Clustering on Multiple Incomplete Datasets via Collective Kernel Learning

Weixiang Shao, Xiaoxiao Shi, Philip S. Yu

2013 IEEE 13th International Conference on Data Mining, Page 1181 - 1186

杜偲豪 *D10207305* 林建宏 *D10407007*

詹鈞揚 *M10401001* 王澤浩 *0034980*

2015.12.09

Introduction

	Name	Age	Country
User1	Bob	31	USA
User2	Angeli	21	USA
User3	?	?	?
User4	Tom	40	Canada

(a) User profile. User3 doesn't complete her profile.

	facebook	twitter	Google+
User1	✔	✖	✖
User2	?	?	?
User3	✔	✖	✔
User4	?	?	?

(b) Browsing behaviors. User2 and User4 are new users, thus no browsing behaviors are available.

	Credit Score
User1	Fair
User2	Good
User3	?
User4	?

(c) Credit scores. No credit scores for User3 and User4.

Fig. 1: Different datasets for grouping the users in the recommendation systems.

- Traditional Method: Prediction
- Method in Paper: CoKL & KCCA
  - Collective Kernel Learning
  - Kernel Canonical Correlation Analysis

CoKL & KCCA

- CoKL:

  Complete the kernel matrices.

- KCCA:

  Based on the completed kernel matrices, KCCA could find the projections that maximize the correlations between the datasets.

Problem Formulation

Derive solution S

based on dataset X and Y

$$X = \begin{pmatrix} X_c \\ X_{m1} \\ X_{m2} \end{pmatrix} \qquad Y = \begin{pmatrix} Y_c \\ Y_{m1} \\ Y_{m2} \end{pmatrix}$$

$$X = \begin{pmatrix} X_c \\ X_{m1} \\ X_{m2} = ? \end{pmatrix} \qquad Y = \begin{pmatrix} Y_c \\ Y_{m1} = ? \\ Y_{m2} \end{pmatrix}$$

? : missing values

Only appear in Y

Only appear in X

$$X = \begin{pmatrix} X_c \\ X_{m1} \\ X_{m2} = ? \end{pmatrix} \qquad Y = \begin{pmatrix} Y_c \\ Y_{m1} = ? \\ Y_{m2} \end{pmatrix}$$

? : missing values

Kernel matrix

Kx = XX' = [Xc Xm1 NA]' [Xc Xm1 NA]
Ky = YY' = [Yc NA Ym2]' [Yc NA Ym2]

$$K_x = \begin{pmatrix} K_x^{cc} & K_x^{cm_1} & K_x^{cm_2} =? \\ (K_x^{cm_1})^T & K_x^{m_1m_1} & K_x^{m_1m_2} =? \\ (K_x^{cm_2})^T =? & (K_x^{m_1m_2})^T =? & K_x^{m_2m_2} =? \end{pmatrix}$$
$$K_y = \begin{pmatrix} K_y^{cc} & K_y^{cm_1} =? & K_y^{cm_2} \\ (K_y^{cm_1})^T =? & K_y^{m_1m_1} =? & K_y^{m_1m_2} =? \\ (K_y^{cm_2})^T & (K_y^{m_1m_2})^T =? & K_y^{m_2m_2} \end{pmatrix} \quad (1)$$

Laplacian regularization

- To build the full kernel matrix Kx and Ky, we borrow the idea from Laplacian regularization.
- Generate the graph Laplacian. Lx = Dx − Kx

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \cdots \\ 0 & 0 & \lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

- tr(LxKy) reflects the “inconsistence” of the kernel matrix Ky, minimize it.

Minimize inconsistence

$$A = \begin{pmatrix} A_c \\ A_{m_1} \\ A_{m_2} \end{pmatrix}$$

$$\min_{K_y \succeq 0} tr(\mathcal{L}_x K_y) = \min_{A} tr\left(\mathcal{L}_x A A^T\right)$$

$$= \min_{A_{m_1}} 2\, tr(A_{m_1} A_c^T \mathcal{L}_x^{cm_1}) + 2\, tr(A_{m_2} A_c^T \mathcal{L}_x^{cm_2}) +$$

$$2\, tr(A_{m_2} A_{m_1}^T \mathcal{L}_x^{m_1 m_2}) + tr(A_{m_1} A_{m_1}^T \mathcal{L}_x^{m_1 m_1})$$

- Taking derivative w.r.t. A_{m_1} , setting it to zero and solving it.

$$A_{m_1} = -(\mathcal{L}_x^{m_1 m_1})^{-1} \left((\mathcal{L}_x^{cm_1})^T A_c - \mathcal{L}_x^{m_1 m_2} A_{m_2}\right)$$

