ניר שטמלר
רותם בן חמו

Mining Low-Support Discriminative Patterns from Dense and High-Dimensional Data

- מושגים בסיסיים

- עקרונות

מושגים בסיסיים

D- n instances of items from I.

$I=\{i_1,i_2....i_m\}$

S1,S2= two class labels of n labeled instances.

D={(xi,yj)| i=1…n, xi ϵ I , yjϵ{S1,S2}}. D is set of instances.

D1={(xi,S1)} i=1…n } ∩ D, D2 ={(xi,S2)| i=1…n} ∩ D.

- |D1|+|D2|=|D| → No instance with two labels.

D היא קבוצה של n פריטים מתוך i, לכל אחד קיטלוג מבין S1 או S2.

For α={α1, α2… αl} <u>C</u> I. D1α is the *set of instances* in D1 that contains α,

D2α is the *set of instances* in D2 that contains α.

$I = \{i1, i2 \ldots i_m\}$

D=n labeled instances from I

α={i1,i7}

D1α

D2α

D1	D2	
(i1,i7),S1	(i1,i2),S2	*Instance 1*
(i1,i7,i2),S1	**(i1,i7,i11),S2**	*Instance 2*
(i11,S1)	(i21),S2	*Instance 3*
	.	.
.	.	.
.	.	.
.	.	.

For example if |D1|=40, |D2|=10
Relsup1(α)=2\40
Relsup2(α)=1\10
Diffsup(α)= 1\20

The **relative supports** of α in classes S1,S2 are**:**

Relsup1(α)= $\frac{|D1\alpha|}{|D1|}$ **and Relsup2(α)=** $\frac{|D2\alpha|}{|D2|}$

The **absolute difference** of the relative supports of α in D1 and D2 is:

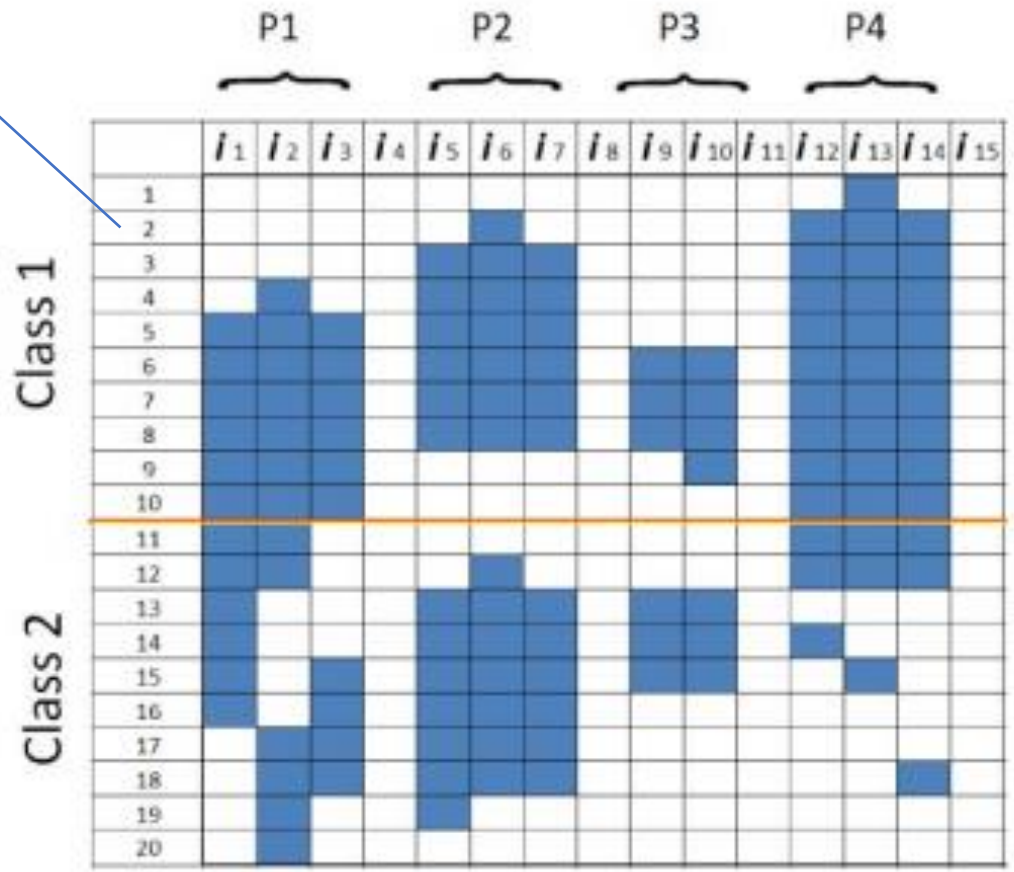
Diffsup(α)=|Relsup1(α)-Relsup2(α)|

An itemset **α** is r-discriminative if Diffsup(**α**)≥r

דוגמא

D- m מוצרים לקנייה בסופרמרקט

בכל מחלקה יש 10 הופעות- 10 עגלות קניות שונות של צרכנים. כל P היא קבוצה αC̲l.



