第1回教員ゼミ

# Vector Space Models for Search and Cluster Mining

From “Survey of Text Mining”

佐々木　稔

# 目次

- はじめに
- ベクトル空間モデル
- Latent Semantic Indexing
  (潜在的意味索引)
- Covariance Matrix Analysis
- 大小クラスタの発見手法
- 実験
- まとめ

# はじめに(1/2)

- 様々な形式を持つ大量のデータからの情報検索
- ベクトル空間モデル
  - 多くのデータフォーマットに対応
  - マルチメディアデータに対応
  - 多言語文書に対応
  - 検索処理を完全自動で行う
  - 計算作業のほとんどを前処理で行うため、検索処理が実時間で可能

# はじめに(2/2)

- VSMによるマイニング、クラスタのラベリングを紹介
  - 文書内での主要なクラスタと細かいクラスタを見つける
- 細かいクラスタを見つける研究はこれまでなかった
- 応用分野
  - 企業 : 顧客の不満についての細かい理由
  - 金融・保険業 : 利子率、保険料率の設定
  - 情報セキュリティ : 個人を特定するデータの発見
  - 科学 : 天気予報、自然災害の予知

# 発表のポイント

- 共分散行列を用いた特徴軸抽出手法の紹介
- 特徴
  - 高い拡張性
  - 大小クラスタを見つけることができる
  - クラスタ重複に対応
- 実験の結果、期待した結果が出力される

# ベクトル空間モデル

- 情報検索の数理モデルのひとつ
- 文書や検索質問をベクトル空間内の「点」と表現

$$\mathbf{d} = (t_1, t_2, \cdots, t_n)$$

- 通常、文書は大量に存在するので行列表現となる

$$\mathbf{d} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{bmatrix}$$

# ベクトル空間モデルの例(1/4)

D1:ソフトバンク、ボーダフォンと新会社
D2:ソフトバンク、ボーダーフォングループと合弁会社
D3:ボーダフォンは「ソフトバンクモバイル」に
D4:ソフトバンク、ボーダフォンと合弁・資本金最大110億円
D5:ソフトバンク、ボーダフォンを社名変更「ソフトバンクモバイル」に

- 索引語
  - t1 :ソフトバンク
  - t2 :ボーダフォン
  - t3 :会社
  - t4 :合弁
  - t5 :モバイル
  - t6 :資本金
  - t7 :最大
  - t8 :110億円
  - t9 :社名
  - t10:変更

# ベクトル空間モデルの例(2/4)

- 文書集合の行列表現

$$\begin{bmatrix} D1 \\ D2 \\ D3 \\ D4 \\ D5 \end{bmatrix} = \begin{bmatrix} 1110000000 \\ 1111000000 \\ 1100100000 \\ 1101011100 \\ 2100100011 \end{bmatrix}$$

# ベクトル空間モデルの例(3/4)

- 検索質問:「ソフトバンクモバイル」

$$q = (1, 0, 0, 0, 1, 0, 0, 0, 0, 0)$$

- 類似度計算(cosine)

$$\cos(D1, q) = 0.408$$
$$\cos(D2, q) = 0.354$$
$$\cos(D3, q) = 0.816$$
$$\cos(D4, q) = 0.289$$
$$\cos(D5, q) = 0.750$$

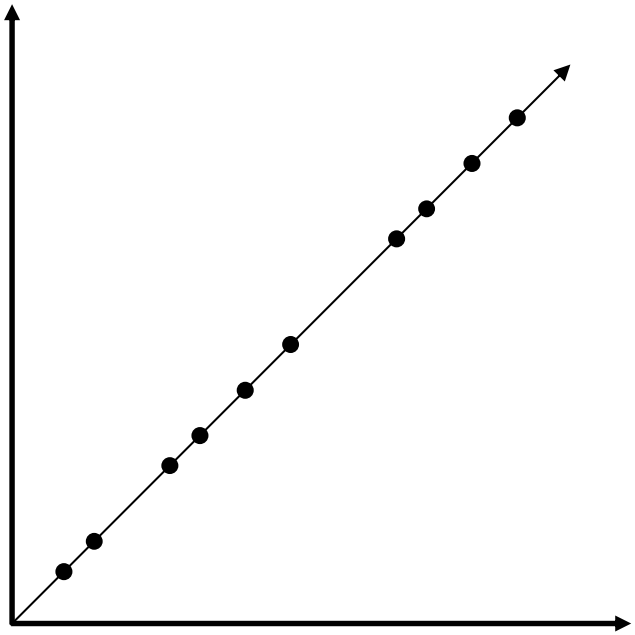
# ベクトル空間モデルの例(4/4)

