output: $X^{(T)} \in \mathbb{R}^{|V_u| \times |V_p|}$

```
1 for  $t = 0$  to  $T$  do
2    $X^{(0)} \leftarrow X$ ;
3   for  $r = 0$  to  $R$  do
4      $\widehat{X}^{(r)} \leftarrow \text{MGC}_{\Theta}(1, q)(X^{(r)})$   $(\widehat{X}^{(r)} \in \mathbb{R}^{|V_u| \times |V_p| \times q})$ ;
5     for  $(i, j)$  do
6        $dx_{ij}^{(r)} \leftarrow \text{LSTM}_{\sigma}([\widehat{x}_{ij,1}^{(r)}, \dots, \widehat{x}_{ij,q}^{(r)}])$ 
7     end
8      $X^{(r+1)} \leftarrow X^{(r)} + dX^{(r)}$ ;
9   end
10   $X \leftarrow X^{(R)}$ ;
11   $\nabla L \leftarrow \text{BACKPROPAGATE}(X, \Theta)$ ;
12   $\Theta \leftarrow \Theta - \epsilon_k \nabla L$ ;
13 end
```

Faktorizált modell

Egy konvolúciós réteg paramétereinek számát tovább csökkenthetjük $\mathcal{O}(|V_u| + |V_p|)$ értékre, ha a mátrixot $X = UP^{\top}$ szorzat alakban keressük, ahol $U \in \mathbb{R}^{|V_u| \times r}$ és $P \in \mathbb{R}^{r \times |V_p|}$ és $r < \min(|V_u|, |V_p|)$. Az P mátrix oszlopai reprezentálják a termékeket, az U mátrix sorai reprezentálják a felhasználókat. A két mátrixra külön-külön elvégezve a megfelelő 1 dimenziós konvolúciót egyszerűsödik a modell ($i \in \{1, \dots, |V_u|\}, j \in \{1, \dots, |V_p|\}$):

$$\widehat{u}_i = \xi(\widehat{g}(\mathcal{L}_{u,n})u_i), \quad \widehat{p}_j = \xi(\widehat{g}(\mathcal{L}_{p,n})p_j).$$

A fenti algoritmusokban az X mátrix helyett minden lépésben külön-külön frissítjük U -t és P -et. A célfüggvény a következőképpen módosul

$$\min_{U, P} \|U\|_{\mathcal{G}_u}^2 + \|P\|_{\mathcal{G}_p}^2 + \frac{\gamma}{2} \|\Omega \circ (UP^{\top} - Y)\|_F^2.$$

6.2. Maximális független csúcshalmaz keresése

6.2.1. Definíció. Adott $\mathcal{G} = (V, E)$ gráf csúcsainak egy $W \subset V$ részhalmazát függetlennek nevezzük, ha W -beli csúcsok nem szomszédosak, azaz minden $v_i, v_j \in W$ -re $(v_i, v_j) \notin E$. Egy csúcshalmaz maximális független (maximal independent set, MIS), ha nem létezik nála nagyobb számosságú független csúcshalmaz.

Megmutatható, hogy a maximális független csúcshalmaz probléma NP-teljes, tehát nem remélhetünk polinomiális algoritmust a feladat megoldására. Azonban neurális hálózatok és klasszikus gráf redukciós módszerek segítségével közel lehet kerülni az optimumhoz. A feladat egy $\mathcal{G} = (V, E)$, $|V| = N$ csúcsú gráfon maximális független csúcshalmaz keresése. Vegyünk egy L rétegű GCN hálót, aminek egy rétegét a következő számolási szabály írja le $H^{(l+1)} = \sigma \left(\widehat{D}^{-\frac{1}{2}} \widehat{W} \widehat{D}^{-\frac{1}{2}} H^{(l)} \Theta^{(l)} \right)$. Az utolsó réteg outputja legyen egy N hosszú vektor, az aktivációs függvény pedig a szigmoid függvény $\sigma(x) = \frac{1}{1+e^{-x}}$, melynek értékkészlete $(0, 1)$ intervallum. A gráfot $x = \mathbf{1}$ vektorral inicializáljuk. A háló megad nekünk egy $f(\mathcal{G}, \Theta)$ heurisztikus függvényt, amelyre úgy gondolunk, hogy minden csúcshoz hozzárendeli, hogy mekkora valószínűséggel esik bele egy maximális független csúcshalmazba. A háló tanítását egy $D = \{(\mathcal{G}_i, y_i)\}_{i=1}^K$ halmazon végezzük el, ahol $y_i \in \{0, 1\}^N$ kódol egy lehetséges optimális csúcshalmazt a \mathcal{G}_i gráfra. A tanításhoz a kereszt-entrópia veszteségfüggvényt használjuk

