クラスタリングに基づく情報の検索と視覚化

堀田, 政二

https://doi.org/10.15017/1398256

付 録A

LSI法

潜在意味解析法 (Latent Semantic Indexing method. LSI 法) は，キーワードと文献間の潜在的な関係を特異値分解 (Singular Value Decomposition. SVD) で抽出することにより，ノイズにロバストな検索結果を得ることのできる検索法である [9. 37]. 例えば表 A.1 のようなキーワードと文献の共起関係が与えられたとする [9]. 表 A.1 の縦はキーワード，横の 1 から 9 は文献の番号であり，要素の 0. 1. 2 は文献にそのキーワードが登場する回数である．例えば，クエリとしてキーワード *trees* を入力し

表 A.1: 文献とキーワードの共起関係行列

keywords	documents								
	1	2	3	4	5	6	7	8	9
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

た場合，表 A.1 をそのまま使えば文献 6. 7. 8 が検索結果として出力される．しかし，表 A.1 を見るとわかるように，この場合，文献 9 も検索されるのが妥当である．なぜなら文献 9 はキーワード *trees* を直接含んではいないが *survey*. *graph*. *minors* を含んでおり，特に *graph*. *minors* を含んでいる文献 6. 7. 8 と内容的には類似していると推測できるからである．LSI 法で検索を行えば，文献 6. 7. 8 に加えて文献 9 も検索結果に含まれるようになる．本付録では LSI 法について概説し，簡単な実験例を示す．

A.1 LSI 法

m 個のキーワードと n 個の文章があり，キーワード i が文章 j に現れる頻度を $d_{ij} (i = 1, ..., m; j = 1, ..., n)$ とする．$m \geq n$ とする．行列 $D = [d_{ij}]$ の特異値分解は

$$D = U \Sigma V^T \tag{A.1}$$

で定義される [43]．ここで U は $m \times m$ 行列，Σ は $m \times n$ 行列，V は $n \times n$ 行列である．このうち Σ は対角行列であり，対角成分の特異値が左上から右下へ降順に並んでいる．LSI 法では，この特異値分解に基づいて D のランクを k に落とす．すなわち

$$D_k = U_k \Sigma_k V_k^T \tag{A.2}$$

とする．ここで U_k は U の最初の k 列を取り出した $m \times k$ 行列，Σ_k は Σ の左上 k 行 k 列を取り出した $k \times k$ 行列，V_k は V の最初の k 列を取り出した $n \times k$ 行列である．LSI 法による検索ではこれらの行列を利用する．クエリを表す m 次元ベクトルを $\boldsymbol{q} = [q_1, ..., q_m]^T$，文章 j を表すベクトルを $\boldsymbol{d}_j = [d_{1j}, ..., d_{mj}]^T (j = 1, ..., n)$，キーワード i を表すベクトルを $\boldsymbol{K}_i = [d_{i1}, ...d_{in}]^T (i = 1, ..., m)$ とする．各ベクトル $\boldsymbol{q}$，$\boldsymbol{d}_j$，$\boldsymbol{K}_i$ は，行列 U_k と V_k とおよび Σ_k の逆行列 Σ_k^{-1} によって，それぞれ以下の式で k 次元ベクトルに射影される [37]:

$$\boldsymbol{q}_k = \boldsymbol{q}^T U_k \Sigma_k^{-1} \tag{A.3}$$

$$\boldsymbol{d}_{jk} = \boldsymbol{d}_j^T U_k \Sigma_k^{-1} \tag{A.4}$$

$$\boldsymbol{K}_{ik} = \boldsymbol{K}_i^T V_k \Sigma_k^{-1} \tag{A.5}$$

キーワードによる文献の検索では，射影後のベクトル $\boldsymbol{q}_k$ と $\boldsymbol{d}_{jk}$ とのコサイン値

$$\cos \theta_j = \frac{\boldsymbol{d}_{jk} \boldsymbol{q}_k^T}{\|\boldsymbol{d}_{jk}\| \|\boldsymbol{q}_k\|} \tag{A.6}$$

を計算して値の大きな文献から出力する [37]．したがって，予めすべての文献について $\boldsymbol{d}_{jk}/\|\boldsymbol{d}_{jk}\|$ を計算しておき，$\boldsymbol{q}$ が入力されたら $\boldsymbol{q}^T U_k \Sigma_k^{-1} / \|\boldsymbol{q}^T U_k \Sigma_k^{-1}\|$ を計算して各