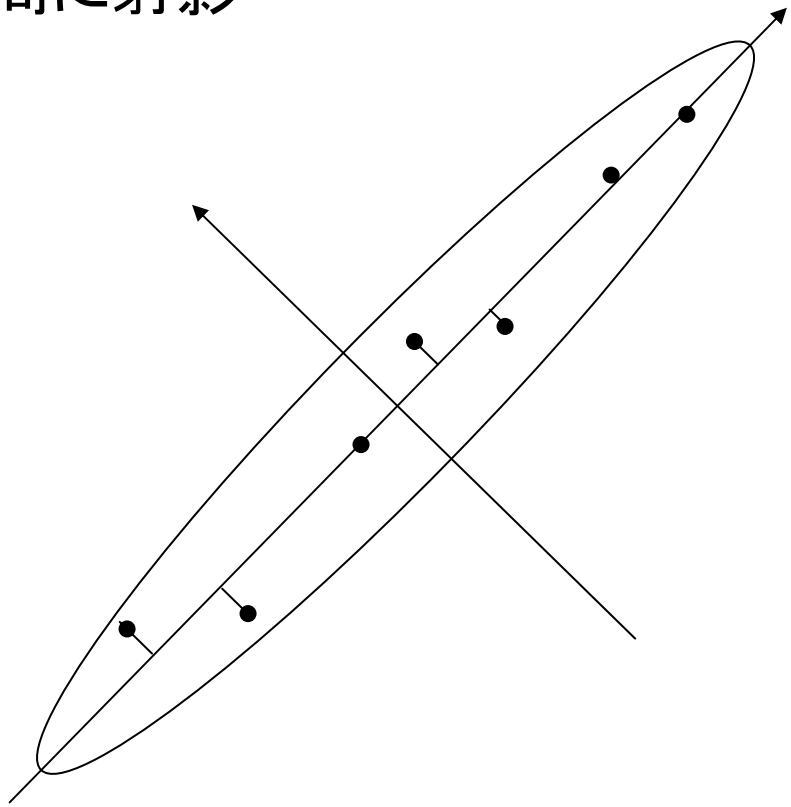
- 類似度の高い順に並び替える

| 順位 | 文書 | 類似度 |
|---|---|---|
| 1 | D3 | 0.816 |
| 2 | D5 | 0.750 |
| 3 | D1 | 0.408 |
| 4 | D2 | 0.354 |
| 5 | D4 | 0.289 |

# Latent Semantic Indexing (LSI)（1/5）

- 文書ベクトル
  - 高次元空間から低次元空間に射影





# Latent Semantic Indexing (LSI)（2/5）

- 特異値分解(Singular Value Decomposition)
- *m* 個の文書, *n* 個の索引語
  - *m* × *n* である文書-索引語行列 **A** (階数 *r* )

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

  - $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_r)$
    $(\sigma_1 \geqq \cdots \geqq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_n)$
  - $U = (u_1, u_2, \cdots, u_m)$, $U^TU = I$
  - $V = (v_1, v_2, \cdots, v_n)$, $VV^T = I$

# Latent Semantic Indexing (LSI)（3/5）

- 特異値ベクトルの計算方法

$$\hat{d}_j = \left[ b_1, \; b_2, \; b_3, \; \cdots, \; b_k \right]^T d_j$$

$$\mathbf{A}\mathbf{A}^T = \mathbf{U}\Lambda\mathbf{U}^T$$

- Λ は $AA^T$ の固有値 $\lambda_i$ が降順に並んだ$m$次対角行列
- U は $AA^T$ の固有ベクトル

# Latent Semantic Indexing (LSI)（4/5）

- 行列Λ$_{m \times n}$を考える

$$\Lambda_{m \times n} = \begin{bmatrix} \lambda_1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & 0_{r \times (n-r)} \\ 0 & \cdots & \lambda_r & \\ 0_{(m-r) \times r} & & & 0_{(m-r) \times (n-r)} \end{bmatrix}$$

$$\Sigma = \left(\Lambda_{m \times n}\right)^{\frac{1}{2}}$$

$$\mathbf{V} = \mathbf{D}^T \mathbf{U} \left(\Lambda_{m \times n}\right)^{-\frac{1}{2}}$$

