Informationssökning med linjär algebra

Startpunkt: En databas med ett antal dokument. Denna är indexerad antingen med (nästan) alla ingående ord eller med utvalda nyckelord.
Modell: Vi representerar den indexerade databasen som en term-dokument-matris A där varje rad svarar mot en viss term (ord) i databasen och varje kolumn svarar mot ett visst dokument. Enklaste varianten är att låta elementen i A vara

$$a_{ij} = \begin{cases} 1, & \text{om ord } i \text{ finns i dokument } j, \\ 0, & \text{annars.} \end{cases}$$

Exempel

1. Introduktion till den diskreta matematiken.
2. Diskret matematik: logik, relationer och grafer.
3. "När Harry mötte Sally", en relations- komedi.
4. Matematisk analys.
5. Utomjordiska relationer, ufon och marsmänniskor.
6. Mänskliga relationer i IT-åldern.

Matrisen blir då:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Sökning

Antag nu att vi vill söka i databasen, t ex efter dokument som behandlar matematiska relationer. Vi skickar kanske in en sökning:

matematik relation

Denna kan representeras som vektorn

$$\mathbf{q} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^t$$

eftersom matematik är ord nummer 3 och relation är ord nummer 5.
Vi vill nu returnera "relevanta" dokument, d v s sådana som ligger "nära" vektorn $\mathbf{q}$. Hur ska man mäta "närhet"?
Variant 1: (Enklast tänkbara) Man räknar helt enkelt hur många av de sökta orden som finns i respektive dokument. Matematiskt är detta ekvivalent med att ta skalärprodukten mellan $\mathbf{q}$ och kolumnerna i A.
På matrisform kan man skriva detta som $A^t\mathbf{q}$ som då ger vektorn med alla skalärprodukter.
I vårt exempel så får vi

$$A^t\mathbf{q} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Detta returnerar alltså dokument 2 som mest relevant och alla andra som lika relevanta.
Variant 2: Populärt är att normera kolumnerna i A (och även $\mathbf{q}$). Detta undviker t ex att verkligt generella dokument (extremexempel är en ordbok) hela tiden returneras trots att de saknar egentlig relevans. Man vill (förmodligen) ha dokument som verkligen behandlar just det man söker.
Så istället för $\mathbf{a}_i^t\mathbf{q}$ så tar man

$$\frac{\mathbf{a}_i^t\mathbf{q}}{|\,\mathbf{a}_i\,||\,\mathbf{q}\,|} = \cos\theta,$$

där θ är vinkeln mellan $\mathbf{a}_i$ och $\mathbf{q}$.
Normeringen av kolumnerna i A gör man lämpligen från början så att A har enhetsvektorer i kolumnerna.
I vårt fall så blir då

$$A = \begin{pmatrix} 0.5774 & 0 & 0 & 0 & 0 & 0 \\ 0.5774 & 0.4472 & 0 & 0 & 0 & 0 \\ 0.5774 & 0.4472 & 0 & 0.7071 & 0 & 0 \\ 0 & 0.4472 & 0 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.5000 & 0 & 0.5000 & 0.5774 \\ 0 & 0.4472 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5000 & 0 & 0 & 0 \\ 0 & 0 & 0.5000 & 0 & 0 & 0 \\ 0 & 0 & 0.5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7071 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5000 & 0 \\ 0 & 0 & 0 & 0 & 0.5000 & 0 \\ 0 & 0 & 0 & 0 & 0.5000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5774 \\ 0 & 0 & 0 & 0 & 0 & 0.5774 \end{pmatrix}$$

och eftersom $\mid \mathbf{q} \mid = \sqrt{2}$ så blir cosinus för vinklarna

$$\frac{1}{\sqrt{2}} A^t \mathbf{q} = \begin{pmatrix} 0.4082 \\ 0.6325 \\ 0.3536 \\ 0.5000 \\ 0.3536 \\ 0.4082 \end{pmatrix}$$

Förbättringar

Viktning: Ett sätt att förbättra resultatet är att vikta term-dokument-matrisen och/eller sökvektorn. Istället för att bara ge värden 0 eller 1 så ger man ett värde som säger hur viktig termen är för dokumentet eller sökningen. Matrisen viktar man främst på två sätt:

- Varje term i ett givet dokument får en vikt efter hur viktig den anses vara för dokumentet. Kan t ex vara antalet gånger det förekommer, om det förekommer i titel/i början av dokumentet eller en subjektiv bedömning av dess betydelse.
- Varje term får en vikt efter hur viktig den anses vara i databasen. Exempelvis skulle kanske matematik ha liten vikt i en databas med matematikböcker (eftersom det är relevant för alla och inte viktigt vid sökning), medan det kanske skulle ha hög vikt i en databas med alla möjliga böcker eftersom det är en term som skiljer ut en viss grupp av böcker.