New kernel matrix

- Set Ac = Yc ; Am2 = Ym2

$$A = \begin{pmatrix} A_c \\ -(\mathcal{L}_x^{m_1 m_1})^{-1} \left((\mathcal{L}_x^{c m_1})^T A_c - \mathcal{L}_x^{m_1 m_2} A_{m_2} \right) \\ A_{m_2} \end{pmatrix}$$

- Ky = AA'

$$K_y = \begin{pmatrix} K_y^{cc} & K_y^{cm_1} & K_y^{cm_2} \\ (K_y^{cm_1})^T & K_y^{m_1 m_1} & K_y^{m_1 m_2} \\ (K_y^{cm_2})^T & (K_y^{m_1 m_2})^T & K_y^{m_2 m_2} \end{pmatrix}$$

V=IR
P=IV
ΔE=hν
v=fλ
ΔP=ρgΔh
v=ωr
E=mc²
sin²+cos²=1
ω=2πf
y=x²+a
PV=nRT
2H₂+O₂ ⇌ 2H₂O
sinα/a = sinβ/b = sinγ/c
CH₄+2O₂ → CO₂+2H₂O

Algorithms

Algorithm 1 - CoKL

Algorithm 1 Collective Kernel Learning (CoKL)

Input: Incomplete Datasets X and Y
Output: The full kernel matrices K_x and K_y

Give initial values to the missing features in the two dataset.
Calculate the kernel matrices K_x and K_y.
$\mathcal{L}_x \leftarrow D_x - K_x$
$\mathcal{L}_y \leftarrow D_Y - K_y$
repeat

$$A \leftarrow \begin{pmatrix} A_c \\ -(\mathcal{L}_x^{m_1 m_1})^{-1} \left((\mathcal{L}_x^{c m_1})^T A_c - \mathcal{L}_x^{m_1 m_2} A_{m_2} \right) \\ A_{m_2} \end{pmatrix}$$

Calculate the new full kernel matrix K'_y using A.

$$B \leftarrow \begin{pmatrix} B_c \\ B_{m_1} \\ -(\mathcal{L}_y^{m_2 m_2})^{-1} \left((\mathcal{L}_y^{c m_2})^T B_c - \mathcal{L}_y^{m_2 m_1} B_{m_1} \right) \end{pmatrix}$$

Calculate the new full kernel matrix K'_x using B.
$K_x \leftarrow K'_x$.
$K_y \leftarrow K'_y$.
until Convergence

Algorithm 2 - CoKL + KCCA

Algorithm 2 Clustering using Collective Kernel Learning and KCCA

Input: Incomplete Datasets X and Y.
Output: The clustering solution $\mathcal{S}$.

$[K_x, K_y] = CoKL(X, Y)$.
$[X_p, Y_p] = KCCA(K_x, K_y)$. $\{X_p$ and Y_p are the projected datasets.$\}$
if The feature space is still too large **then**
  $X_p = PCA(X_p)$.
  $Y_p = PCA(Y_p)$.
end if
Apply k-means to the projected datasets X_p and Y_p.

Experiments

Step I - Randomly split the feature set into two disjoint part.

area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
17.63	15.98	0.8673	6.191	3.561	4.0760	6.060	2
13.07	13.92	0.8480	5.472	2.994	5.3040	5.395	3

dataset: UCI seeds

Step I - Randomly split the feature set into two disjoint part.

area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
17.63	15.98	0.8673	6.191	3.561	4.0760	6.060	2
13.07	13.92	0.8480	5.472	2.994	5.3040	5.395	3

dataset: UCI seeds

dataset X

dataset Y

Class (ignore)

Step II - Randomly delete the instances in both of the datasets.

area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
15.26	**14.84**	**0.8710**	**5.763**	**3.312**	**2.221**	**5.220**	1
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
17.63	15.98	0.8673	6.191	3.561	4.0760	6.060	2
13.07	13.92	0.8480	5.472	2.994	5.3040	5.395	3

dataset: UCI seeds

Case 1

Present in both datasets X and Y

$$C = \{(x_1, y_1), \ldots, (x_c, y_c)\}$$

Step II - Randomly delete the instances in both of the datasets.

area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
14.88	**14.57**	**0.8811**	**5.554**	?	?	?	1
17.63	15.98	0.8673	6.191	3.561	4.0760	6.060	2
13.07	13.92	0.8480	5.472	2.994	5.3040	5.395	3

dataset: UCI seeds

Case 2

Only present in dataset X

$$M_1 = \{x_{c+1}, \ldots, x_{c+m_1}\}$$

Step II - Randomly delete the instances in both of the datasets.