# Latent Semantic Indexing (LSI)（5/5）

- 行列 **A** の次元削減
  - 行列 **Σ** の特異値で小さいものを 0 に置換

- 元の階数 $r$ より低い階数 $k$ の近似行列 $\mathbf{A}_k$

$$\left\|\mathbf{A}-\mathbf{A}_{\mathbf{k}}\right\| = \min_{\mathrm{rank}(\mathrm{A}')\le k}\left\|\mathbf{A}-\mathbf{A}'\right\|_F$$

$$= \sqrt{\sigma_{k+1}^2+\cdots+\sigma_r^2}$$

# Covariance Matrix Analysis (COV) (1/2)

- 文書-索引語行列 $A = \{d_1^T, d_2^T, \cdots, d_n^T\}$

$$d_i = [a_{i,1}\ a_{i,2}\ a_{i,3} \cdots a_{i,n}]$$

$$\bar{d} = [\bar{a}_1\ \bar{a}_2\ \bar{a}_3 \cdots \bar{a}_n]^T,\ \bar{a}_j = \frac{1}{m}\sum_{i=1}^{m} a_{i,j}$$

- 文書ベクトルの共分散行列

$$C \equiv \frac{1}{m}\sum_{i=1}^{m} d_i d_i^T - \bar{d}\bar{d}^T$$

# Covariance Matrix Analysis (COV) (2/2)

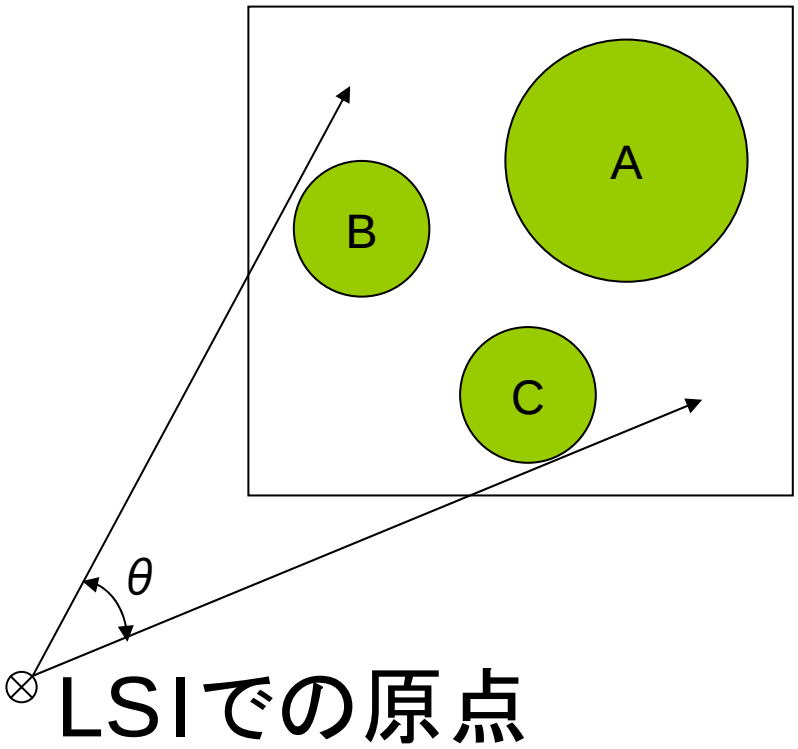
- $C$ の固有値（常に 0 以上）
  - 固有ベクトルを軸としたときの分散に等しい
- $C$ を対角化

$$C = V \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} V^T, (\lambda_i \geq \lambda_{i+1}, i = 1,2,\cdots,n-1)$$