$$\ell(f(\mathcal{G}_i, \Theta), y_i) = \sum_{j=1}^N [y_i(j) \log(f(\mathcal{G}_i, \Theta)(j)) + (1 - y_i(j)) \log(1 - f(\mathcal{G}_i, \Theta)(j))],$$

ahol $y_i(j)$ az i . gráf j . csúcsa és hasonlóan $f(\mathcal{G}_i, \Theta)(j)$ a j . csúcsra kapott heurisztikus érték. Ha az f függvénynek egy új ismeretlen gráfot adunk, akkor az outputból egy mohó algoritlussal nyerhetünk megoldást. Nagyság szerint sorbarendezzük $f(\mathcal{G}_i, \Theta)$ értékeit és a legnagyobb értékkel rendelkező csúcst beválasztjuk a független halmazba, azaz a címkéjét 1-re cseréljük, majd minden szomszédját 0-ra állítjuk. Ezt az eljárást ismételjük, amíg egy olyan csúcst választunk ki f alapján, aminek már 0 az értéke. Kiszedve a gráfból a már megcímkézett csúcsokat, kapunk egy kisebb \mathcal{G}' gráfot, erre úra használva a heurisztikus függvényt haladhatunk tovább.

A háló utolsó rétegének channel számát megnövelve $H^{(L)} \in \mathbb{R}^{N \times M}$ output mátrixot kapjuk, amire gondolhatunk úgy, mint M darab különböző heurisztika. A tanításhoz ez esetben

használjuk a

$$J(D, \Theta) = \sum_i^K \min_m \ell(f(\mathcal{G}_i, \Theta), y_i).$$

veszteségfüggvényt, ahol ℓ a kereszt-entrópia. A veszteségfüggvénybe az M darab heurisztika közül, csak a legjobb eredményt adó számít bele.

Algorithm 3: Maximális független csúcshalmaz keresése GCN heurisztikával

Input : \mathcal{G} gráf, T : futási idő

Output: $y_i \in \{0, 1\}^N$ címkézés

```

1  $U = [\mathcal{G}]$ 
2 for  $t = 0$  to  $T$  do
3    $\mathcal{G}' \leftarrow U[\text{random}]$ 
4   for  $m = 0$  to  $M$  do
5      $V \leftarrow \text{argsort}(f^m(\mathcal{G}', \Theta))$ 
6     for  $i = 1$  to  $|V|$  do
7       if  $V[i] = 0$  then break for;
8        $V[i] \leftarrow 1$ 
9        $N(V[i]) \leftarrow 0$ 
10    end
11    if  $\mathcal{G}'$  teljesen címkézett then
12      Legjobb megoldás frissítése
13    else
14       $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \{\text{címkézett csúcsok}\}$ 
15    end
16     $U.\text{append}(\mathcal{G}')$ 
17  end
18 end

```

Ennek az eljárásnak az az előnye, hogy egyszerre több részmegoldást is fent tudunk tartani. Egy adott \mathcal{G} gráfhoz a heurisztikus függvény megad M darab részmegoldást. Ezeket egy listában eltároljuk, majd bővítjük őket használva a heurisztikus függvényeket. Így egy szélességi kereséshez hasonló módon nyerünk különböző megoldásokat, amelyek közül a legjobbat tartjuk meg. Így egy idő után exponenciálisan sok részmegoldásunk lesz, az algoritmust addig futtatjuk, amíg van rá számítási kapacitásunk.

Tanító adathalmaz

Ahhoz hogy megkapjuk a háló megfelelő paramétereit, szükségünk van egy tanító adathalmzra. Ennek egy módja Boole-függvények segítségével hozunk létre gráfokat. Egy Boole-függvényről azt mondjuk, hogy konjunktív normálformában van megadva, ha

$$B(x) = \bigwedge_i \bigvee_j (\neg)^{k_{ij}} x_{ij}$$

ahol, $x_{ij} \in \{0, 1\}$ és $k_{ij} \in \{0, 1\}$, mondja meg, hogy x_{ij} -t tagadjuk, vagy nem. A változókat vagy negáltjait közös néven literálnak hívjuk, a \vee (AND) jellel összekötött literálokat pedig klóznak nevezzük. Egy Boole-függvényt kielégíthetőnek nevezünk, ha létezik olyan $x \in \{0, 1\}^K$, amire $B(x) = 1$. Cook tétele szerint B kielégíthetőségének eldöntése NP-teljes probléma.