Class 1- עגלות של אנשים עשירים

Class 2- עגלות של אנשים עניים

נתבונן ב α =P1:

α ={i1,i2,i3}

*D1*α=6. (sets 5,6,7,8,9,10)

*D2*α=0.

*Relsup1(*α)= 6\10,

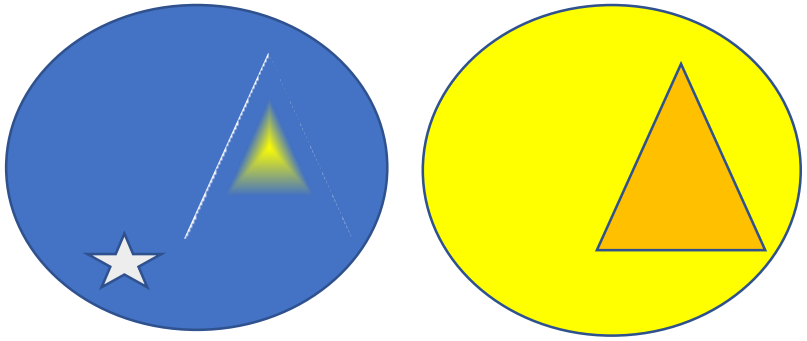
Relsup2(α)=0.

Diffsup(α)=0.6-0=0.6

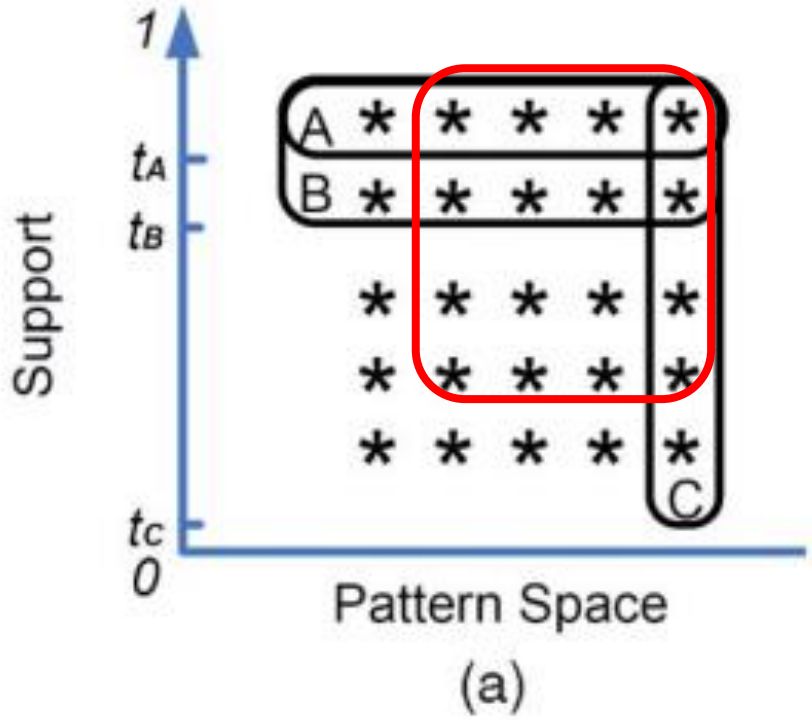
Diffsup(P2)=Diffsup(P3)=0

בהקשר שלנו, **α** היא קבוצת קוגים משותפת לחיידקים שונים שהוחלט לבדוק, ו-S1, S2 הן תגיות של רצפים (למשל- עמיד ולא עמיד לאנטיביוטיקה). אם **α- קבוצת קוגים כלשהי,** תמצא כמבדילה אז התבנית **α** תחשב כמבדילה בין גנים של חיידקים שעמידים לאנטיביוטיקה לבין חיידקים שאינם עמידים.

תבניות המוגדרות כ**low support** אלו תבניות שמבדילות בין שני סוגי החיידקים, אך הרצפים במאגר שמכילים את התבניות האלה הם מעטים. למשל 100 רצפים שיכילו גן זהה כשכל 100 החיידקים עמידים לאנטיביוטיקה, ואף חיידק שאינו עמיד לא מחזיק בגן. אין ספק שניתן להתייחס לגן הנ"ל כמבדיל, אך מתוך מאגר גדול של עשרות אלפי חיידקים יהיה קשה לזהות את הגן. גן כזה וכמותו נקראים low support discriminative patterns.



במאמר מוצגים עקרונות של אלגוריתמים שונים למציאת "תבניות מבדילות" בין תגיות במאגרי מידע.