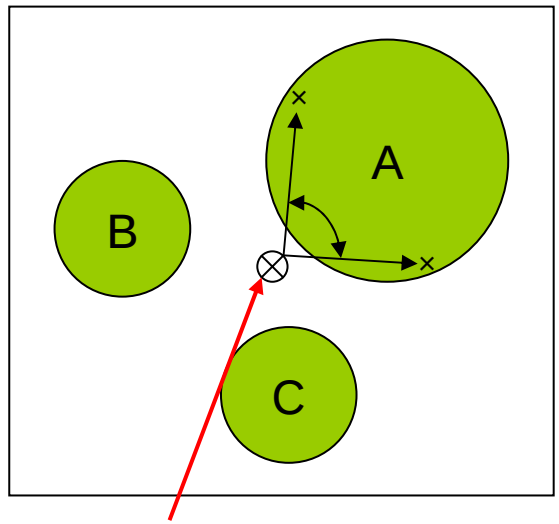
- $V$ : $C$ を対角化する直交行列
- k 個の固有値に対応する直交ベクトルに射影

# LSIとCOVの違い(1/2)

- 原点の取り方
  - LSI ： 原点は動かない
  - COV: 文書ベクトルを区別しやすいように原点が動く




LSIでの原点




COVでの原点

# LSIとCOVの違い(2/2)

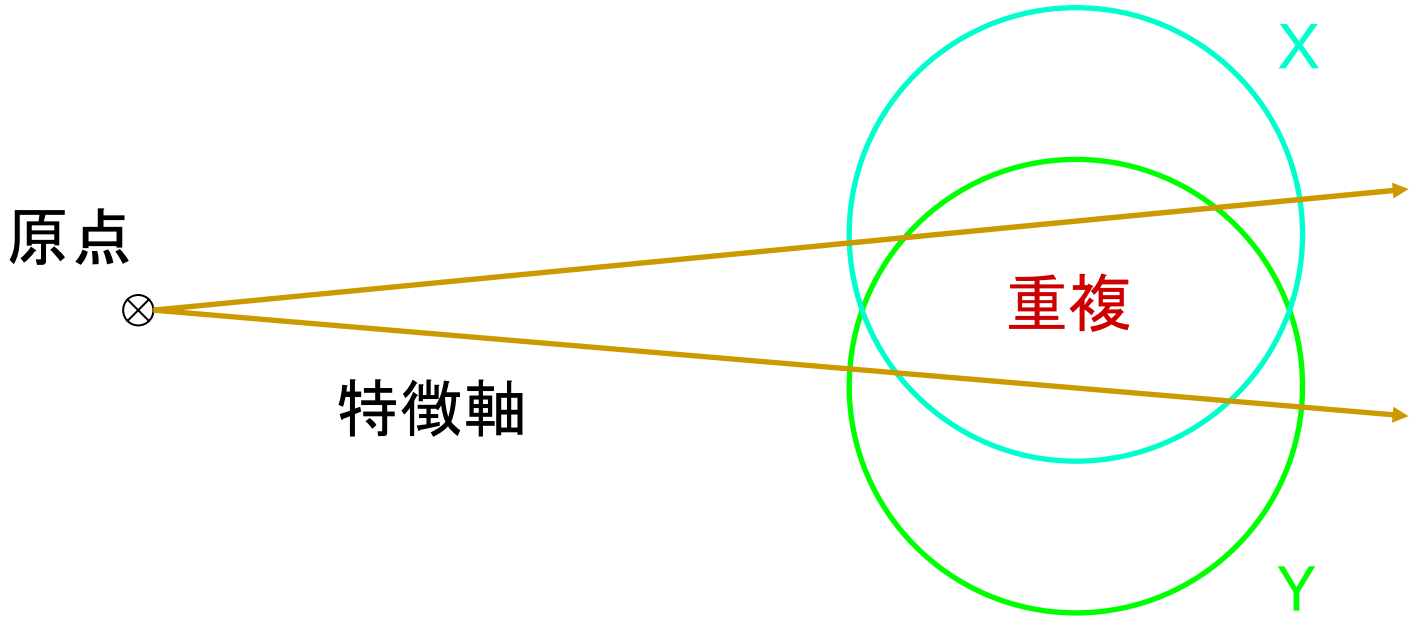
- 部分空間を決める基準
  - LSI ： フロベニウスノルムで A と最も近い k 次元行列
  - COV: 行ベクトルの最小二乗誤差が最小の k 次元行列
- 計算量
  - LSI ： 文書数増加で計算量が増加
  - COV: 索引語数のみに依存
    （2万語位であればメインメモリで実行可能）