Elementen i matrisen A beräknas i regel som en produkt av dessa två typer:

$$a_{ij} = b(i,j)c(i),$$

där $b(i,j)$ är av den första typen (beror på termen och dokumentet) medan $c(i)$ är av den andra typen och alltså bara beror på termen (d v s raden i A). I vårt exempel kan man t ex tänka sig att man gör följande (subjektiva) viktning av matrisen:

$$A = \begin{pmatrix} 0.3333 & 0 & 0 & 0 & 0 & 0 \\ 0.6667 & 0.6030 & 0 & 0 & 0 & 0 \\ 0.6667 & 0.6030 & 0 & 0.8944 & 0 & 0 \\ 0 & 0.3015 & 0 & 0 & 0 & 0 \\ 0 & 0.3015 & 0.3780 & 0 & 0.3780 & 0.5774 \\ 0 & 0.3015 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3780 & 0 & 0 & 0 \\ 0 & 0 & 0.3780 & 0 & 0 & 0 \\ 0 & 0 & 0.7559 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7559 & 0 \\ 0 & 0 & 0 & 0 & 0.3780 & 0 \\ 0 & 0 & 0 & 0 & 0.3780 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5774 \\ 0 & 0 & 0 & 0 & 0 & 0.5774 \end{pmatrix}$$

Detta skulle i sin tur ge följande sökresultat:

$$\frac{1}{\sqrt{2}}A^t\mathbf{q} = \begin{pmatrix} 0.4714 \\ 0.6396 \\ 0.2673 \\ 0.6325 \\ 0.2673 \\ 0.4082 \end{pmatrix}$$

Man kan också tänka sig att man viktar sökningen. I vårt exempel skulle man kunna tänka sig att låta matematik få högre vikt för att utesluta alla andra betydelser av relation. Om vi t ex ger matematik dubbla vikten så blir den normerade sökvektorn

$$\mathbf{q} = \begin{pmatrix} 0 & 0 & 0.8944 & 0 & 0.4472 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^t$$

Resultatet från sökningen blir då

$$A^t\mathbf{q} = \begin{pmatrix} 0.5963 \\ 0.6742 \\ 0.1690 \\ 0.8000 \\ 0.1690 \\ 0.2582 \end{pmatrix}$$

Upprepad sökning med feedback: Ett sätt att förbättra sökresultatet är att låta den som söker markera ett eller flera dokument från första sökningen som kan anses relevanta och sedan modifiera sökvektorn.
Om $\mathbf{q}$ är den ursprungliga sökvektorn och den som söker markerar det dokument som har kolumn $\mathbf{a}_i$ så tar man t ex istället

$$\tilde{\mathbf{q}} = \mathbf{q} + \mathbf{a}_i.$$

I vårt exempel markerar man förmodligen nummer 2 och får i andra sökningen följande resultat:

$$A^t\tilde{\mathbf{q}} = \begin{pmatrix} 0.7043 \\ 0.9054 \\ 0.2105 \\ 0.6471 \\ 0.2105 \\ 0.3216 \end{pmatrix}$$

Singulärvärdesuppdelning (SVD)

Låt A vara en $t \times d$-matris (t ex vår term-dokument-matris). Man kan visa att det finns ON-matriser U respektive V av storlek $t \times t$ respektive $d \times d$ samt en ”diagonal” matris

$$S = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k),$$

där $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > \sigma_{r+1} = \cdots = \sigma_k = 0$ och $k = \min\{t, d\}$ sådana att

$$A = USV^t.$$

Observera att A är av typ $t \times d$ och alltså inte behöver vara en kvadratisk matris.
I vårt exempel får man följande matriser om man beräknar SVD av den viktade matrisen:

$$U = \begin{pmatrix} -0.1304 & 0.0427 & -0.0000 & -0.0071 & -0.1610 & \cdots \\ -0.4977 & 0.0618 & -0.0000 & 0.0161 & -0.6719 & \cdots \\ -0.7990 & 0.1855 & 0.0000 & -0.0369 & 0.3935 & \cdots \\ -0.1185 & -0.0118 & -0.0000 & 0.0152 & -0.1750 & \cdots \\ -0.1964 & -0.6522 & -0.0000 & 0.1375 & -0.0468 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$S = \begin{pmatrix} 1.5237 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.1829 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9258 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8696 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7039 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4124 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} -0.5959 & 0.1514 & 0 & -0.0186 & -0.3399 & -0.7114 \\ -0.5989 & -0.0462 & -0.0000 & 0.0439 & -0.4085 & 0.6858 \\ -0.0772 & -0.5378 & -0.7071 & -0.4479 & 0.0344 & -0.0545 \\ -0.5132 & 0.1637 & 0.0000 & -0.0515 & 0.8384 & 0.0654 \\ -0.0772 & -0.5378 & 0.7071 & -0.4479 & 0.0344 & -0.0545 \\ -0.1044 & -0.6080 & -0.0000 & 0.7706 & 0.1111 & -0.1152 \end{pmatrix}$$

Här är U en 15×15-matris, S en 15×6-matris och V en 6×6-matris.
Av diagonalelementen i S är de r första positiva. Detta tal r är lika med antalet linjärt oberoende kolumner i A (vilket kallas för rangen av A).
Låt $\{\mathbf{u}_i\}$ vara kolumnerna i U och $\{\mathbf{v}_i\}$ kolumnerna i V. Produkten mellan de tre matriserna kan då skrivas som

$$A = USV^t = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^t.$$

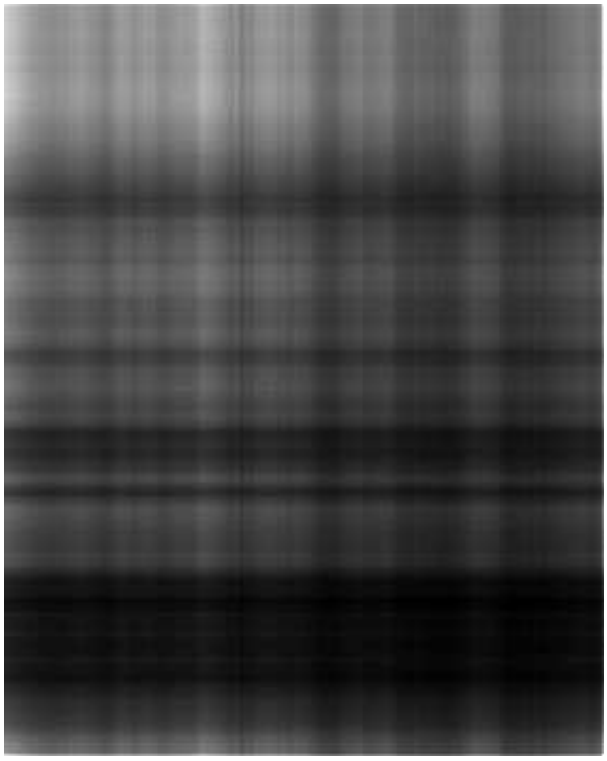
Observera att $\mathbf{u}_i\mathbf{v}_i^t$ är en $t \times d$-matris så man får A som en summa av r stycken matriser.

Eftersom $|\mathbf{u}_i| = |\mathbf{v}_i| = 1$ för alla i och σ_i avtar med i så blir bidragen i summan mindre och mindre.
Vi betraktar matriserna

$$A = USV^t = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^t.$$

för olika $k < r$ där vi helt enkelt ”kastar bort” de sista termerna. Det visar sig att detta ofta är en mycket bra approximation av A även för k mycket mindre än r.

1 terms in approximation.



Type q to quit. Next image: Click mouse.

10 terms in approximation.

Type q to quit. Next image: Click mouse.

50 terms in approximation.

Type q to quit. Next image: Click mouse.

352−by−281 image.

Man kan betrakta de bortkastade termerna som ”brus” som har liten relevans för den ursprungliga matrisen.
I fallet då matrisen representerar en bild så kan detta användas för att komprimera datan. (Man behöver bara spara $\mathbf{u}_1, \ldots, \mathbf{u}_k$, $\mathbf{v}_1, \ldots, \mathbf{v}_k$ samt $\sigma_1, \ldots, \sigma_k$ vilket är mindre än A om $k < r/2$.)