6.2.2. Állítás. *Egy L klózból álló B Boole-függvényhez létezik \mathcal{G} gráf, amire teljesül, hogy pontosan akkor létezik benne L méretű független csúcshalmaz, ha B kielégíthető.*

Bizonyítás. Minden literálhoz rendelünk egy csúcsot. Egy klózon belüli literálhoz tartozó csúcsokat kössük össze éllel. Klózon kívül két csúcsot pontosan akkor kössünk össze, ha a hozzájuk ugyanaz a változó tartozik, és egyik a másiknak tagadása. Ha ebben a gráfban találunk egy L méretű független halmazt, akkor a megfelelő Boole-függvény is kielégíthető. Mivel minden klózhoz tartozó klikkben pontosan egy csúcsot választhatunk ki, L méretű a maximális független halmaz. Visszafelé, ha B kielégíthető, akkor a változók értékei meghatároznak egy $\{0, 1\}$ címkézést a gráfon, ami független csúcsokat határoz meg. \square

6.2.3. Megjegyzés. *Ez előző állításból már következik, hogy a Boole-kielégíthetőség feladat (SAT) polinomiálisan visszavezethető a maximális független csúcshalmaz (MIS) feladatra, azaz MIS is NP-teljes.*

Tehát megfelelő mennyiségű Boole-függvény birtokában (pl. SATLIB könyvtár) létre tudunk hozni tanító halmazt. Az eljárás potenciális hátránya, hogy a tanító halmaz csak egy bizonyos típusú gráfcsaládot tartalmaz.

2-improvement

A kapott eredményt tovább javíthatjuk, klasszikus algoritmusok bevetésével. Egy ilyen eljárás az úgynevezett 2-javítás, mely megpróbál kicserélni egy darab 1-es csúcsot két darabra. Menjünk végig az 1-es címkével ellátott csúcsokon. Ha egy ilyen v csúcsnak létezik

v_i, v_j 0 cimkéjű szomszédja, amire $(v_i, v_j) \notin E$ és $N(v_i)$ és $N(v_j)$ halmazokban pontosan egy-egy csúcs cimkéje 1-es (nevezetesen v -nek), akkor nagyobb független halmazhoz jutunk, ha v_i és v_j cimkéjét 1-esre cseréljük, és v cimkéjét 0-ra. Ez az algoritmus $\mathcal{O}(E)$ idő alatt javítja az eredményt.

Gráfredukció

A cimkézés előtt érdemes redukálni a gráfot néhány egyszerű megfigyelés alapján. Célunk az, hogy egy kisebb, de bizonyos értelemben ekvivalens gráfon végezzük el a független halmaz keresését.

- (1) Ha $\deg(v) = 1$, akkor létezik olyan MIS, aminek v része, ezért v -t és szomszédját elhagyhatjuk a gráfból.
- (2) Ha $N(v) = \{u, w\}$ és $(u, w) \notin E$, akkor vagy $v \in \text{MIS}$ vagy $u, w \in \text{MIS}$, tehát $\{u, v, w\}$ csúcshalmaz helyettesíthető egy darab z csúccsal, z -ből éleket húzunk be $N(u)(w)$ csúcsokba a módosított gráfban. Ha $z \in \text{MIS}$, akkor $\{u, w\} \in \text{MIS}$, ha $z \notin \text{MIS}$, akkor $v \in \text{MIS}$.
- (3) Legyen $\deg(u) = \deg(v) = 3$ és $N(u) = N(v)$. Ha $\mathcal{G}(N(u))$ részgráfban van él, akkor $u, v \in \text{MIS}$. Ha nincs él, akkor helyettesítsük $\{u, v, N(u)\}$ csúcsokat z csúccsal és kössük össze $N(u)$ szomszédjaival. Ha $z \in \text{MIS}$, akkor $N(z) \notin \text{MIS}$, tehát $N(u) \in \text{MIS}$. Ha pedig $z \notin \text{MIS}$, akkor $u, v \in \text{MIS}$.

Hivatkozások

- [1] Eigenvalues and the laplacian of a graph. <http://web.archive.org/web/20210318195022/http://www.math.ucsd.edu/~fan/research/cb/ch1.pdf>.
- [2] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.
- [3] Robert M Freund. Symmetric matrices and eigendecomposition, 2014.
- [4] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, USA, 2nd edition, 2012.
- [5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.
- [6] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [7] George M Phillips. *Interpolation and approximation by polynomials*, volume 14. Springer Science & Business Media, 2003.
- [8] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.
- [9] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [10] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018.

- [11] E. Sonnendrücker A. Ratnani. B-splines and isogeometric analysis. https://web.archive.org/web/20210427133200/https://www-m16.ma.tum.de/foswiki/pub/M16/Allgemeines/AdvFE16/bsplines_Ahmed.pdf.
- [12] Angela Kunoth, Tom Lyche, Giancarlo Sangalli, and Stefano Serra-Capizzano. *Splines and PDEs: From approximation theory to numerical linear algebra*. Springer, 2018.
- [13] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, USA, 2004.
- [14] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.
- [15] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.
- [16] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [17] Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. Grakel: A graph kernel library in python. *Journal of Machine Learning Research*, 21(54):1–5, 2020.
- [18] Federico Monti, Michael M Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. *arXiv preprint arXiv:1704.06803*, 2017.
- [19] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. *arXiv preprint arXiv:1810.10659*, 2018.