area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
?	?	?	?	3.561	4.0760	6.060	2
13.07	13.92	0.8480	5.472	2.994	5.3040	5.395	3

dataset: UCI seeds

Case 3

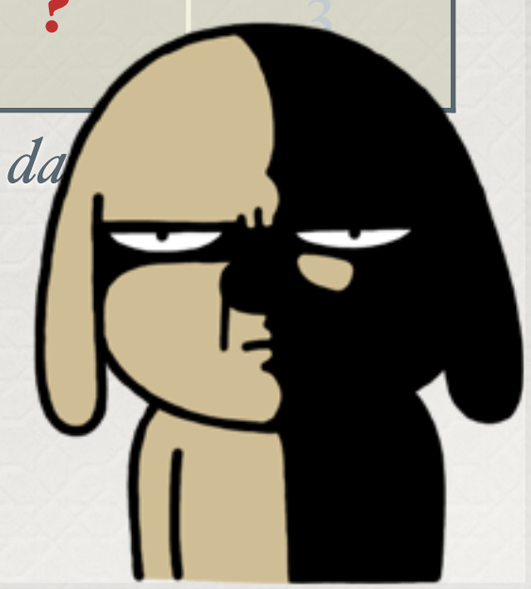
Only present in dataset Y

$$M_2 = \{y_{c+m_1+1}, \ldots, y_{c+m_1+m_2}\}$$

Step II - Randomly delete the instances in both of the datasets.

area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
17.63	15.98	0.8673	6.191	3.561	4.0760	6.060	2
?	?	?	?	?	?	?	?

da



This case NOT exists

We increase to 7 rows datasets, then repeat step I & II.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	?	?	?	1
6	13.32	13.94	0.8613	5.541	?	?	?	3
7	?	?	?	?	3.337	2.699	4.825	1

dataset: UCI seeds

We increase to 7 rows datasets, then repeat step I & II.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	NA	NA	NA	1
6	13.32	13.94	0.8613	5.541	NA	NA	NA	3
7	NA	NA	NA	NA	3.337	2.699	4.825	1

dataset: UCI seeds

$C = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$; $c = 4$

$M_1 = \{x_5, x_6\}$; $m_1 = 2$

$M_2 = \{y_7\}$; $m_2 = 1$

Step III - Fill the missing value by average of each column.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	**3.338**	**3.796**	**5.476**	1
6	13.32	13.94	0.8613	5.541	**3.338**	**3.796**	**5.476**	3
7	**15.17**	**14.82**	**0.8652**	**5.754**	3.337	2.699	4.825	1

dataset: UCI seeds

$C = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$; $c = 4$

$M_1 = \{x_5, x_6\}$; $m_1 = 2$

$M_2 = \{y_7\}$; $m_2 = 1$

Step III - Fill the missing value by average of each column.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	**3.338**	**3.796**	**5.476**	1
6	13.32	13.94	0.8613	5.541	**3.338**	**3.796**	**5.476**	3
7	**15.17**	**14.82**	**0.8652**	**5.754**	3.337	2.699	4.825	1

X_c

dataset: UCI seeds

$C = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$; $c = 4$

$M_1 = \{x_5, x_6\}$; $m_1 = 2$

$M_2 = \{y_7\}$; $m_2 = 1$

Step III - Fill the missing value by average of each column.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	**3.338**	**3.796**	**5.476**	1
6	13.32	13.94	0.8613	5.541	**3.338**	**3.796**	**5.476**	3
7	**15.17**	**14.82**	**0.8652**	**5.754**	3.337	2.699	4.825	1

X_c Y_c

dataset: UCI seeds

$C = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$; $c = 4$

$M_1 = \{x_5, x_6\}$; $m_1 = 2$

$M_2 = \{y_7\}$; $m_2 = 1$

Step III - Fill the missing value by average of each column.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	3.338	3.796	5.476	1
6	13.32	13.94	0.8613	5.541	3.338	3.796	5.476	3
7	15.17	14.82	0.8652	5.754	3.337	2.699	4.825	1

X_c Y_c X_{m1} Y_{m1}

dataset: UCI seeds

$C = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$; $c = 4$

$M_1 = \{x_5, x_6\}$; $m_1 = 2$

$M_2 = \{y_7\}$; $m_2 = 1$

Step III - Fill the missing value by average of each column.

	area	perimeter	compactness	length of kernel	width of kernel	asymmetry coefficient	length of kernel groove	class of wheat
1	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
2	17.63	15.98	0.8673	6.191	3.561	4.076	6.060	2
3	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
4	13.07	13.92	0.8480	5.472	2.994	5.304	5.395	3
5	14.88	14.57	0.8811	5.554	3.338	3.796	5.476	1
6	13.32	13.94	0.8613	5.541	3.338	3.796	5.476	3
7	15.17	14.82	0.8652	5.754	3.337	[illegible]	4.825	1

X_c Y_c X_{m1} Y_{m1} X_{m2} Y_{m2}

dataset: UCI seeds

$C = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$; $c = 4$

$M_1 = \{x_5, x_6\}$; $m_1 = 2$

$M_2 = \{y_7\}$; $m_2 = 1$

Step IV - Get initial kernel matrix - Kx & Ky.

