Modellering en simulatie
Practicum 1: Lagerangbenaderingen

Wim Michiels en Nick Vannieuwenhoven

22 oktober 2015

1 Praktische informatie

Het practicum wordt individueel opgelost. Het verslag dient de volgende componenten te bevatten: de oplossingen van de opgaves en de broncode van je programma's. Het verslag wordt uiterlijk **vrijdag 27 november 2015** ingediend. Een papieren versie van het verslag dient zich ten laatste maandagvoormiddag 30 november 2015 om 12u00 in de studentenbrievenbus in gebouw 200A te bevinden. Deze uitdraai dient tevens de broncode van de programma's te bevatten. Een elektronische versie van het verslag en een gecomprimeerde map die de Matlabbronbestanden en het verslag bevat, dien je op te sturen per elektronische post naar `nick.vannieuwenhoven@cs.kuleuven.be`. Zorg ervoor dat je bronbestanden de **opgelegde naamgeving** respecteren (zie verder) want de codes zullen automatisch geverifieerd worden. De naam van de gecomprimeerde map (de bestandsextensie buiten beschouwing latende) dient overeen te stemmen met je studentennummer; "s0123456.zip," "s0123456.tar.gz" en dergelijke zijn allen aanvaardbaar.

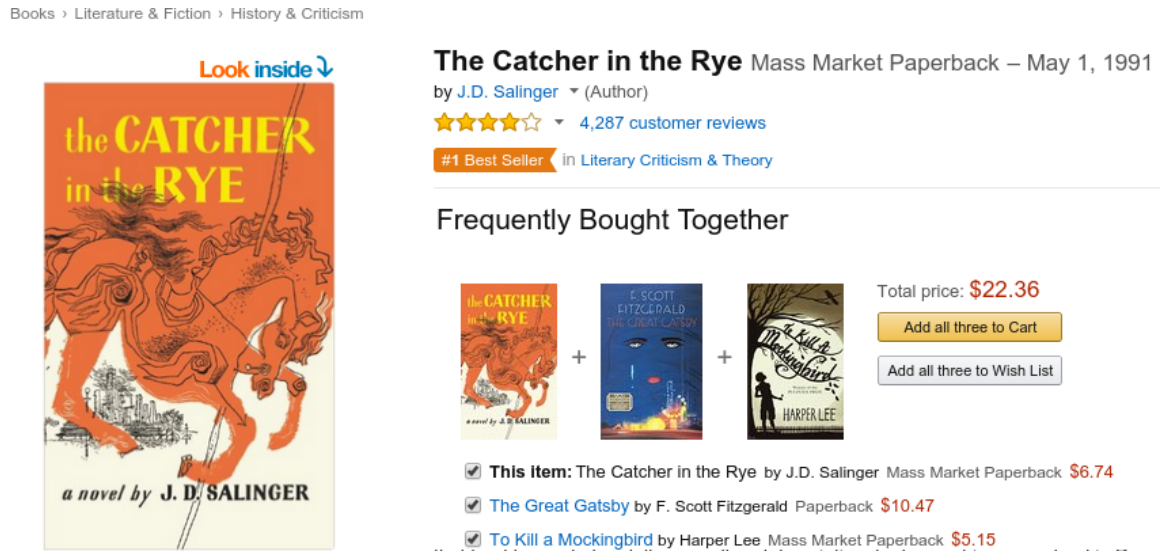
Veel succes!

> **In wat volgt vervang je elk voorkomen van "StudNum" met jouw studentennummer (r- of s-nummer)—bijvoorbeeld "s0123456". Elk van de functies dien je in een afzonderlijk bestand met dezelfde naam als de functie (met extensie ".m") te implementeren. Je codes zullen automatisch geverifieerd worden. Indien de opgelegde naamgeving niet gerespecteerd wordt, dan wordt dit beschouwd als het niet oplossen van de betrokken vraag.**
>
> Voor het oplossen van de opgaven mag je **alle** ingebouwde Matlabfuncties gebruiken. Naast de verplichte functies die de opgelegde naamgeving dienen te respecteren, mag je ook nog zelf extra hulpfuncties implementeren indien je dit nodig acht. Zorg ervoor dat deze hulproutines, voor zover ze in een eigen Matlabbronbestand geïmplementeerd worden, een naam dragen die start met "StudNum_".
>
> Het geniet de aanbeveling om de opgave eerst volledig te lezen alvorens met de opdrachten te starten.

2 Een aanbevelingssysteem voor Amazon.com

Voor het aanbevelen van producten aan gebruikers maakt Amazon.com, Inc. gebruik van een zogenaamd aanbevelingssysteem. Dit systeem zal aan de bezoekers die een specifiek product raadplegen aanbevelingen maken op basis van de aankopen die de andere bezoekers pleegden die dit product kochten. In Figuur 1 wordt hiervan een voorbeeld gegeven; in dit geval zal Amazon.com de boeken *The Great Gatsby* en *To Kill a Mockingbird* aanbevelen wanneer een bezoeker de webpagina van het boek *The Catcher in the Rye* raadpleegt. De aanbevelingen die Amazon.com in de rubriek *Frequently Bought Together* maakt, zijn niet gepersonaliseerd: ze houden geen rekening met de producten die je zelf eerder hebt gekocht als anonieme of ingelogde gebruiker.

