AMSC808N/CMSC828V

CUR decomposition

Maria Cameron

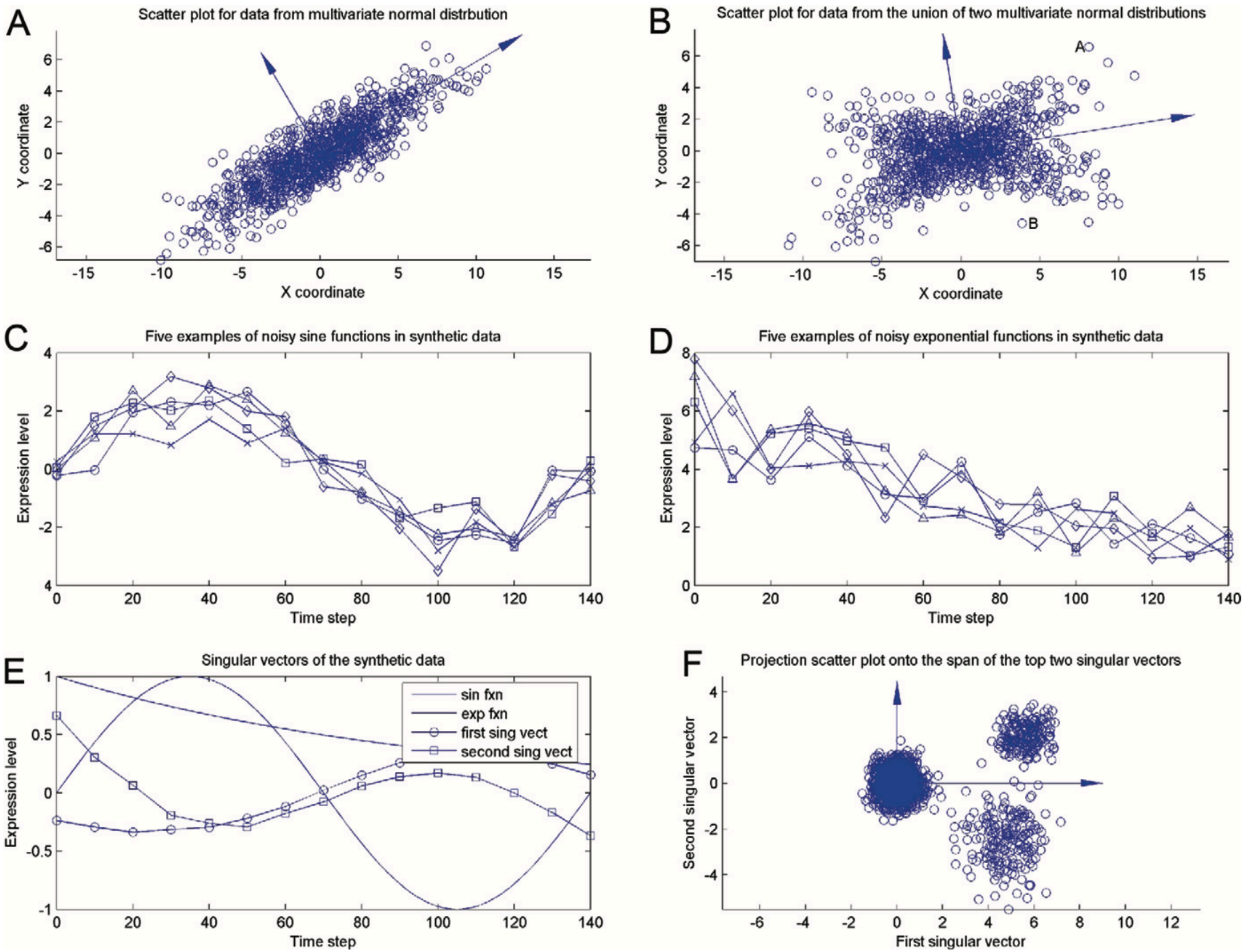
October 21, 2021

Motivation

- Need for low-rank approximation for a matrix.
- SVD $A = U\Sigma V^T$ gives the best approximation: $\| A - A_k \|_F \to min$, but columns of U and V are hard-to-interpret
- Columns often correspond to products or features, while rows correspond to users. A joking example from Mahoney and Drineas:

$$[(1/2)\text{age} - (1/\sqrt{2})\text{height} + (1/2)\text{income}],$$

Visual examples (Mahoney and Drineas, PNAS 2009)



Even more examples...