	1	2	3	4	5	6	7
1	487.0640	542.6112	524.8387	438.2947	476.0627	442.8158	485.2812
2	542.6112	605.2580	585.1773	487.4783	530.3120	492.6441	540.5801
3	524.8387	585.1773	565.8541	471.7775	512.9638	476.7262	522.8896
4	438.2947	487.4783	471.7775	395.2532	428.4347	399.1879	436.7377
5	476.0627	530.3120	512.9638	428.4347	465.3226	432.8410	474.3228
6	442.8158	492.6441	476.7262	399.1879	432.8410	403.1905	441.2343
7	485.2812	540.5801	522.8896	436.7377	474.3228	441.2343	483.5076

Kx = XX' — X is dataset X

Step IV - Get initial kernel matrix - Kx & Ky.

	1	2	3	4	5	6	7
1	487.0640	542.6112	524.8387	438.2947	476.0627	442.8158	485.2812
2	542.6112	605.2580	585.1773	487.4783	530.3120	492.6441	540.5801
3	524.8387	585.1773	565.8541	471.7775	512.9638	476.7262	522.8896
4	438.2947	487.4783	471.7775	395.2532	428.4347	399.1879	436.7377
5	476.0627	530.3120	512.9638	428.4347	465.3226	432.8410	474.3228
6	442.8158	492.6441	476.7262	399.1879	432.8410	403.1905	441.2343
7	485.2812	540.5801	522.8896	436.7377	474.3228	441.2343	483.5370

K_x^{cc} K_x^{cm1} K_x^{cm2}

$K_x^{cm1'}$ K_x^{m1m1} K_x^{m1m2}

$K_x^{cm2'}$ $K_x^{m1m2'}$ K_x^{m2m2}

Kx = XX' — ***X is dataset X***

Step IV - Get initial kernel matrix - Kx & Ky.

	1	2	3	4	5	6	7
1	487.0640	542.6112	524.8387	438.2947	476.0627	442.8158	485.2812
2	542.6112	605.2580	585.1773	487.4783	530.3120	492.6441	540.5801
3	524.8387	585.1773	565.8541	471.7775	512.9638	476.7262	522.8896
4	438.2947	487.4783	471.7775	395.2532	428.4347	399.1879	436.7377
5	476.0627	530.3120	512.9638	428.4347	465.3226	432.8410	474.3228
6	442.8158	492.6441	476.7262	399.1879	432.8410	403.1905	441.2343
7	485.2812	540.5801	522.8896	436.7377	474.3228	441.2343	483.5976

K_x^{cm2}

K_x^{m1m2}

$K_x^{cm2'}$

$K_x^{m1m2'}$

K_x^{m2m2}

Kx = XX' — ***X is dataset X***

Step IV - Get initial kernel matrix - Kx & Ky.

	1	2	3	4	5	6	7
1	43.15058	52.48003	52.60012	49.85821	48.06441	48.06441	42.23312
2	52.48003	66.01810	67.07644	64.97444	60.53454	60.53454	52.12368
3	52.60012	67.07644	68.53301	66.93371	61.54875	61.54875	52.60046
4	49.85821	64.97444	66.93371	66.20248	59.66124	59.66124	50.33735
5	48.06441	60.53454	61.54875	59.66124	55.52160	55.52160	47.79908
6	48.06441	60.53454	61.54875	59.66124	55.52160	55.52160	47.79908
7	42.23312	52.12368	52.60046	50.33735	47.79908	47.79908	41.70079

Ky = YY' — Y is dataset Y

Step IV - Get initial kernel matrix - Kx & Ky.

	1	2	3	4	5	6	7
1	43.15058	52.48003	52.60012	49.85821	48.06441	48.06441	42.23312
2	52.48003	66.01810	67.07644	64.97444	60.53454	60.53454	52.12368
3	52.60012	67.07644	68.53301	66.93371	61.54875	61.54875	52.60046
4	49.85821	64.97444	66.93371	66.20248	59.66124	59.66124	50.33735
5	48.06441	60.53454	61.54875	59.66124	55.52160	55.52160	47.79908
6	48.06441	60.53454	61.54875	59.66124	55.52160	55.52160	47.79908
7	42.23312	52.12368	52.60046	50.33735	47.79908	47.79908	47.79908

K_y^{cc} K_y^{cm1} K_y^{cm2}

$K_y^{cm1'}$ K_y^{m1m1} K_y^{m1m2}

$K_y^{cm2'}$ $K_y^{m1m2'}$ K_y^{m2m2}

$K_y = YY'$ — Y *is dataset* Y

Step IV - Get initial kernel matrix - Kx & Ky.