In deze opgave zullen we een nieuw aanbevelingssysteem ontwikkelen voor Amazon.com op basis van de gebruikersbeoordelingen van de aangeboden producten. Het algoritme steunt enkel op eenvoudige principes uit de lineaire algebra en statistiek. De prestaties van het algoritme zullen beoordeeld worden op



Figuur 1: Een voorbeeld van een aanbeveling door Amazon.com.

een zeer kleine deelverzameling van de *Amazon review data* database die door Prof. J. McAuley verzameld werd.[1] Deze database bevat ongeveer 147 miljoen beoordelingen van geregistreerde gebruikers voor allerhande producten (boeken, CDs, videogames, computeraccessoires, etc) die op Amazon.com worden aangeboden. Teneinde de resultaten eenvoudig te kunnen interpreteren, zullen we ons in dit practicum beperken tot slechts 110 boeken en 174 gebruikers die tezamen 3154 beoordelingen produceerden. De boeken werden geselecteerd uit de 1000 meest beoordeelde boeken die in juli 2014 door Amazon.com verkocht werden.

Het basisidee van het aanbevelingssysteem voor Amazon.com dat we ontwikkelen in dit practicum bestaat eruit om de beoordeling van boek i te modelleren als een discrete stochastische variabele B_i die de waarden $\{1, 2, 3, 4, 5\}$ kan aannemen.[2] Een maat voor de statistische afhankelijkheid tussen de twee stochastische variabelen B_i en B_j is de zogenaamde *Pearson correlatiecoefficiënt*. Deze correlatiecoefficiënt is een reëel getal tussen -1 en 1. Een waarde dicht bij 1 geeft aan dat de stochastische variabelen B_i en B_j sterk positief verbonden zijn, een waarde dicht bij -1 duidt op een sterk negatief verband en een waarde rond 0 impliceert dat er geen bijzonder lineair verband bestaat tussen de beoordelingen van boek i en j. In dit practicum zullen we op zoek gaan naar clusters van boeken waarvoor de beoordelingen onderling sterk positief gecorreleerd zijn.

2.1 Matrixvervolledigiing

De vereiste Pearson correlatiecoefficiënten zullen we moeten schatten op basis van een steekproef, aangezien we geen veronderstellingen zullen maken over de eigenschappen van de stochastische variabelen B_i. De steekproef die we in dit practicum gebruiken, bestaat uit de voornoemde 3154 beoordelingen afkomstig van 174 gebruikers. De beoordeling van een gebruiker voor een boek kan op natuurlijke wijze voorgesteld worden door een matrix. Laat ons een beknopt voorbeeld uitwerken. Veronderstel dat onze catalogus bestaat uit de boeken *Catch-22*, *Cloud Atlas*, *Dune*, *I Am Legend* en *The Road*. De beoordelingen—waarbij 1 de laagst mogelijke en 5 de hoogst mogelijke beoordeling is—van 15 willekeurige gebruikers zou men dan als volgt kunnen voorstellen:

$$R^T = \begin{array}{r} \text{Catch-22} \\ \text{Cloud Atlas} \\ \text{Dune} \\ \text{I Am Legend} \\ \text{The Road} \end{array} \begin{bmatrix} ? & ? & \mathbf{3} & \mathbf{4} & ? & ? & ? & \mathbf{5} & ? & ? & ? & \mathbf{5} & \mathbf{5} & \mathbf{1} & ? \\ ? & \mathbf{4} & ? & ? & ? & ? & \mathbf{5} & \mathbf{5} & ? & \mathbf{3} & \mathbf{3} & \mathbf{5} & ? & ? & ? \\ ? & ? & \mathbf{3} & ? & \mathbf{4} & ? & ? & ? & \mathbf{5} & \mathbf{4} & \mathbf{4} & ? & ? & ? & ? \\ \mathbf{2} & \mathbf{4} & ? & ? & \mathbf{1} & \mathbf{2} & \mathbf{1} & \mathbf{3} & \mathbf{3} & \mathbf{2} & ? & ? & \mathbf{3} & ? & \mathbf{2} \\ \mathbf{4} & \mathbf{5} & ? & \mathbf{2} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & ? & ? & ? & ? & \mathbf{5} & \mathbf{5} & \mathbf{3} \end{bmatrix}. \quad (1)$$

Elke rij van R geeft de beoordeling van één gebruiker voor enkele van de boeken en elke kolom van R geeft de beoordelingen van enkele gebruikers voor één boek. De meeste waarden in bovenstaande matrix

[1] Zie J. McAuley, R. Pandey, J. Leskovec, *Inferring networks of substitutable and complementary products*, Knowledge Discovery and Data Mining, 2015.

[2] Amazon.com hanteert een beoordelingssysteem waarbij de gebruiker een score kan geven van één tot vijf sterren.

zijn onbekend, zoals wordt aangegeven met een "?".[3] Men noemt R een onvolledige matrix.