- Spearman, a social scientist interested in models for human intelligence, invented ***"factor analysis"***. Computed first principal component of a set of mental tests and reined it as an entity. He called it ***"the general intelligence factor"***. He called the subsequent principal component ***"group factors"***.

- Application of this analysis resulted in ranking individuals in single intelligence scale, dubious social applications of data analysis such as involuntary sterilization of imbeciles in Virginia. See S. J. Gould "Mismeasure of Man".

What do we want from a matrix decomposition

- Provable worst-case optimality and algorithmic properties
- Should have a natural statistical interpretation associated with its construction
- Should perform well in practice

CUR decompositions

- G. W. Stewart: quasi-Gram-Schmidt method, applied to A and A^T (1999, 2004)
- Goreinov, Tyrtyshnikov, Zamarashkin (1997): CUR with choice of columns related to max uncorrelatedness.
- Frieze, Kannan, Vempala (2004): random sampling of columns according to a probability distribution that depended on columns Euclidean norm. Worst-case scenario guarantee: $||A - P_C A||_F \leq ||A - A_k||_F + \varepsilon ||A||_F$ with high probability.
- Drineas, Kannan, Mahoney (2006): CUR, columns and rows chosen simultaneously based on their Euclidean norm. Worst-case scenario guarantee: $||A - CUR||_F \leq ||A - A_k||_F + \varepsilon ||A||_F$ with high probability.
- Drineas, Mahoney, Muthukrishnan (2008 SIAM J Matr Anal Appl): CUR based on ***leverage scores***: $||A - P_C A||_F \leq (1 + \varepsilon/2)||A - A_k||_F$
- Mahoney and Drineas (PNAS, 2009) “CUR matrix decomposition for improved data analysis”: $||A - CUR||_F \leq (2 + \varepsilon)||A - A_k||_F$