	1	2	3	4	5	6	7
1	43.15058	52.48003	52.60012	49.85821	48.06441	48.06441	42.23312
2	52.48003	66.01810	67.07644	64.97444	60.53454	60.53454	52.12368
3	52.60012	67.07644	68.53301	66.93371	61.54875	61.54875	52.60046
4	49.85821	64.97444	66.93371	66.20248	59.66124	59.66124	50.33735
5	48.06441	60.53454	61.54875	59.66124	55.52160	55.52160	47.79908
6	48.06441	60.53454	61.54875	59.66124	55.52160	55.52160	47.79908
7	42.23312	52.12368	52.60046	50.33735	47.79908	47.79908	41.70079

K_y^{cm1}

$K_y^{cm1'}$

K_y^{m1m1}

K_y^{m1m2}

$K_y^{m1m2'}$

$Ky = YY'$ — *Y is dataset Y*

Step V - Generate the graph Laplacian - Lx & Ly.

	1	2	3	4	5	6	7
1	3396.968	0	0	0	0	0	0
2	0	3784.061	0	0	0	0	0
3	0	0	3660.227	0	0	0	0
4	0	0	0	3057.164	0	0	0
5	0	0	0	0	3320.26	0	0
6	0	0	0	0	0	3088.64	0
7	0	0	0	0	0	0	3384.553

Lx = Dx - Kx — Dx is the diagonal matrix consisting of the row sums of Kx

Step V - Generate the graph Laplacian - Lx & Ly.

	1	2	3	4	5	6	7
1	2909.9043	-542.6112	-524.8387	-438.2947	-476.0627	-442.8158	-485.2812
2	-542.6112	3178.8030	-585.1773	-487.4783	-530.3120	-492.6441	-540.5801
3	-524.8387	-585.1773	3094.3731	-471.7775	-512.9638	-476.7262	-522.8896
4	-438.2947	-487.4783	-471.7775	2661.9109	-428.4347	-399.1879	-436.7377
5	-476.0627	-530.3120	-512.9638	-428.4347	2854.9369	-432.8410	-474.3228
6	-442.8158	-492.6441	-476.7262	-399.1879	-432.8410	2685.4493	-441.2343
7	-485.2812	-540.5801	-522.8896	-436.7377	-474.3228	-441.2343	2901.0457

$$Lx = Dx - Kx$$

Step V - Generate the graph Laplacian - Lx & Ly.

	1	2	3	4	5	6	7
1	2909.9043	-542.6112	-524.8387	-438.2947	-476.0627	-442.8158	-485.2812
2	-542.6112	3178.8030	-585.1773	-487.4783	-530.3120	-492.6441	-540.5801
3	-524.8387	-585.1773	3094.3731	-471.7775	-512.9638	-476.7262	-522.8896
4	-438.2947	-487.4783	-471.7775	2661.9109	-428.4347	-399.1879	-436.7377
5	-476.0627	-530.3120	-512.9638	-428.4347	2854.9369	-432.8410	-474.3228
6	-442.8158	-492.6441	-476.7262	-399.1879	-432.8410	2685.4493	-441.2343
7	-485.2812	-540.5801	-522.8896	-436.7377	-474.3228	-441.2343	2901.1457

L_x^{cc} L_x^{cm1} L_x^{cm2}

$L_x^{cm1'}$ L_x^{m1m1} L_x^{m1m2}

$L_x^{cm2'}$ $L_x^{m1m2'}$ L_x^{m2m2}

$$Lx = Dx - Kx$$

Step V - Generate the graph Laplacian - Lx & Ly.

	1	2	3	4	5	6	7
1	336.4509	0	0	0	0	0	0
2	0	423.7418	0	0	0	0	0
3	0	0	430.8412	0	0	0	0
4	0	0	0	417.6287	0	0	0
5	0	0	0	0	388.6512	0	0
6	0	0	0	0	0	388.6512	0
7	0	0	0	0	0	0	334.5936

Ly = Dy - Ky — Dy is the diagonal matrix consisting of the row sums of Ky

Step V - Generate the graph Laplacian - Lx & Ly.