De statistische methoden die we in de volgende sectie zullen toepassen, kunnen niet overweg met een onvolledige steekproef. Een typische aanpak in de statistiek bestaat eruit om deze ontbrekende waarden in R te vervangen door (zinvolle) numerieke waarden. Dit proces noemt men "*imputation*" in de statistische literatuur. Duid het aantal gebruikers in de steekproef aan met m en het aantal boeken uit de catalogus met n. We zullen in dit project een zogenaamd matrixvervolledigingsprobleem oplossen om de onbekende waarden te voorspellen in de matrix $R \in \mathbb{R}^{m\times n}$. Hiervoor veronderstellen we dat de volle gebruiker-beoordelingenmatrix R een kleine rang $r \ll \min\{m, n\}$ heeft: de beoordeling $r_{i,j}$ van gebruiker i voor boek j voldoet—bij benadering—aan de relatie

$$r_{i,j} \approx w_{i,1} \cdot f_{j,1} + w_{i,2} \cdot f_{j,2} + \cdots + w_{i,r} \cdot f_{j,r},$$

hetgeen equivalent is met

$$R \approx WF^T = \mathbf{w}_1\mathbf{f}_1^T + \mathbf{w}_2\mathbf{f}_2^T + \cdots + \mathbf{w}_r\mathbf{f}_r^T \tag{2}$$

en waarbij $\mathbf{w}_i \in \mathbb{R}^m$, respectievelijk $\mathbf{f}_i \in \mathbb{R}^n$, de i^{de} kolom is van de matrix $W \in \mathbb{R}^{m\times r}$, respectievelijk $F \in \mathbb{R}^{n\times r}$. We kunnen model (2) als volgt interpreteren. Elke vector $\mathbf{f}_i$ kunnen we interpreteren als een gewicht voor een bepaalde i^{de} feature of kenmerk van het boek, terwijl $\mathbf{w}_i$ een voorliefde voorstelt van de gebruikers voor het i^{de} kenmerk. De featurevectoren $\mathbf{f}_i$ bevatten dus enkel informatie over de boeken, terwijl de voorliefdevectoren $\mathbf{w}_i$ enkel informatie bevatten over de gebruikers. Vanwege deze eigenschap spreekt men ook wel eens van een "gescheiden voorstelling." De veronderstelling dat R goed benaderd kan worden door een matrix van rang r impliceert dat de beoordeling van de meeste gebruikers grotendeels verklaard kan worden door r kenmerken van een boek in combinatie met de voorliefde van de gebruikers voor de betrokken kenmerken. Zo zouden er bijvoorbeeld featurevectoren $\mathbf{f}_i$ kunnen zijn die een numerieke score toekennen aan de hoeveelheid actie, drama, humor, horror, intrige, romantiek en spanning voor elk van de boeken. Het is duidelijk dat er een objectief verschil bestaan in de hoeveelheid actie in *To Kill a Mockingbird* en *The Hunt for Red October*. Een featurevector die de hoeveelheid actie in elk boek voorstelt, genoteerd als $\mathbf{f}_1$, zou dus zeker een grotere numerieke waarde bevatten voor het laatst vernoemde boek dan voor het eerste. Het is eveneens duidelijk dat de gebruikers een verschillende voorliefde voor actie kunnen bezitten. Deze verschillen kunnen door de voorliefdevector $\mathbf{w}_1$ voorgesteld worden; een positieve numerieke waarde duidt op een voorliefde voor actie, een negatieve voor een afkeer en een waarde dicht bij 0 geeft indifferentie met betrekking tot actie aan. De rang-1 matrix $\mathbf{w}_1\mathbf{f}_1^T$ bevat dan op positie (i, j) de bijdrage van het kenmerk "actie" aan de totaalscore van gebruiker i voor boek j.

Om de onbekende waarden in de matrix R te voorspellen, veronderstellen we dus dat R een lagerangontbinding $R = WF^T$ heeft zoals in vergelijking (2). Deze lagerangontbinding kunnen we op de volgende manier berekenen. Het welbekende Eckart–Young theorema stelt dat de optimale benadering van een matrix $X \in \mathbb{R}^{m\times n}$ met $m \geq n$ van rang $r \leq n$ gegeven wordt door de *afgeknotte singuliere waardenontbinding*. Laat de singuliere waardenontbinding van X gegeven zijn door

$$X = U\begin{bmatrix}\Sigma\\ 0\end{bmatrix}V^T = \begin{bmatrix}U_1 & U_2\end{bmatrix}\begin{bmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T\\ V_2^T\end{bmatrix}$$

waarbij $U \in \mathbb{R}^{m\times m}$, $\Sigma \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{n\times n}$, $U_1 \in \mathbb{R}^{m\times r}$, $\Sigma_1 \in \mathbb{R}^{r\times r}$ en $V_1 \in \mathbb{R}^{n\times r}$; de overige matrices hebben dimensies die compatibel zijn met de partitionering in bovenstaande vergelijking. De matrices U en V zijn orthogonale matrices en Σ is een diagonaalmatrix met positieve getallen op de diagonaal die aflopend gesorteerd zijn. Het Eckart–Young theorema stelt dan formeel dat

$$\min_{\text{rank}(B)\leq r} \|X - B\|_F = \|X - U_1\Sigma_1V_1^T\|_F.$$

Helaas kunnen we niet zonder meer een afgeknotte singuliere waardenontbinding berekenen van de onvolledige matrix R om tot een (optimale) benadering zoals in vergelijking (2) te komen. Om dit probleem op te lossen, zullen we een iteratief algoritme gebruiken. In de eerste stap vervangen we alle onbekende waarden in R door een constante waarde, bijvoorbeeld 1.5. Noem de (volle) matrix die je aldus bekomt R_0. Aangezien R_0 een volle matrix is, kunnen we een afgeknotte singuliere waardenontbinding berekenen om zo de beste rang-r benadering van R_0 te vinden. Laat ons deze optimale rang-r benadering aanduiden met A_1. We kunnen nu de waarden van A_1 gebruiken om een betere voorspelling te maken voor de