Leverage scores

S. Chatterjee and A. S. Hadi (1986)

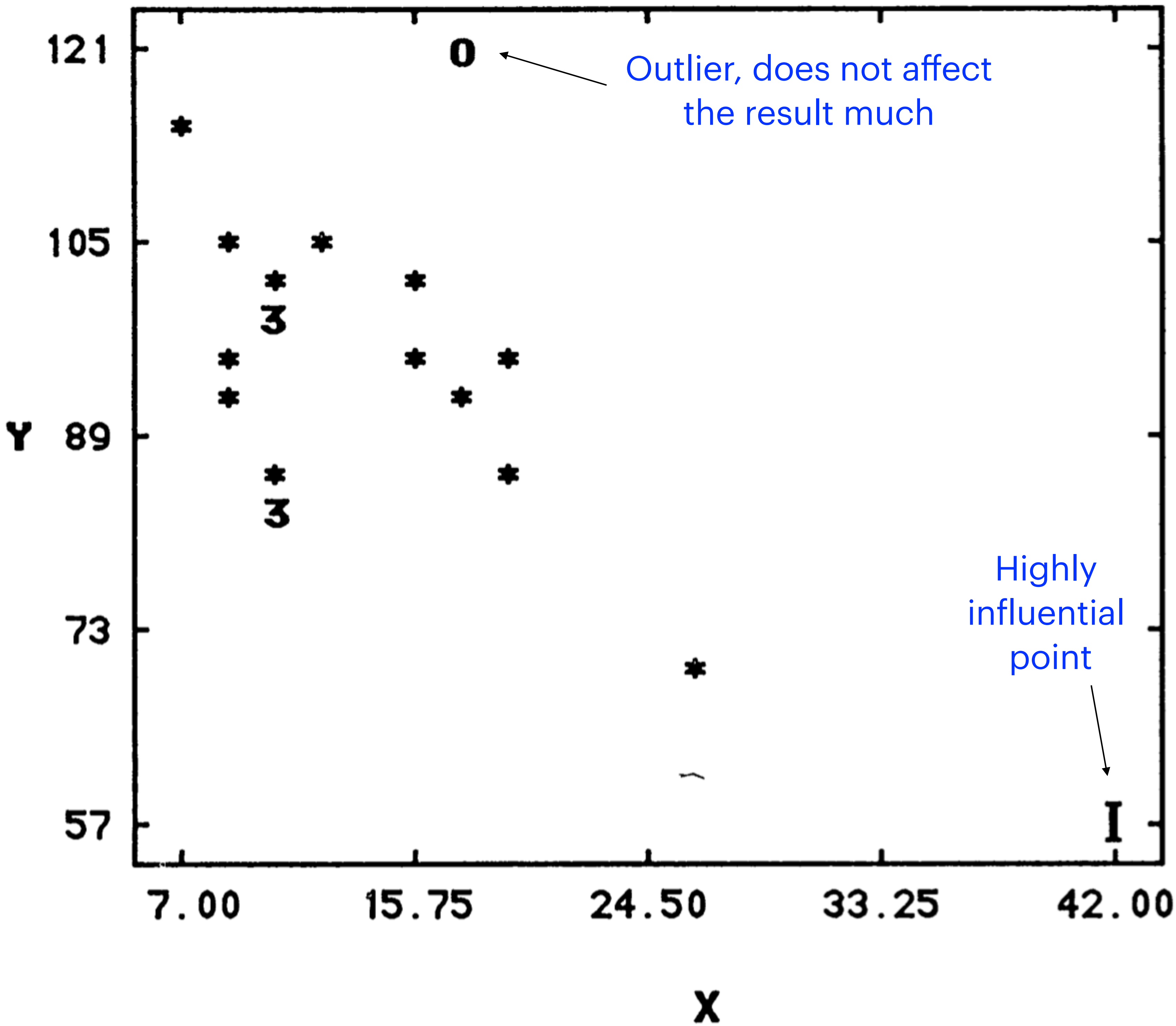
Linear regression model

$$Y = X\beta + \epsilon, \quad Y \in \mathbb{R},\ X \in \mathbb{R}^{n\times d},\ \beta \in \mathbb{R}^d$$

ϵ is a random variable, mean 0, variance σ^2

$$\hat{\beta} = (X^T X)^{-1} X^T Y,$$
$$\mathrm{Var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1},$$
$$\hat{Y} = X\hat{\beta} = PY,$$

$$P = X(X^T X)^{-1} X^T,$$
$$\mathrm{Var}(\hat{Y}) = \sigma^2 P,$$
$$e = Y - \hat{Y} = (I - P)Y,$$
$$\mathrm{Var}(e) = \sigma^2 (I - P),$$


121
105
89
73
57
Y
7.00
15.75
24.50
33.25
42.00
X
O
3
3
I
Outlier, does not affect the result much
Highly influential point

Leverage scores

$$\hat{Y} = PY = X(X^\top X)^{-1} X^\top Y$$

Diagonal entries $\quad x_i (X^\top X)^{-1} x_i^\top$

can be though of as the amount of leverage of

$$y_i \quad \text{on} \quad \hat{y}_i$$

CUR Algorithm

Mahoney and Drineas, PNAS 2009

$$A^j = \sum_{\xi=1}^{r} (\sigma_\xi u^\xi) v_j^\xi,$$

Since we seek columns of A that are simultaneously correlated with the span of all top k right singular vectors, we then compute the *normalized statistical leverage scores*:

$$\pi_j = \frac{1}{k} \sum_{\xi=1}^{k} \left(v_j^\xi\right)^2, \qquad \textbf{[3]}$$

for all $j = 1, \ldots, n$. With this normalization, it is straightforward to show that $\pi_j \geq 0$ and that $\sum_{j=1}^{n} \pi_j = 1$, and thus that these scores form a probability distribution over the n columns.

