IGTD Algorithm

$X \in \mathbb{R}^{M \times N}$ הטבלה שנרצה להמיר לתמונות

goal: to transform each sample x_i into $N_r \times N_c$ image when $N_r \times N_c = N$

Define: (i) the pairwise distances between features
- Euclidean
-
-

- These pairwise distances are then ranked ascendingly so small distances getting small rank
- An N by N rank matrix denoted by R is formed (symmetric)

(ii) the pairwise distances between pixels
- Euclidean
-

- These pairwise distances are then ranked ascendingly so small distances getting small rank
- An N by N rank matrix denoted by Q is formed (symmetric)

To transform tabular data to images each feature needs to be assigned to a pixel position in the image

A simple way is to assign the i-th feature (the ith row & column) in R to the ith pixel (the ith row & column) in Q.

$\Rightarrow$ An Error function is defined to measure the difference

$$err(R,Q) = \sum_{i=2}^{N} \sum_{j=1}^{i-1} diff(r_{ij}, q_{ij})$$

כונה שמודדת את ה"שוני" בין ה Rank של הפיצר ה i וה j לעומת ה Rank של הפיקסל שמייצג את הפיצר ה i והפיקסל שמייצג את הפיקסל ה j. בדרך כלל פונקציות:

- absolute error
- square error
- ...

key point: features similar to each other are close in the image $\Leftrightarrow$ err(Q,R) is small

IGTD Algo (S_{max}, S_{con}, t_{con}, t_{swap})

$\in \mathbb{N}^+$

$S_{max} >> S_{con}$

S_{max} → Max # iters

S_{con} → # of iters to check convergence

t_{con} → still epsilon

t_{swap} → error rate to determine if a feature swap should exe.

} input params

step 1) init: • iter index $s = 0$

• $e_0 = err(R, Q)$

• h - vector of negative infinities, $len(h) = N$
record the last iter which each feature was consider for feature swap

• $k_0 = [1, \_, N]$ order of features before opt.

step 2) identifies the feature that has not been considered for feature swap for the longest time & searches for a feature swap for it meanning:

$$n^* = \arg\min_{i \in [N]} h[i]$$

$$l^* = \arg\max_{l \in \{1, n^*-1, n^*+1, \_, N\}} err(R, Q) - err(R_{n^* \sim l}, Q)$$

• $s = s + 1$

step 3) if $[err(R, Q) - err(R_{n^* \sim l^*}, Q)] > t_{swap}$ {

(i) $k_s = k_{s-1}$ & swap the n^*th and l^*th elements in k_s

(ii) $e_s = err(R_{n^* \sim l^*}, Q)$

(iii) $h_{n^*} = h_{l^*} = s$

(iv) $R = R_{n^* \sim l^*}$

}

else { (i) $h_{n^*} = s$

(ii) $e_s = e_{s-1}$

(iii) $k_s = k_{s-1}$

}

step 4) check if the algo should terminate or continue

if $s = s_{max}$ OR $\frac{e_s - s_{con} - e_u}{e_s - s_{con}} < t_{con}$ $\forall u \in \{s - s_{con} + 1, \ldots, s\}$

so the algo identifies the iteration with minimum error $v^* = \arg\min_{v \in \langle s \rangle} e_v$

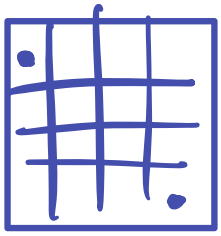
it then terminates and output k_{v^*} and e_{v^*} which are the optimized error resulted from reordering the features according to k_{v^*}

Q

n_row = 2
n_col = 2 } as the number of features

Corrdinate:

(0,0) (1,0)
(0,1) (1,1)
(0,2) (1,2)
(0,3) (1,3)



corr dist:

dist for cell ij

$$\sqrt{\left[i\cdot\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}-\begin{pmatrix}0\\0\\0\\0\\1\\\vdots\\3\end{pmatrix}\right]^2+\left[j\cdot\begin{pmatrix}1\\2\\\vdots\\1\end{pmatrix}-\begin{pmatrix}0\\1\\2\\3\\\vdots\end{pmatrix}\right]^2}$$

corr dist = Q :

	0	1	2	3
0	0	1	1	1.4
1	1	0	1.4	1
2	1	1.4	0	1
3	1.4	1	1	0

המספרים מדגלים
בקצוות לכן נבדה
שגם יהיו הפיצרים
שהכי פחות קשורים
לגיל לדוגמה

R

	age	gender	color	food
age	0	2	7	1
gender	2	0	5	-3
color	7	5	0	2
food	1	-3	2	0

corr

⇓

	age	gender	color	food
age	0	4	6	2
gender	4	0	5	1
color	6	5	0	3
food	2	1	3	0

feature ranking

age	gender	color	food
0	0	0	0
⋮	8		

4
0
6
2

i = 0
j = 1

limitations:

- what if * features isn't even? use padding?

-

נקודות חשובות מהמאמר של IGTD

Image Generator for Tabular Data – IGTD - שיטה להמרה של דאטא טבלאי לתמונות

יש התמקדות ב-3 שיטות שקיימות כיום להמרה של data טבלאי לתמונות

1) **Sharma et al. developed DeepInsight**
שיטה זו מבוססת על הטלה של וקטורי הפיצ'רים על מרחב דו ממדי באמצעות t-SNE (שיטה לא לינארית להורדת ממדים של data לממד 2 או 3), אשר ממזער את ה KL divergence בין התפלגויות הפיצ'רים במרחב המוטל מול ההתפלגות המקורית במרחב המלא. אח"כ על ההטלה הדו ממדית, האלגוריתם מזהה מלבן אשר כולל את כל נקודות הפיצ'רים המוטלות ושטחו המינימלי האפשרי ומלבן זה יוצר את ה feature representation.

2) **Bazgir et al. developed REFINED**
משתמש ב Bayesian multidimensional scaling כ global distortion minimizer כדי להטיל את ה data על מרחב דו ממדי שמשמר את התפלגויות הפיצ'רים המקוריות. אח"כ מבצעים השמה של פיצ'רים לפיקסלים על פי ההטלה ו-hill climbing algorithm ממומש כדי למקסם לוקאלית את הסידור של הפיצ'רים במיקום בתמונה (למקסם את ההשמה).

3) **Ma and Zhang developed OmnicsMapNet**
נוצר כדי להמיר gene expression data של חולי סרטן לתמונות 2-D על מנת לבצע פרדיקצה של tumor grade באמצעות CNNs. הרעיון ליצור תמונה כך שגנים בעלי molecular functions דומות יהיו סמוכים בתמונה.

מה היתרונות של IGTD על פני 3 השיטות שהוצגו מקודם?

- בשונה מ OmnicsMapNet שדורש domain knowledge על הפיצ'רים (לא התעמקתי למה), ב IGTD אין צורך ב domain knowledge כלל.
- בשונה מ DeepInsight אשר בגלל השימוש שלהם בהטלת t-SNE הפלט יהיה תמונה לא שלמה ומכילה פיקסלים שלא מייצגים אף פיצ'ר, כלומר הם blanks, ב IGTD מחזירים תמונה שלמה ומלאה כאשר כל פיקסל מייצג פיצ'ר באופן unique. כתוצאה מכך התמונות של DeepInsight לרוב גדולות יותר מאשר התמונות ש IGTD מפיק מה שגורר זמני אימון גבוהים יותר ויותר צריכה של זיכרון.
- ב IGTD ה feature representation משמר יותר טוב את ה feature neighborhood structure מאשר ב-REFINED – הם טוענים את זה בלי הסבר כ"כ אומרים שרואים את זה בדוגמה שהם מביאים.