	1	2	3	4	5	6	7
1	293.30031	-52.48003	-52.60012	-49.85821	-48.06441	-48.06441	-42.23312
2	-52.48003	357.72367	-67.07644	-64.97444	-60.53454	-60.53454	-52.12368
3	-52.60012	-67.07644	362.30823	-66.93371	-61.54875	-61.54875	-52.60046
4	-49.85821	-64.97444	-66.93371	351.42618	-59.66124	-59.66124	-50.33735
5	-48.06441	-60.53454	-61.54875	-59.66124	333.12962	-55.52160	-47.79908
6	-48.06441	-60.53454	-61.54875	-59.66124	-55.52160	333.12962	-47.79908
7	-42.23312	-52.12368	-52.60046	-50.33735	-47.79908	-47.79908	292.89277

$$Ly = Dy - Ky$$

Step V - Generate the graph Laplacian - Lx & Ly.

	1	2	3	4	5	6	7
1	293.30031	-52.48003	-52.60012	-49.85821	-48.06441	-48.06441	-42.23312
2	-52.48003	357.72367	-67.07644	-64.97444	-60.53454	-60.53454	-52.12368
3	-52.60012	-67.07644	362.30823	-66.93371	-61.54875	-61.54875	-52.60046
4	-49.85821	-64.97444	-66.93371	351.42618	-59.66124	-59.66124	-50.33735
5	-48.06441	-60.53454	-61.54875	-59.66124	333.12962	-55.52160	-47.79908
6	-48.06441	-60.53454	-61.54875	-59.66124	-55.52160	333.12962	-47.79908
7	-42.23312	-52.12368	-52.60046	-50.33735	-47.79908	-47.79908	291.84277

L_y^{cc} L_y^{cm1} L_y^{cm2}

$L_y^{cm1'}$ L_y^{m1m1} L_y^{m1m2}

$L_y^{cm2'}$ $L_y^{m1m2'}$ L_y^{m2m2}

$$L_y = D_y - K_y$$

Step VI - Optimize A via Laplacian Lx.

	width of kernel	asymmetry coefficient	length of kernel groove
1	3.312	2.221	5.220
2	3.561	4.076	6.060
3	3.484	4.675	5.877
4	2.994	5.304	5.395
5	3.338	3.796	5.476
6	3.338	3.796	5.476
7	3.337	2.699	4.825

	width of kernel	asymmetry coefficient	length of kernel groove
1	3.312	2.221	5.220
2	3.561	4.076	6.060
3	3.484	4.675	5.877
4	2.994	5.304	5.395
5	1.993	2.677	3.523
6	2.079	2.723	3.652
7	3.337	2.699	4.825

***A** is dataset Y then optimize the Am1*

$$Am1 = -(L_x^{m1m1})^{-1}((L_x^{cm1})^T Ac - L_x^{m1m2} Am2)$$

Step VII - Optimize B via Laplacian Ly.

	area	perimeter	compactness	length of kernel
1	15.26	14.84	0.8710	5.763
2	17.63	15.98	0.8673	6.191
3	16.84	15.67	0.8623	5.998
4	13.07	13.92	0.8480	5.472
5	14.88	14.57	0.8811	5.554
6	13.32	13.94	0.8613	5.541
7	**15.17**	**14.82**	**0.8652**	**5.754**

	area	perimeter	compactness	length of kernel
1	15.26	14.84	0.8710	5.763
2	17.63	15.98	0.8673	6.191
3	16.84	15.67	0.8623	5.998
4	13.07	13.92	0.8480	5.472
5	14.88	14.57	0.8811	5.554
6	13.32	13.94	0.8613	5.541
7	**6.01**	**5.54**	**0.2962**	**2.140**

***B** is dataset X then optimize the Bm2*

$$Bm2 = -(L_y^{m2m2})^{-1}((L_y^{cm2})^T Bc - (L_y^{m1m2})^T Bm1)$$

Step VIII - Calculate the new kernel matrix Kx' & Ky'.

Same to step IV

Using $Ky' = YY' = AA'$; $Kx' = XX' = BB'$

Then assign $Ky = Ky'$; $Kx = Kx'$

Step IX - Repeat step V - VIII until convergence.