# 大小クラスタの発見手法

- 「近くに集まった文書は、同じ質問に関連しやすい」
  （クラスタ仮説）
- 類似した内容を持つ文書集合
  - 前処理でまとめる（意味ある構造、分類の抽出）
  - 検索の高速化（クラスタ検索）
- LSI、COVによる特徴軸抽出
  - 主要なクラスタ（トピック）を見つける
  - 小さいクラスタの抽出は容易ではない

# LSI、COVによるクラスタ解析

- クラスタの重複を認める（良いか悪いかは不明）
- 一般的なクラスタリングは重複を認めない



# Ando's Rescaling Algorithm

- 入力
  - 文書-索引語行列 $A$
  - Scale Factor $q$
  - 射影する次元 $k$
- 出力　基底集合
  - $\{b_1, b_2, \cdots, b_k\}$
- 文書 $d_j$ の射影行列

$$\hat{d}_j = [b_1, b_2, b_3, \cdots, b_k]^T d_j$$

$R = A;$

for $(i = 1;\ i \le k;\ i{+}{+})$ {

$R_s = R;$

$R_s = \left[\ |r_1|^q r_1,\ |r_2|^q r_2,\ \cdots,\ |r_m|^q r_m\ \right]^T;$

$b_i =$ the first eigenvector of $R_s^T R_s;$

$R = R - R b_i b_i^T;$ (the residual matrix)

}

# Ando's Algorithmの問題点

- 大小すべてのクラスタを見つけることができない
- Scale Factor $q$ が大きいと固有値計算が不安定
  - 残差計算で行列から情報が大幅に減少する
- 基底ベクトルが常に直交にならない
  - 数値計算による誤差？
- 文書数が大きいと計算途中でメモリ不足が生じる
  - 初めはスパースな行列（メモリの消費は少ない）
  - 残差行列は密な行列（メモリの消費が大きくなる）

# Dynamic Rescaling of LSI

- 入力
  - 文書-索引語行列 $A$
  - Scale Factor $q$
  - 射影する次元 $k$
- 出力　基底集合
  - $\{b_1, b_2, \cdots, b_k\}$

$$q = \begin{cases} t_{\max}^{-1} & (t_{\max} > 1) \\ 1 + t_{\max} & (t_{\max} \approx 1) \\ 10^{t_{\max}^{-2}} & (t_{\max} < 1) \end{cases}$$

$R = A;$

for $(i = 1;\ i \le k;\ i{+}{+})$ {

$R_s = R = [\ r_1,\ r_2,\ \cdots,\ r_m\ ]^T;$

$t_{\max} = \max(|r_1|,\ |r_2|,\ \cdots,\ |r_m|);$

$q = \mathrm{func}(t_{\max});$

$R_s = [\ |r_1|^q r_1,\ |r_2|^q r_2,\ \cdots,\ |r_m|^q r_m\ ]^T;$

$\mathrm{SVD}(R_s);$ (singular value decomposition)

$\hat{b}_i =$ the first row vector of $V^T$;

$b_i = MGS(\hat{b}_i);$ (modified Gram - Schmidt)

$R = R - R b_i b_i^T;$ (the residual matrix)

}

# Dynamic Rescaling of COV

- 入力
  - 文書-索引語行列 $A$
  - Scale Factor $q$
  - 射影する次元 $k$
- 出力　基底集合
  - $\{b_1, b_2, \cdots, b_k\}$

$$q = \begin{cases} t_{\max}^{-1} & (t_{\max} > 1) \\ 1 + t_{\max} & (t_{\max} \approx 1) \\ 10^{t_{\max}^{-2}} & (t_{\max} < 1) \end{cases}$$

$R = A;$

for $(i = 1;\ i \le k;\ i{+}{+})$ {

$R_s = R = [\ r_1,\ r_2,\ \cdots,\ r_m\ ]^T;$

$t_{\max} = \max(\ |r_1|,\ |r_2|,\ \cdots,\ |r_m|\ );$

$q = \mathrm{func}(t_{\max});$

$R_s = [\ |r_1|^q r_1,\ |r_2|^q r_2,\ \cdots,\ |r_m|^q r_m\ ]^T;$

$C = COV(R_s);$ (covariance matrix)

$\mathrm{SVD}(C);$ (singular value decomposition)