[3] Men mag verwachten dat een gemiddelde gebruiker er niet in slaagt om de meer dan 8 miljoen boeken in de volledige catalogus van Amazon.com te lezen.

onbekende waarden in R. Duid met R_1 de matrix aan die je bekomt door de onbekende waarden van R te vervangen door de overeenstemmende waarden in A_1. Vervolgens kan je een nieuwe optimale rang-r benadering berekenen van de volle matrix R_1, enzovoort. In de laatste stap, zeg κ, worden alle waarden in A_κ begrensd tot het interval $[1, 5]$ en de resulterende matrix vormt de gezochte lagerangbenadering. Formeel zouden we het voorgaande algoritme dus kunnen schrijven als volgt. Zij D een binaire matrix waarvoor $d_{i,j} = 0$ als $r_{i,j}$ een onbekende waarde is en $d_{i,j} = 1$ als $r_{i,j}$ een gekende beoordeling is van gebruiker i voor boek j. Laat ons tevens met $C = A \odot B$ het Hadamard- of elementsgewijze product aanduiden: $c_{ij} = a_{ij}b_{ij}$. Met de notatie $1_{m\times n}$ duiden we de $m \times n$ matrix aan van wie alle elementen 1 zijn. We berekenen de lagerangbenadering van de onvolledige beoordelingenmatrix R dan als volgt.

Algorithm 1 Lagerangbenadering van een onvolledige matrix R.

$R_0 \leftarrow D \odot R + 1.5 \cdot (1_{m\times n} - D)$;
for $k \leftarrow 1$ to κ **do**
  $R_{k-1} = U\Sigma V^T$, de singuliere waardenontbinding van R_{k-1};
  $A_k \leftarrow U_1\Sigma_1 V_1^T$, waarbij U_1 de eerste r kolommen van U bevat, Σ_1 de principale $r \times r$ deelmatrix is van Σ en V_1 de eerste r kolommen van V bevat;
  $R_k \leftarrow D \odot R + (1_{m\times n} - D) \odot A_k$;
end for
$A \leftarrow \max\{1, \min\{5, A_\kappa\}\}$;

Wanneer we bovenstaand algoritme toepassen op de matrix R uit vergelijking (1) met $r = 2$ en $\kappa = 1000$, dan vinden we, na afronding tot op één cijfer na de komma, de volgende voorspelling voor de onbekende waarden:

$$A^T = \begin{array}{r} \\ \text{Catch-22} \\ \text{Cloud Atlas} \\ \text{Dune} \\ \text{I Am Legend} \\ \text{The Road} \end{array} \begin{bmatrix} 3.5 & 5.0 & \mathbf{3.0} & \mathbf{4.0} & 2.6 & 3.8 & 2.6 & \mathbf{5.0} & 4.5 & 3.3 & 3.3 & \mathbf{5.0} & \mathbf{5.0} & \mathbf{1.0} & 3.2 \\ 3.8 & \mathbf{4.1} & 1.0 & 1.3 & 5.0 & 4.9 & \mathbf{5.0} & \mathbf{4.8} & 2.7 & \mathbf{3.0} & \mathbf{3.0} & \mathbf{5.0} & 4.6 & 5.0 & 2.7 \\ 4.4 & 5.0 & \mathbf{3.0} & 4.1 & \mathbf{4.0} & 5.0 & 4.0 & 5.0 & \mathbf{5.0} & \mathbf{4.0} & \mathbf{4.0} & 5.0 & 5.0 & 2.6 & 3.8 \\ \mathbf{2.0} & \mathbf{4.0} & 2.2 & 2.9 & \mathbf{1.0} & \mathbf{2.0} & \mathbf{1.0} & \mathbf{2.9} & \mathbf{3.0} & \mathbf{2.0} & 2.0 & 2.9 & \mathbf{3.0} & 1.0 & \mathbf{2.0} \\ \mathbf{4.0} & \mathbf{4.9} & 1.3 & \mathbf{2.0} & \mathbf{5.0} & \mathbf{5.0} & \mathbf{4.9} & \mathbf{5.0} & 3.3 & 3.3 & 3.3 & 5.0 & \mathbf{5.0} & \mathbf{5.0} & \mathbf{3.0} \end{bmatrix}.$$

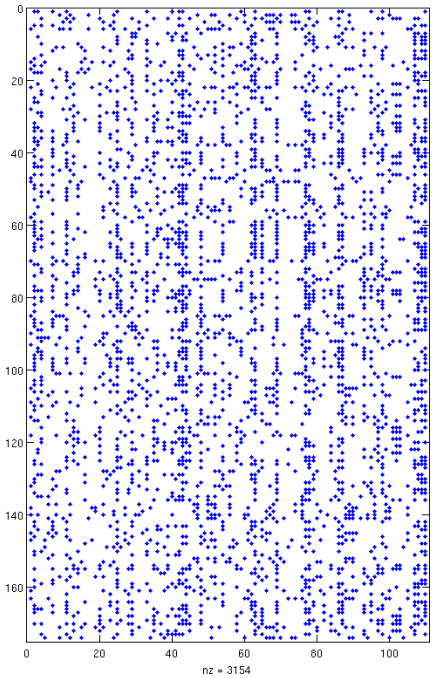
Merk op dat de gekende beoordelingen met Algoritme 1 niet noodzakelijk exact geschat worden. De absolute fout tussen de geschatte matrix A en de originele onvolledige matrix R wordt op natuurlijke wijze gemeten als de som van kwadraten van het verschil tussen de gekende beoordeling en de overeenkomstige geschatte beoordeling. De in het vet weergegeven elementen leiden aldus tot de volgende absolute benaderingsfout:

$$\begin{aligned} e^2 &= (3-3.0)^2 + (4-4.0)^2 + (5-5.0)^2 + (5-5.0)^2 + (5-5.0)^2 + (1-1.0)^2 \\ &\quad + (4-4.1)^2 + (5-5.0)^2 + (5-4.8)^2 + (3-3.0)^2 + (3-3.0)^2 + (5-5.0)^2 \\ &\quad + (3-3.0)^2 + (4-4.0)^2 + (5-5.0)^2 + (4-4.0)^2 + (4-4.0)^2 \\ &\quad + (2-2.0)^2 + (4-4.0)^2 + (1-1.0)^2 + (2-2.0)^2 + (1-1.0)^2 + (3-2.9)^2 + (3-3.0)^2 + (2-2.0)^2 \\ &\qquad + (3-3.0)^2 + (2-2.0)^2 \\ &\quad + (4-4.0)^2 + (5-4.9)^2 + (2-2.0)^2 + (5-5.0)^2 + (5-5.0)^2 + (5-4.9)^2 + (5-5.0)^2 + (5-5.0)^2 \\ &\qquad + (5-5.0)^2 + (3-3.0)^2 \\ &= 0.1^2 + 0.2^2 + 0.1^2 + 0.1^2 + 0.1^2 \\ &= 0.08. \end{aligned}$$

Opdracht 1. Laad het bestand `AmazonBookReviews.mat` in. Dit bestand bevat de volgende elementen: `labels` is een *cell array* die de titels bevat van de 110 boeken in de beperkte catalogus en `Ratings` is een onvolledige 174×110 matrix die de gekende beoordelingen van de 174 gebruikers voor de 110 boeken bevat. De i^{de} kolom van `Ratings` komt overeen met het boek met titel `labels{i}`. De `Rating` matrix bevat slechts 3154 niet-nulwaarden; dit zijn alle gekende beoordelingen. Alle beoordelingen die gelijk zijn aan 0 zijn *onbekende* beoordelingen. Wanneer je het commando `spy(Ratings)` uitvoert in Matlab dan dient Figuur 2 weergegeven te worden. ♦

Opdracht 2. Schrijf een functie met de hoofding

```
function [err] = StudNum_approximationError(Ratings, Predicted)
```



Figuur 2: `spy(Ratings)`

die als resultaat `err` een maat voor de relatieve fout tussen de gekende beoordelingen `Ratings` en de voorspelde beoordelingen `Predicted` geeft. Concreet is het de vierkantswortel van de deling van de som van de kwadraten van het verschil tussen een gekende beoordeling in `Ratings` en de overeenkomstige geschatte beoordeling in `Predicted` door de som van de kwadraten van de gekende beoordelingen in de beoordelingenmatrix `Ratings`. Je mag veronderstellen dat de twee invoerargumenten `Ratings` en `Predicted` matrices zijn met dezelfde dimensies en dat onbekende beoordelingen in `Ratings` worden voorgesteld door de waarde 0. ⧫

Opdracht 3. Schrijf een functie met de hoofding

```
function [ Predicted, err ] = StudNum_predictRatings(Ratings, r, kappa)
```

die Algoritme 1 implementeert. De invoer `Ratings` is de onvolledige beoordelingenmatrix R. De rang r van de lagerangbenaderingen die worden opgesteld in de lus in Algoritme 1 wordt gegeven door het argument `r`. Het laatste invoerargument `kappa` is het aantal iteraties κ. Het uitvoerargument `Predicted` van de functie is de matrix A. Het uitvoerargument `err` is een rij van lengte κ dewelke op positie i de relatieve benaderingsfout bevat tussen de voorspelde beoordelingenmatrix A_i en de gekende beoordelingenmatrix R. Maak gebruik van de functie uit de voorgaande opdracht om deze relatieve benaderingsfout te berekenen. ⧫

Opdracht 4. Pas bovenstaande functie toe op de gekende beoordelingenmatrix `Ratings` die je hebt ingeladen in Opdracht 1, waarbij je $r = 15$ en $\kappa = 10000$ als parameterwaarden kiest. Maak een duidelijke figuur van de evolutie van de benaderingsfout van het iteratieve algoritme uit de voorgaande opdracht, zoals deze gegeven wordt door het tweede uitvoerargument `err`. Hoeveel bedraagt de convergentie-orde? Verklaar duidelijk (met behulp van formules) hoe je deze kan aflezen uit je figuur. ⧫

Opdracht 5. Schrijf een functie met de hoofding

```
function [] = StudNum_plotAllApproximationErrors(Ratings)
```

die een grafiek plot van de relatieve benaderingsfout tussen de gekende beoordelingenmatrix `Ratings` en de voorspelde beoordelingenmatrix in functie van de rang $r = 1, 2, 3, \ldots, 40$ van de lagerangbenaderingen. Gebruik $\kappa = 5000$ voor alle berekeningen. Neem de gegenereerde figuur op in het verslag.