SVD och datasökning

Vi ska nu applicera SVD på vår datasökningsmodell. Precis som i bildexemplet så ersätter vi A med approximationen A_k för lämpligt k och beräknar vinkeln mellan kolumnerna i A_k och sökvektorn.
Sökresultat när man ersatt den viktade matrisen A med approximationerna A_k för $k = 1, \ldots, 6$. (Observera att $A_6 = A$.)
Fall 1: Ej viktad sökvektor.

$$\begin{pmatrix} 0.6390 & 0.5799 & 0.5799 & 0.5788 & 0.5201 & 0.4714 \\ 0.6423 & 0.6604 & 0.6604 & 0.6631 & 0.5926 & 0.6396 \\ 0.0828 & 0.2928 & 0.2928 & 0.2651 & 0.2710 & 0.2673 \\ 0.5504 & 0.4865 & 0.4865 & 0.4833 & 0.6280 & 0.6325 \\ 0.0828 & 0.2928 & 0.2928 & 0.2651 & 0.2710 & 0.2673 \\ 0.1119 & 0.3493 & 0.3493 & 0.3970 & 0.4161 & 0.4082 \end{pmatrix}$$

Fall 2: Viktad sökvektor.

$$\begin{pmatrix} 0.7286 & 0.7061 & 0.7061 & 0.7056 & 0.6264 & 0.5963 \\ 0.7324 & 0.7392 & 0.7392 & 0.7403 & 0.6452 & 0.6742 \\ 0.0944 & 0.1744 & 0.1744 & 0.1633 & 0.1713 & 0.1690 \\ 0.6275 & 0.6032 & 0.6032 & 0.6019 & 0.7972 & 0.8000 \\ 0.0944 & 0.1744 & 0.1744 & 0.1633 & 0.1713 & 0.1690 \\ 0.1276 & 0.2181 & 0.2181 & 0.2372 & 0.2631 & 0.2582 \end{pmatrix}$$

Fall 3: Ej viktad sökvektor med feedback av kolumn 2.

$$\begin{pmatrix} 0.8105 & 0.7724 & 0.7724 & 0.7715 & 0.7771 & 0.7043 \\ 0.8147 & 0.8263 & 0.8263 & 0.8286 & 0.8353 & 0.9054 \\ 0.1050 & 0.2402 & 0.2402 & 0.2167 & 0.2161 & 0.2105 \\ 0.6981 & 0.6569 & 0.6569 & 0.6542 & 0.6404 & 0.6471 \\ 0.1050 & 0.2402 & 0.2402 & 0.2167 & 0.2161 & 0.2105 \\ 0.1420 & 0.2947 & 0.2947 & 0.3352 & 0.3334 & 0.3216 \end{pmatrix}$$

I praktiska tillämpningar är ofta t och/eller d GIGANTISKA. Då ger SVD-approximationen A_k också en avsevärd uppsnabbning av sökningen jämfört med att använda hela A.
Man behöver bara beräkna SVD en gång ända tills databasen ändras. Då finns dessutom olika trick att lägga in nya dokument utan att räkna om hela SVD.

Sammandrag av sökresultaten.

Direkt skalärprodukt:

$$A^t\mathbf{q} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Normerad skalärprodukt:

$$A^t\mathbf{q} = \begin{pmatrix} 0.4082 \\ 0.6325 \\ 0.3536 \\ 0.5000 \\ 0.3536 \\ 0.4082 \end{pmatrix}$$

Viktad term-dokument-matris:

$$A^t\mathbf{q} = \begin{pmatrix} 0.4714 \\ 0.6396 \\ 0.2673 \\ 0.6325 \\ 0.2673 \\ 0.4082 \end{pmatrix}$$

Viktad matris och sökvektor:

$$A^t\mathbf{q} = \begin{pmatrix} 0.5963 \\ 0.6742 \\ 0.1690 \\ 0.8000 \\ 0.1690 \\ 0.2582 \end{pmatrix}$$

Viktad matris med feedback:

$$A^t\tilde{\mathbf{q}} = \begin{pmatrix} 0.7043 \\ 0.9054 \\ 0.2105 \\ 0.6471 \\ 0.2105 \\ 0.3216 \end{pmatrix}$$