$\hat{b}_i =$ the first row vector of $V^T;$

$b_i = MGS(\hat{b}_i);$ (modified Gram - Schmidt)

$R = R - Rb_i b_i^T;$ (the residual matrix)

}

# 実験1

- 小規模データ集合での実験（140文書、40ターム）
  - Clinton（25文書、Majorクラスタ）
    - Clinton+Gore（10文書）、Clinton+Hillary（10文書）
    - Clinton+Gore+Hillary（10文書） （いずれかは”5”の誤り）
  - Java（25文書、Majorクラスタ）
    - Java+JSP（10文書）、Java+Applet（5文書）、
    - Java+JSP+Applet（10文書）
  - Minorクラスタ（各5文書）
    - Bluetooth、Soccer、Matrix、DNA
  - noise 文書（70文書）
- 40次元から6次元に次元削減

# 実験1の結果

| Vector | LSI | COV | Ando | LSI-Rescale | COV-Rescale |
|---|---|---|---|---|---|
| b1 | C | C, J | J | J | C, J |
| b2 | J | C, J | C | C | noise, all-m, C, J |
| b3 | noise | noise, all-m | noise | noise | m, d |
| b4 | C | all-m, noise | b, s | M, D | all-m |
| b5 | J | all-m, noise | m, d | b, s | b, s |
| b6 | noise | all-m, noise | m, d | all-m | noise, all-m |

# 実験2

- 中規模データ集合での実験
  （10,000文書、500タターム）
  - 5 Majorクラスタ: M1, M2, ･･･, M5（各500文書）
  - 20 Minorクラスタ: m1, m2, ･･･, m20（各200文書）
  - 残りは noise (3,500文書)
- 500次元から25次元に次元削減

# 実験2の結果(1/3)

| Vector | LSI | COV | COV-Rescale |
|---|---|---|---|
| b1 | M1 | M3 | M1, M3 |
| b2 | M5 | M1 | M2, M4 |
| b3 | M4 | M5 | m12, m17 |
| b4 | M2 | M4 | m14, m17 |
| b5 | M3 | M2 | m4, m14 |
| b6 | M1 | M3 | m1, m14 |
| b7 | M5 | M1 | m11, m18 |
| b8 | M4 | M5 | m18 |
| b9 | M2 | M4 | M1, M3 |

# 実験2の結果(2/3)

| Vector | LSI | COV | COV-Rescale |
|---|---|---|---|
| b10 | M3 | M2 | M2, M4 |
| b11 | m4 | m8, m19 | m12, m17 |
| b12 | m2 | m7 | m14, m17 |
| b13 | M2 | m12, m16 | m4, m14 |
| b14 | m19 | m12, m16 | m1, m14 |
| b15 | m16 | m12, m13 | m11, m18 |
| b16 | m14 | m13, m15 | m18 |
| b17 | m12 | m3, m15 | m3, m11 |
| b18 | m5 | m3, m11 | m11, m9 |

# 実験2の結果(3/3)

| Vector | LSI | COV | COV-Rescale |
|---|---|---|---|
| b19 | m13 | m9, m11 | m9, m19 |
| b20 | m20 | m9, m14 | m5 |
| b21 | m17 | m5, m14 | m15 |
| b22 | m10 | m5, m20 | M5 |
| b23 | m7 | m1, m20 | M5 |
| b24 | m15 | m1, m17 | M4 |
| b25 | m3 | m2, m17 | noise |

# 実験2の考察

- 25個の基底ベクトルの計算時間（使用マシン不明）
  - LSI　　：0.96 秒
  - COV　　：1.05 秒
  - COV-Rescale　：　263 秒
- COV-Rescaleでは共分散行列の計算量が大きい
- COV-Rescaleは最も多い大小クラスタを抽出
  - LSI　　：19 (5 Major + 14 Minor)
  - COV　　：　21 (5 Major + 16 Minor)
  - COV-Rescale　：25 (all Major and Minor)