$\boldsymbol{d}_{jk}/\|\boldsymbol{d}_{jk}\|$ との内積を計算すればよい．この内積は k 次元ベクトル間の演算なので n 次元での演算よりも計算量が少ない．加えてノイズも平滑化されて検索性能も高められる [9, 37]．

A.2 実験例

まず最初に，表 A.1 の共起関係行列を特異値分解した．特異値分解によって得られた行列 U，Σ，V をそれぞれ $k = 2$ としてランクを落とした．次に，キーワード *trees* をクエリとして各文献とのコサイン値を計算した．図 A.1 に各文献のコサイン値を示す．図から，文献 6, 7, 8 と文献 9 のコサイン値が大きくなっており，妥当な結果となっている．

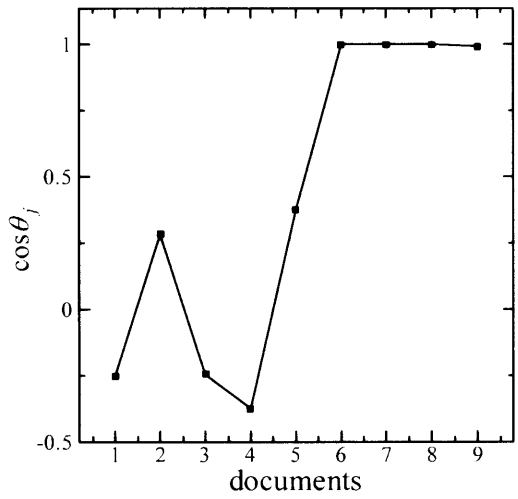

図 A.1: *trees* をクエリとしたときの各文献のコサイン値

LSI 法では $k \leq 3$ であればデータ分布を視覚化できる．図 A.2 に $k = 2$ として文献とキーワード，およびクエリを 2 次元平面上にプロットしたものを示す．図中の数字は文献の番号である．クエリは，図中の文献 6 と同じ座標となったので省略している．潜在的に関係の深い文献とキーワードが互いに近くに配置されているのがわかる．前述のハンティング検索はこの空間で行われており，射影後のクエリベクトルが x 軸となす方向ベクトル (図中矢印) の延長上に文献 6, 7, 8, 9 があることから，これらの文献とクエリとのコサイン値が大きくなることは視覚的にも明らかである．

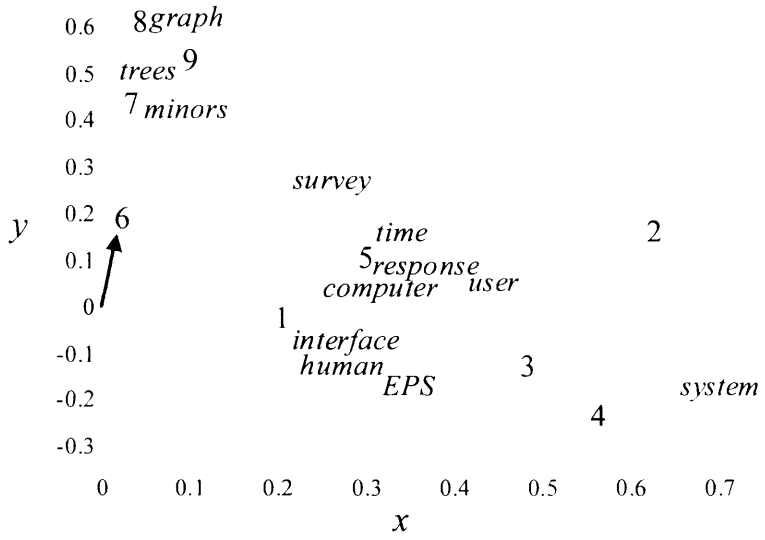

図 A.2: キーワードと文献の同時プロット

A.3 まとめ

LSI 法は，共起関係行列のランクを落とすことにより，ノイズの平滑化と検索の高速化を同時に実現した優れた検索手法である．しかし，1) 最適なランク数 k の決定方法が不明瞭であること，2) 他のデータ行列と組合せることが困難であること，などの問題点がある．