$$RMSx = \sqrt{avg((Kx' - Kx)^2)}$$

$$RMSy = \sqrt{avg((Ky' - Ky)^2)}$$

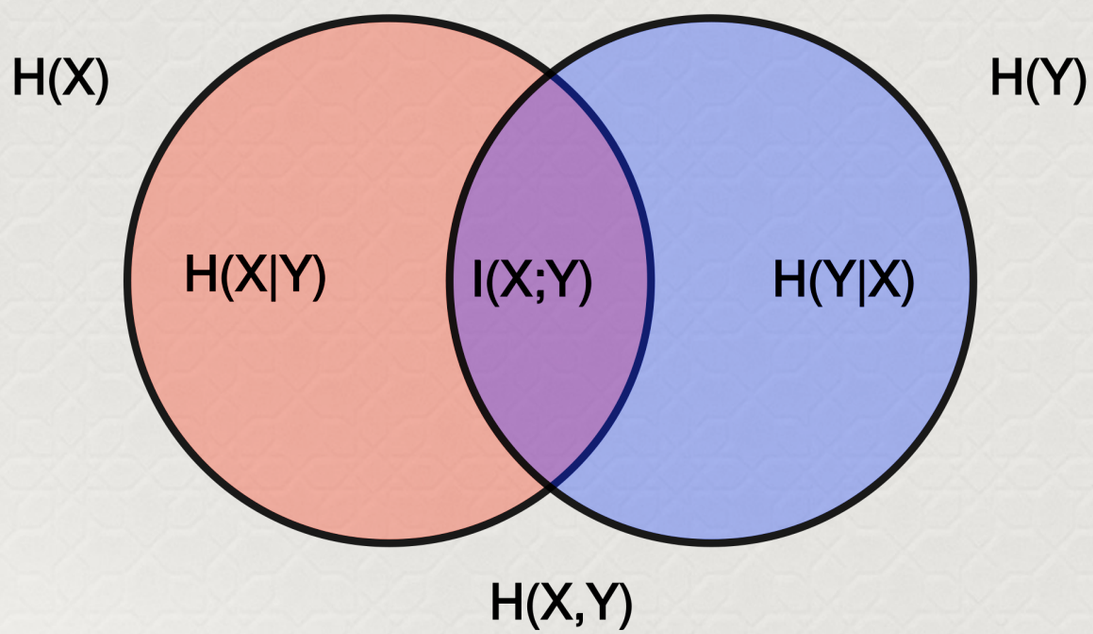
	RMSx	RMSy
1	159.1148	15.02907
2	41.75921	9.006552
3	26.79682	2.776422
4	7.556251	1.741403
5	4.804507	0.50513
6	1.370548	0.31985848
...	...	...
15	0.0009430079	0.00009946435

< 0.001

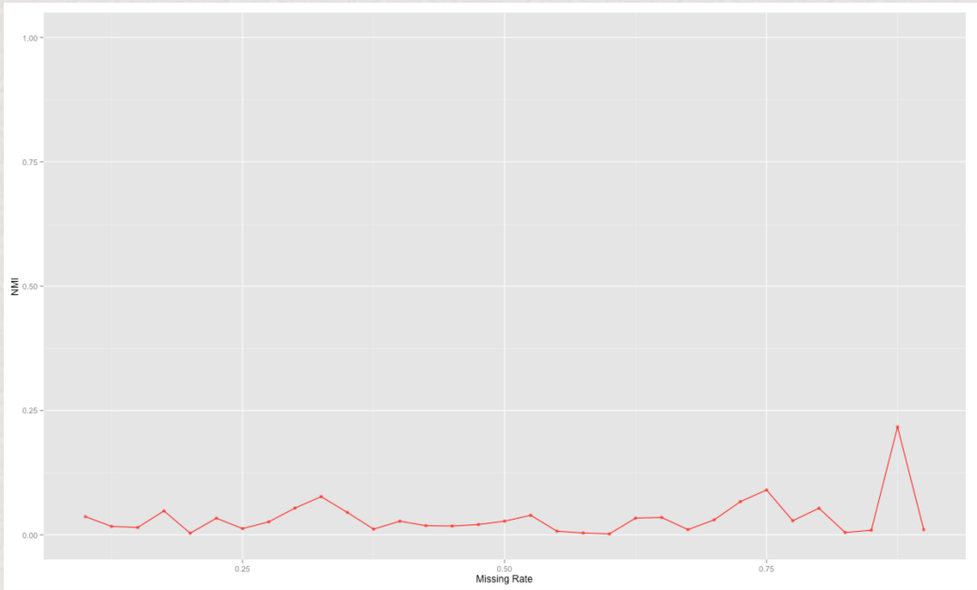
Step X - Using KCCA by CoKL kernel matrix, then clustering.