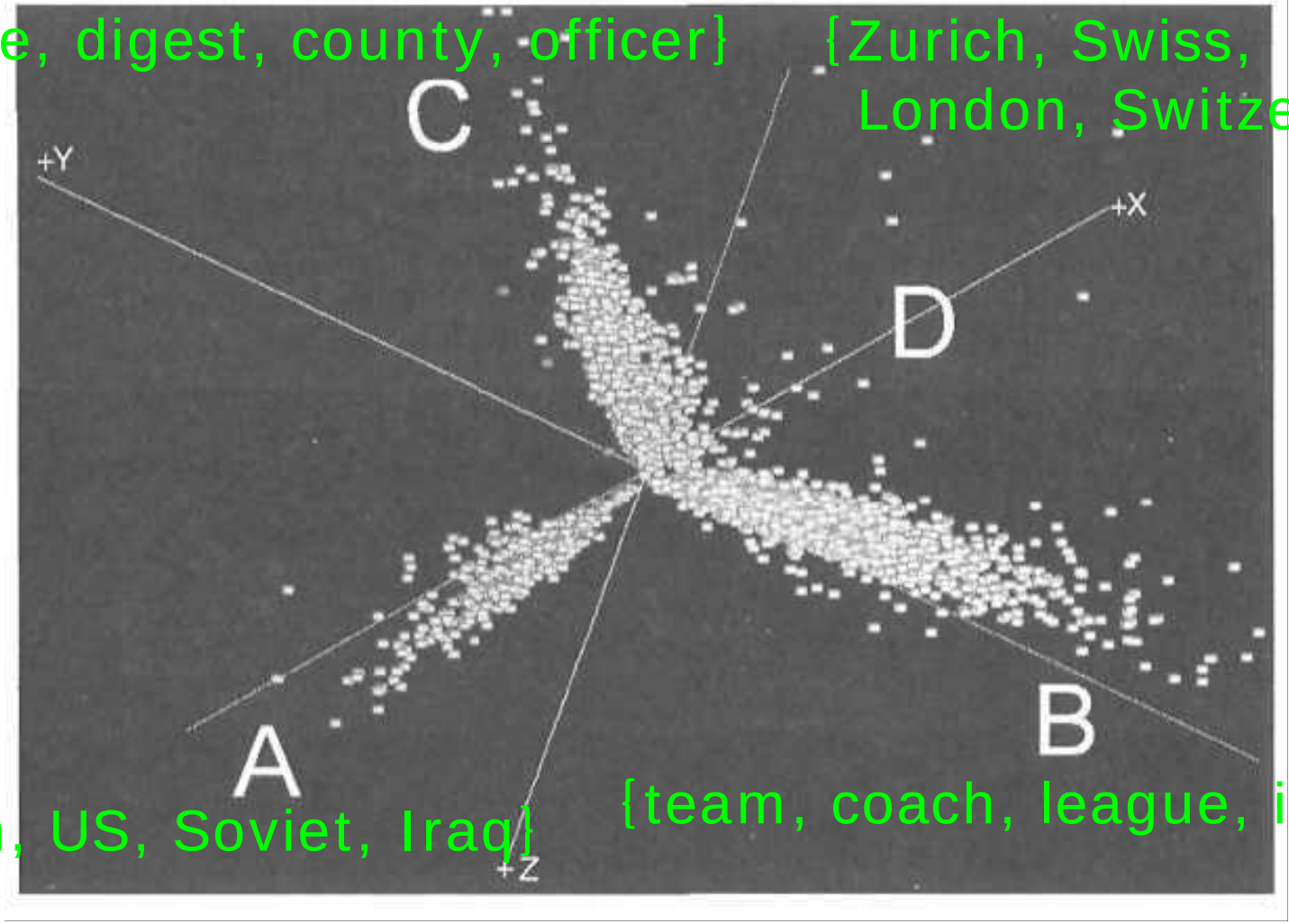
# 実験3

- 大規模データ集合での実験
  - L.A. Times の新聞記事
    （約10万文書、11,800ターム）
  - 文書ベクトル化への前処理
    - stemming (語幹変換)
    - stop words (不要語)、低頻度語の削除
    - TF-IDF による重み付け
  - COV-Rescaleを用いて次元削減
    - 11,800次元から200次元に削減