Wat is de kleinste rang waarvoor de benaderingsfout kleiner is dan 10^{-13}? Merk op dat we voor deze keuze van de rang dus mogen spreken over statistische *imputation* van de ontbrekende waarden in de onvolledige gebruiker-beoordelingenmatrix R. ⧫

Opdracht 6. Bereken de voorspelde beoordelingenmatrix met de functie uit Opdracht 3 met R de gekende beoordelingenmatrix `Ratings`, $r = 26$ en $\kappa = 5000$. In wat volgt duiden we deze voorspelde beoordelingenmatrix aan met `Predicted26`. Voer het commando

```
imagesc(Predicted26); colorbar; axis('square');
```

uit en neem deze figuur op in het verslag. ♦

Opdracht 7. We kunnen de voorspelde beoordelingen in `Predicted26` reeds gebruiken om gelijkaardige boeken te detecteren. Hiertoe kunnen we kijken hoe gelijkaardig de beoordelingen van alle gebruikers voor twee boeken zijn. Om de gelijkaardigheid tussen twee boeken te meten, gaan we als volgt te werk. Stel dat $\mathbf{g} \in \mathbb{R}^{174}$ de vector van geschatte beoordelingen van boek i bevat en dat $\mathbf{h} \in \mathbb{R}^{174}$ de geschatte beoordelingen van boek j bevat. Dan wordt de principale hoek tussen deze vectoren gegeven door

$$p(\mathbf{g}, \mathbf{h}) = \arccos\left(\frac{|\mathbf{g}^T\mathbf{h}|}{\|\mathbf{g}\|_2\|\mathbf{h}\|_2}\right) \in [0, \tfrac{\pi}{2}].$$

Hoe kleiner de principale hoek, hoe gelijkaardiger de boeken zouden zijn volgens deze similariteitsmaat. Schrijf een functie met hoofding

```
function [books] = StudNum_similarBooks(bookNumber,nb,Model,labels)
```

die een cell array `books` van lengte `nb` oplevert met daarin de titels van de `nb` boeken van wie de principale hoeken met het boek met nummer `bookNumber` het kleinst zijn, gesorteerd volgens oplopende principale hoek. De titel van het boek met nummer `bookNumber` verschijnt dus steeds als eerste element in `books` aangezien $p(\mathbf{g}, \mathbf{g}) = 0$ voor elke niet-triviale vector $\mathbf{g} \in \mathbb{R}^m$. Het laatste invoerargument `labels` bevat de *cell array* met de namen van de boeken uit de catalogus zoals je deze hebt ingeladen in Opdracht 1. ♦

Opdracht 8. Welke 6 boeken lijken het meeste op het eerste boek uit de catalogus, *Harry Potter and the Sorcerer's Stone*? Welke 3 boeken lijken het meeste op het eenentwintigste boek, *A Game of Thrones: A Song of Ice and Fire: Book One*? Welke 10 boeken lijken het meeste op het honderdeerste boek, *The Arrangement 2*? Zijn deze resultaten realistisch? Indien niet, verklaar omstandig. ♦

Opdracht 9. Beschouw de singuliere waarden van `Predicted26` en plot deze door middel van het commando `semilogy(svd(Predicted26), 'x')`. Neem deze figuur op in het verslag. Waarom zijn de singuliere waarden met volgnummers strikt groter dan 26 niet van de grootte-orde 10^{-13}, ondanks het feit de gekende beoordelingen in R correct geïnterpoleerd worden door de rang-26 matrix A_{5000} in Algoritme 1? ♦

2.2 Clustering

In de vorige sectie werd een methode besproken om de ontbrekende waarden in de onvolledige beoordelingenmatrix R te vervangen door zinvolle waarden, resulterende in de voorspelde beoordelingenmatrix `Predicted26`. Laat ons deze specifieke voorspelde beoordelingenmatrix in het vervolg aanduiden met A. Elk van de $n = 110$ kolommen van A bevat nu de (voorspelde) beoordelingen voor elk van de $m = 174$ gebruikers. We kunnen nu de correlatiematrix $C \in \mathbb{R}^{n\times n}$ van de stochastische variabelen $B_1, B_2, \ldots, B_n$ schatten op basis van de steekproef A. Deze steekproefcorrelatiematrix wordt als volgt elementsgewijs gedefinieerd:

$$c_{i,j} = \mathrm{corr}(B_i, B_j) = \frac{1}{m-1}\frac{\sum_{k=1}^m (a_{k,i} - \mu_{B_i})(a_{k,j} - \mu_{B_j})}{\sigma_{B_i}\sigma_{B_j}},$$

waarbij

$$\mu_{B_i} = \frac{1}{m}\sum_{k=1}^m a_{k,i} \quad \text{en} \quad \sigma_{B_i} = \sqrt{\frac{1}{m-1}\sum_{k=1}^m (a_{k,i} - \mu_{B_i})^2}$$

respectievelijk het steekproefgemiddelde en de steekproefstandaardafwijking zijn. Zij

$$\Sigma = \mathrm{diag}(\sigma_{B_1}, \sigma_{B_2}, \ldots, \sigma_{B_n})$$

een diagonaalmatrix die de steekproefstandaardafwijkingen bevat van de stochastische variabelen B_i, $\boldsymbol{\mu}^T = \begin{bmatrix}\mu_{B_1} & \mu_{B_2} & \cdots & \mu_{B_n}\end{bmatrix} \in \mathbb{R}^n$ een vector die de steekproefgemiddelden van de statistische variabelen B_i bevat en $\mathbf{1}^T = \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix} \in \mathbb{R}^m$. De steekproefcorrelatiematrix wordt dan expliciet gegeven door

$$C = \frac{1}{m-1}\big((A - \mathbf{1}\boldsymbol{\mu}^T)\Sigma^{-1}\big)^T\big((A - \mathbf{1}\boldsymbol{\mu}^T)\Sigma^{-1}\big). \tag{3}$$