ColumnSelect(A,k,epsilon)

1. Compute $v^1, \ldots, v^k$ (the top k right singular vectors of A) and the normalized statistical leverage scores of Eq. **3**.
2. Keep the jth column of A with probability $p_j = \min\{1, c\pi_j\}$, for all $j \in \{1, \ldots, n\}$, where $c = O(k \log k/\epsilon^2)$.
3. Return the matrix C consisting of the selected columns of A.

AlgorithmCUR(A,k,epsilon)

1. Run ColumnSelect on A with $c = O(k \log k/\epsilon^2)$ to choose columns of A and construct the matrix C.
2. Run ColumnSelect on A^T with $r = O(k \log k/\epsilon^2)$ to choose rows of A (columns of A^T) and construct the matrix R.
3. Define the matrix U as $U = C^+ A R^+$, where X^+ denotes a Moore–Penrose generalized inverse of the matrix X (17).

$$\|A - CUR\|_F \leq (2 + \epsilon) \|A - A_k\|_F ,$$

Drineas, Mahoney, Muthukrishnan (2008)

Two variants of algorithm for selecting columns

Exactly(c)

Data : $A \in \mathbb{R}^{m \times n}$, $p_i \geq 0, i \in [n]$ s.t. $\sum_{i \in [n]} p_i = 1$, positive integer $c \leq n$.

Result : Sampling matrix S, rescaling matrix D, and sampled and rescaled columns C.

Initialize S and D to the all zeros matrices.

for $t = 1, \ldots, c$ **do**

  Pick $i_t \in [n]$, where $\mathbf{Pr}(i_t = i) = p_i$;

  $S_{i_t t} = 1$;

  $D_{tt} = 1/\sqrt{cp_{i_t}}$.

end

$C = ASD$.

Algorithm 4. The EXACTLY(c) algorithm to create S, D, and C.

Expected(c)

Data : $A \in \mathbb{R}^{m \times n}$, $p_i \geq 0, i \in [n]$ s.t. $\sum_{i \in [n]} p_i = 1$, positive integer $c \leq n$.

Result : Sampling matrix S, rescaling matrix D, and sampled and rescaled columns C.

Initialize S and D to the all zeros matrices.

$t = 1$;

for $j = 1, \ldots, n$ **do**

  Pick j with probability $\min\{1, cp_j\}$;

  if *j is picked* **then**

    $S_{jt} = 1$;

    $D_{tt} = 1/\min\{1, \sqrt{cp_j}\}$;

    $t = t + 1$;

  end

end

$C = ASD$.

Algorithm 5. *The* Expected(c) *algorithm to create S, D, and C.*

THEOREM 4. *Let $A \in \mathbb{R}^{m \times n}$, let $C \in \mathbb{R}^{m \times c}$ be a matrix consisting of any c columns of A, and let $\epsilon \in (0, 1]$. If we set $r = 3200c^2/\epsilon^2$ and run Algorithm 2 by choosing r rows exactly from A and from C with the* EXACTLY(c) *algorithm, then with probability at least* 0.7

$$\|A - CUR\|_F \leq (1 + \epsilon) \left\|A - CC^+A\right\|_F . \tag{18}$$

Similarly, if we set $r = O(c \log c/\epsilon^2)$ and run Algorithm 2 by choosing no more than r rows in expectation from A and from C with the EXPECTED(c) *algorithm, then* (18) *holds with probability at least* 0.7.

$$\|A - P_C A\|_F \leq (1 + \epsilon/2) \|A - A_k\|_F$$

$$\|A - CUR\|_F = \left\|A - CC^+AR^+R\right\|_F$$

$$\begin{aligned}
\|A - CUR\|_F &\leq \left\|A - CC^+A\right\|_F + \left\|CC^+A - CC^+AR^+R\right\|_F \\
&\leq \left\|A - CC^+A\right\|_F + \left\|A - AR^+R\right\|_F \\
&= \|A - P_C A\|_F + \|A - AP_R\|_F .
\end{aligned}$$

$$\|A - CUR\|_F \leq (2 + \epsilon) \|A - A_k\|_F$$

Experiment with text categorization data