# 実験3の結果1

{police, digest, county, officer}

{Zurich, Swiss, London, Switzerland}

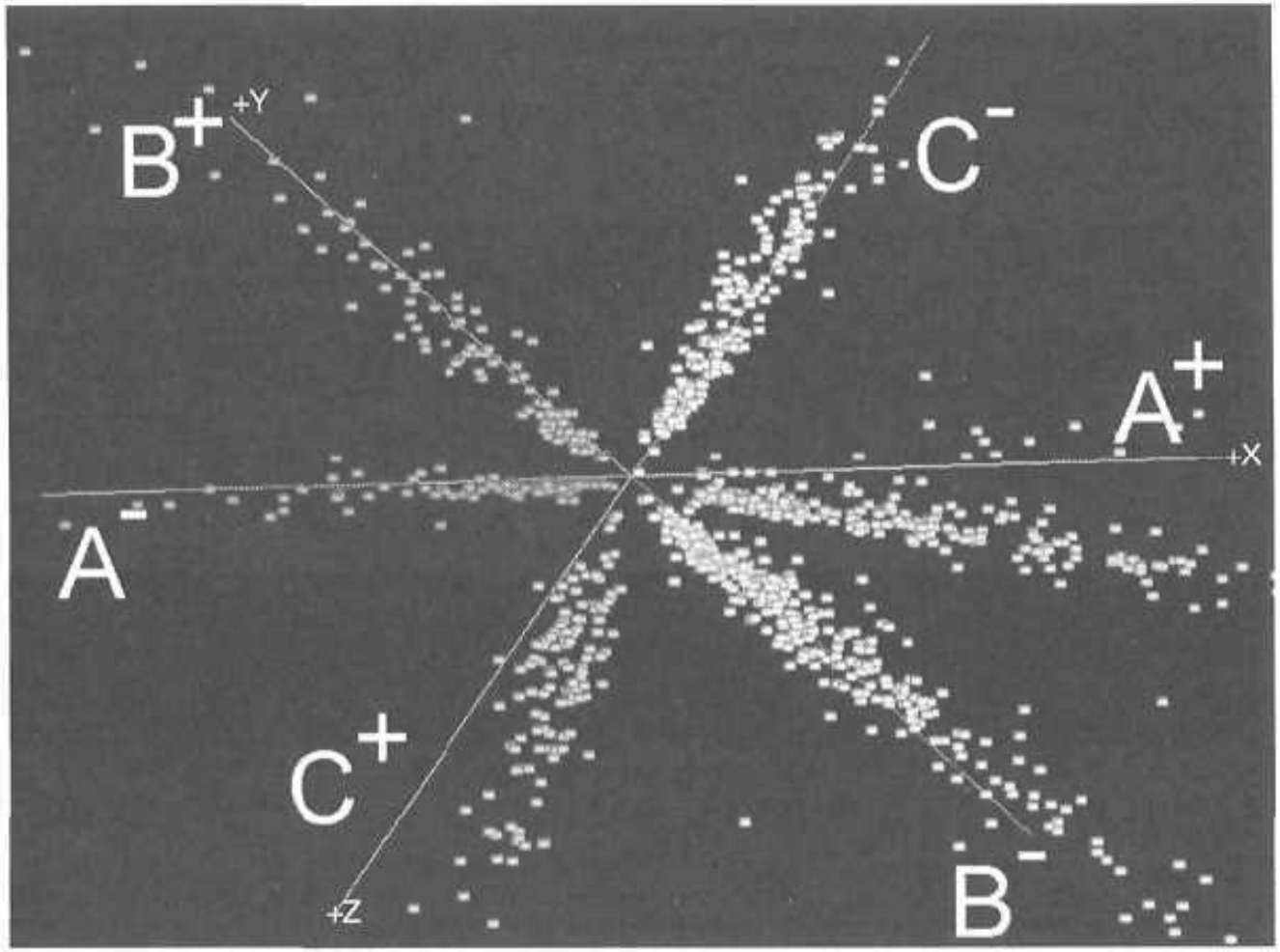



{Bush, US, Soviet, Iraq}

{team, coach, league, inning}

# 実験3の結果2

63th Basis




58th Basis

104th Basis

# まとめ

- 共分散行列を用いた特徴軸抽出手法の紹介
- 特徴
  - 高い拡張性
  - 大小クラスタを見つけることができる
  - クラスタ重複に対応
- 実験の結果、期待した結果が出力された
- 共分散による基底ベクトルの有効性
  - ベクトルの正と負で2つのクラスタを発見
  - SVDよりも少ない計算量で多くの特徴抽出が可能