Zo wordt de steekproefcorrelatiematrix van de voorspelde beoordelingenmatrix A in het voorbeeld uit de voorgaande sectie gegeven door

$$C = \begin{bmatrix} 1.0000 & -0.0057 & 0.8856 & 0.8883 & 0.1022 \\ -0.0057 & 1.0000 & 0.3176 & -0.2174 & 0.9856 \\ 0.8856 & 0.3176 & 1.0000 & 0.6657 & 0.4159 \\ 0.8883 & -0.2174 & 0.6657 & 1.0000 & -0.0812 \\ 0.1022 & 0.9856 & 0.4159 & -0.0812 & 1.0000 \end{bmatrix}.$$

De volgende stap bestaat eruit om sterk gecorreleerde boeken te ontwaren uit de steekproefcorrelatiematrix C. Hiertoe zullen we enkele technieken uit de grafentheorie aanwenden. Zij $V = \{B_1, B_2, \ldots, B_n\}$ de verzameling van stochastische variabelen die de score van een boek voorstellen. We zeggen dat twee stochastische variabelen B_i en B_j "sterk gecorreleerd" zijn wanneer $c_{i,j} = \operatorname{corr}(B_i, B_j) \geq \tau$ met $\tau \approx 1$ een gegeven waarde. Laat E de verzameling zijn van alle paren van stochastische variabelen (B_i, B_j) die sterk gecorreleerd zijn:

$$E = \{(B_i, B_j) \in V \times V \mid \operatorname{corr}(B_i, B_j) \geq \tau\}. \tag{4}$$

Dan definieert $G = (V, E)$ een grafe G. Herinner dat de incidentiematrix H van een grafe G de matrix is waarvoor

$$h_{i,j} = \begin{cases} 1 & \text{als } (B_i, B_j) \in E, \\ 0 & \text{anders.} \end{cases}$$

Wanneer we bijvoorbeeld $\tau = 0.85$ kiezen, dan wordt de incidentiematrix H van de grafe $G = (V, E)$ in het voorbeeld gegeven door

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

We zullen veronderstellen dat de relatie "is sterk gecorreleerd met" een reflexieve, symmetrische en transitieve relatie is. Concreet wil dit zeggen dat wanneer er een sterke correlatie bestaat tussen de scores van boek i en boek k alsook tussen de scores van boek k en boek j, dan veronderstellen we dat er ook een sterke correlatie bestaat tussen de scores van boek i en j. In onze toepassing willen we de stochastische variabelen, of equivalent de knopen van G, zo partitioneren dat alle knopen in een van de clusters met elkaar verbonden zijn via de transitieve "is sterk gecorreleerd met" relatie. In de grafentheorie stelt men dat we zoeken naar de geconnecteerde componenten van G. Er bestaan verschillende algoritmen om zulke geconnecteerde componenten te vinden. In dit practicum zullen we een eenvoudig geïtereerd kwadrateren algoritme gebruiken. Laat H_k een incidentiematrix van een grafe G zijn en stel $K_k = H_k \cdot H_k = H_k^2$ waarbij $H_k \cdot H_k$ het gebruikelijke matrixproduct is. Dan definieert men H_{k+1} als volgt:

$$(H_{k+1})_{i,j} = \begin{cases} 1 & \text{als } (K_k)_{i,j} > 0, \\ 0 & \text{anders.} \end{cases}$$

Merk op dat H_{k+1} opnieuw een incidentiematrix van een grafe is, zodat we bovenstaande operatie ook kunnen toepassen op H_{k+1}. Het is niet moeilijk om in te zien dat er een (minimale) ℓ bestaat zodanig dat $H_\ell = H_{\ell+1} = \cdots = H_\infty$. De grafe die overeenstemt met de incidentiematrix H_∞ heeft de eigenschap dat ze bestaat uit een verzameling van cliques. Deze cliques definieren de partitionering van de stochastische variabelen B_i. Beschouw bij wijze van voorbeeld de grafe met incidentiematrix $H_1 = H$ uit vergelijking (5). We vinden bij de eerste kwadratering van H_1 de volgende matrices:

$$K_1 = H_1^2 = \begin{bmatrix} 3 & 0 & 2 & 2 & 0 \\ 0 & 2 & 0 & 0 & 2 \\ 2 & 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 2 & 0 \\ 0 & 2 & 0 & 0 & 2 \end{bmatrix} \text{ en } H_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Bij de volgende kwadratering bekomen we

$$K_2 = H_2^2 = \begin{bmatrix} 3 & 0 & 3 & 3 & 0 \\ 0 & 2 & 0 & 0 & 2 \\ 3 & 0 & 3 & 3 & 0 \\ 3 & 0 & 3 & 3 & 0 \\ 0 & 2 & 0 & 0 & 2 \end{bmatrix} \text{ en } H_3 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Daar $H_1 \neq H_2 = H_3 = \cdots = H_\infty$ volgt dat $\ell = 2$. Men kan nagaan dat H_∞ twee cliques bevat, namelijk deze met knopen $\{B_1, B_3, B_4\}$ en $\{B_2, B_5\}$. We besluiten dat de boeken *Catch-22*, *Dune* en *I Am Legend* een cluster van gerelateerde boeken vormen, alsook de twee boeken *Cloud Atlas* en *The Road*.