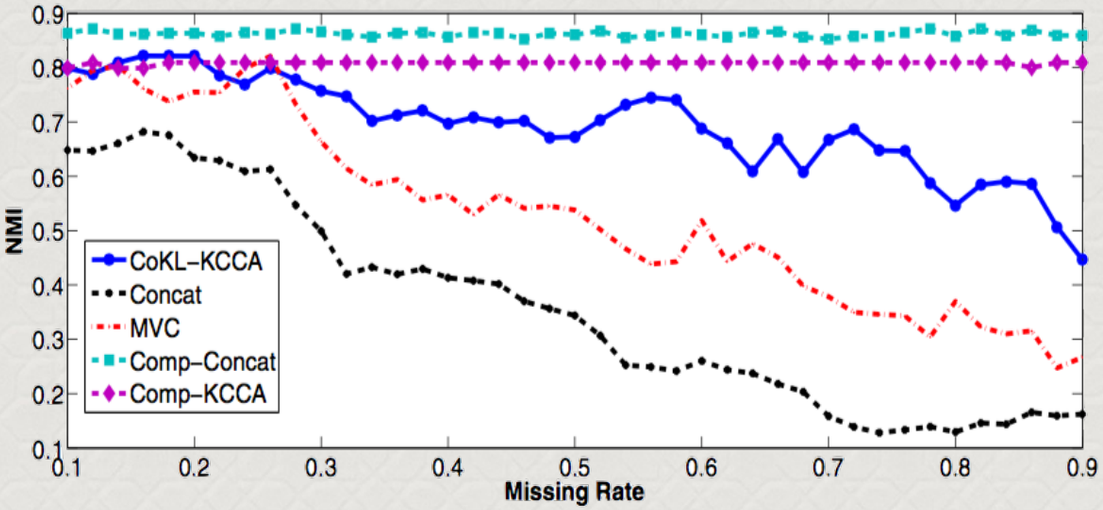
```
              [,1]          [,2]          [,3]          [,4]          [,5]          [,6]
[1,] -6.722129e-20  8.129916e-16  2.184374e-14  2.941248e-15 -1.066549e-14 -3.840457e-14
[2,]  7.157864e-17 -1.669764e-16 -4.300243e-17 -1.174821e-16  3.232658e-16  3.883736e-16
[3,]  2.037493e-15  9.368752e-15  4.397392e-15  8.607854e-13 -7.581357e-12 -1.611085e-01
[4,]  1.122327e-16  3.267914e-11  7.538620e-12  4.279702e-16 -3.718314e-09 -1.380525e-06
[5,]  4.536959e-18 -1.752188e-15  7.379340e-15  9.754506e-16  1.301297e-15 -1.680233e-11
[6,]  6.524370e-17  1.423458e-14 -1.756786e-14  7.442259e-09 -2.538897e-12 -3.559732e-12
[7,]  2.407868e-16  3.340377e-15  4.992741e-15  1.308806e-14 -1.020300e-14  3.582647e-15
```

Step XI - Normalized Mutual Information (NMI).

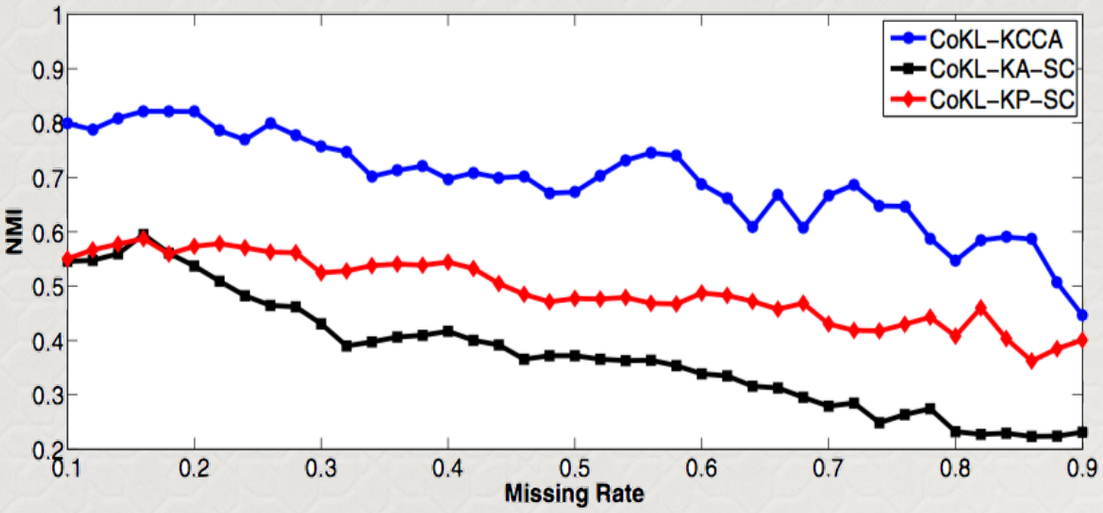


Experiments Result





(a) NMI for seeds dataset on different missing rates.



(c) NMI for seeds dataset on different missing rates.

```
> for (i in 1:10) print(NMI(kmeans(seeds[,1:7], 3)$cluster, seeds$class_of_wheat))
[1] 0.694925
[1] 0.694925
[1] 0.7100638
[1] 0.694925
[1] 0.7100638
[1] 0.694925
[1] 0.694925
[1] 0.7100638
[1] 0.694925
[1] 0.7100638
```

Kernel Matrix

> ***Linear kernel***

Kernel in KCCA

> ***Radial Basis***
> ***Polynomial***
> ***Linear***
> ***...***

Thanks for listening