Opdracht 10. Schrijf een functie met hoofding

```
function [C] = StudNum_correlationMatrix(Predicted)
```

die de steekproefcorrelatiematrix $C \in \mathbb{R}^{n \times n}$ berekent van de steekproef in de $m \times n$ matrix `Predicted`. ♦

Opdracht 11. Bereken met de functie uit de voorgaande opdracht de correlatiematrix `C26` op basis van de voorspelde beoordelingenmatrix `Predicted26`. Voer het commando `imagesc(C26); colorbar` uit en neem de gegenereerde figuur op in het verslag.

Opdracht 12. Schrijf een functie met de volgende hoofding:

```
function [Hinf] = StudNum_buildCliques(C, tau)
```

Het invoerargument `C` is een correlatiematrix en `tau` is de parameterwaarde voor τ in vergelijking (4). Laat H_1 de incidentiematrix voorstellen van de grafe met knopen $V = \{B_1, B_2, \ldots, B_n\}$ en bogen E zoals gedefinieerd in vergelijking (4). Het uitvoerargument `Hinf` van de functie is de incidentiematrix H_∞ die je bekomt door het geïtereerd kwadrateren algoritme toe te passen op H_1. ♦

Opdracht 13. Bereken met de functie uit de voorgaande opdracht de incidentiematrix `Hinf26`. Als invoer voor de functie gebruik je de correlatiematrix `C26` en $\tau = 0.82$. Voer het commando

```
p = symamd(Hinf26); figure; spy(Hinf26(p,p)); figure; imagesc(C26(p,p)); colorbar
```

uit en neem de gegenereerde figuren op in het verslag.

Opdracht 14. Schrijf een functie met hoofding

```
function [Cluster] = StudNum_cluster(Hinf, labels)
```

die als uitvoer `Cluster` een *cell array* geeft met een lengte die gelijk is aan het aantal cliques in de grafe voorgesteld door de incidentiematrix `Hinf`. In de i^{de} cel van `Cluster` bevindt zich een *cell array* met de labels van de boeken die tot de i^{de} cluster behoren. Als invoer aanvaardt de functie de incidentiematrix `Hinf` van een grafe die enkel uit cliques bestaat, zoals bijvoorbeeld de matrices die als uitvoer van de functie uit Opdracht 12 optreden. Het tweede invoerargument `labels` bevat de labels van de boeken zoals je deze hebt ingeladen in Opdracht 1.

Hint: wanneer je `p = symamd(X)` in Matlab toepast op een incidentiematrix `X` van een grafe die enkel uit cliques bestaat, dan zal de uitvoer van deze functie altijd een permutatievector zijn met de eigenschap dat `X(p,p)` een blokdiagonaalmatrix is. ♦

Opdracht 15. Pas de voorgaande functie toe op `Hinf26` en neem alle clusters die uit minstens 3 boeken bestaan op in het verslag. Vind je deze resultaten realistisch? Indien niet, verklaar omstandig. ♦

Opdracht 16. Bereken met de functie uit Opdracht 12 de incidentiematrix `Hinf26b` waarbij je de correlatiematrix `C26` en $\tau = 0.5$ als invoerargumenten kiest. Voer vervolgens het volgende commando uit

```
p = symamd(Hinf26b); imagesc(C26(p,p)); colorbar; colormap jet
```

en neem de gegenereerde figuur op in het verslag. Heb je een verklaring voor de bijzondere structuur van de correlatiematrix `C26`? Interpreteer het voorkomen van de twee grote clusters die duidelijk negatief gecorreleerd zijn door op Amazon.com de covers van de boeken op te zoeken. Om je argument kracht bij te zetten neem je enkele van deze covers op in het verslag. ♦

Opdracht 17. Bewijs dat de correlatiematrix C in vergelijking (3) hoogstens rang $r + 1$ heeft wanneer de voorspelde beoordelingenmatrix A exact rang r heeft. Hint: denk aan vergelijking (2). ♦

Opdracht 18. Bewijs dat het matrixproduct $K = H^2 = H \cdot H$ op positie (i, j) enkel verschilt van 0 indien er een pad van lengte hoogstens 2 bestaat tussen knoop i en j. Hint: beschouw de definitie van $k_{i,j}$ in het matrixproduct. ♦

3 Evaluatie

Opdracht 19. Hoeveel tijd heb je gespendeerd aan het oplossen van de opdrachten? Hoeveel tijd heb je gespendeerd aan het schrijven van het verslag?

Opdracht 20. In de loop van deze opgave hebben we allerhande veronderstellingen gemaakt om ons nieuw aanbevelingssysteem op te stellen. Wat zijn je bedenkingen hierbij? Vind je de resultaten realistisch? Zou je het aanbevelingssysteem van Amazon.com durven vervangen door het aanbevelingssysteem dat je in dit practicum hebt geïmplementeerd? Wat zijn de voordelen en nadelen van een aanpak gebaseerd op gebruikersbeoordelingen in vergelijking met het huidige systeem dat Amazon.com gebruikt en dat gebaseerd is op slechts 1 bit aan informatie (namelijk of een gebruiker een bepaald product heeft gekocht)?

Opdracht 21. Welke eindbedenkingen heb je bij dit practicum? Was de opgave (veel) te gemakkelijk, (veel) te moeilijk of van een gepaste moeilijkheidsgraad? Wat zou je zelf anders aangepakt hebben?