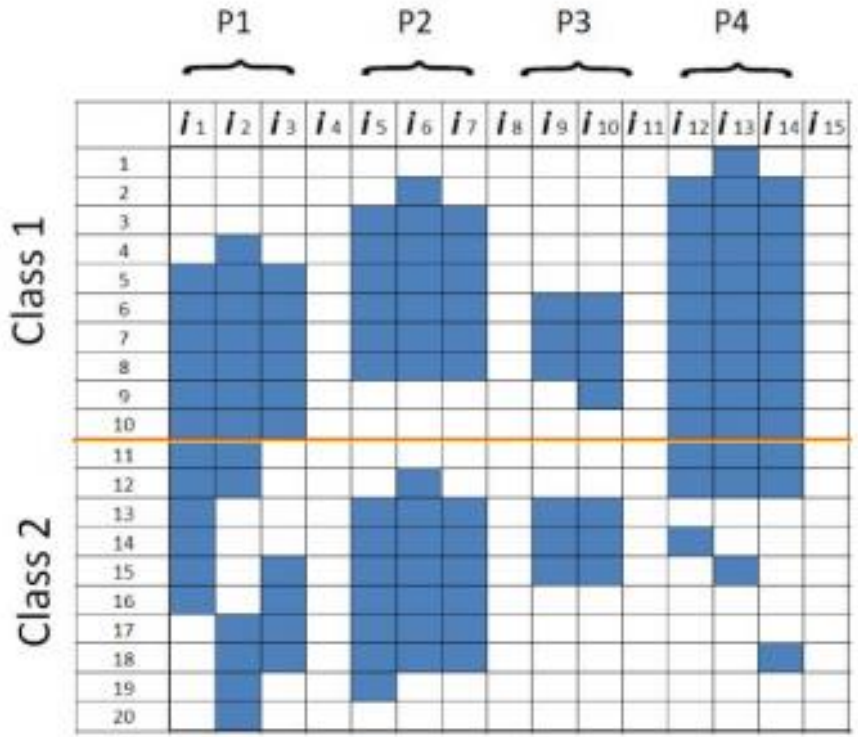


*בציר הY הt מייצג את התמיכה הנמוכה ביותר (lowest support) של תבניות שמתגלות כמבדילות על ידי האלגוריתמים השונים **בזמן חישוב נתון**. A, B וC אלו גישות שונות של אלגוריתמים שונים.*
ציר הx מייצג את כיסויי התבניות המבדילות על ידי האלגוריתמים.
הגרף משקף את הבעיות בקבוצות האלגוריתמים A,B וC.
A וB מצליחים לכסות מרחב רחב של תבניות מבדילות, אך לא מצליחים לגלות תבניות שהן low support. לעומת זאת אלגוריתמים מקבוצה C מזהים תבניות low support אך מכסות מרחב צר של תבניות מבדילות.
המוטיבציה למאמר ולרעיון המוצג היא ביצוע trade-off בין שני העקרונות ונסות להגיע לתוצאות כבקבוצה האדומה.

המשך הגדרות- SupMax

$SupMax1(\alpha)=Relsup1(\alpha)-max_{b\epsilon\alpha}(RelSup2(\{b\}))$

בדוגמא שהוצגה קודם:



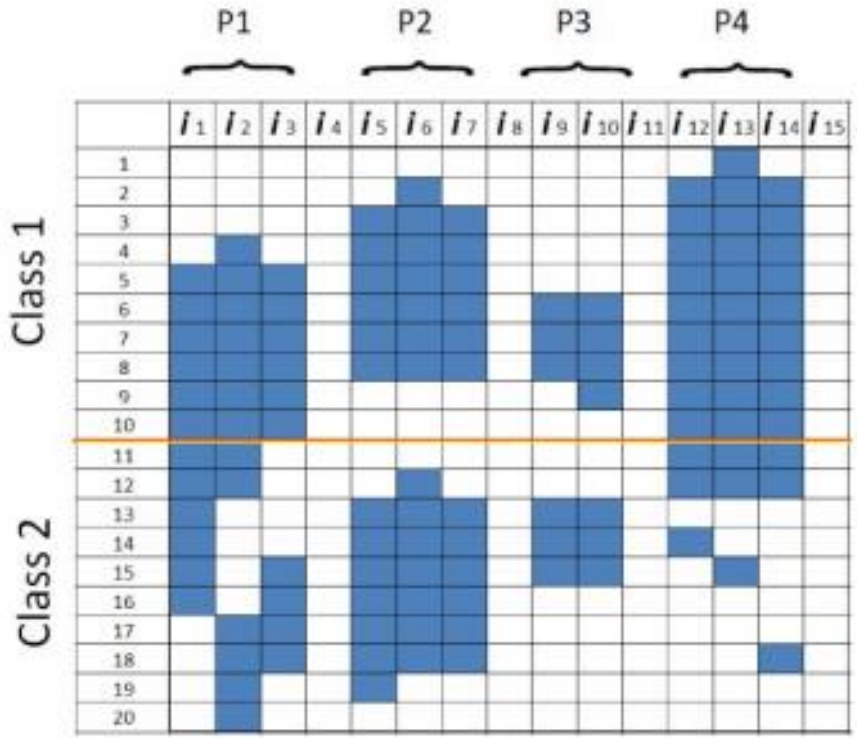
$SupMax1(P1)=6-max\{6,6,4\}=0$

$RelSup(P1)=6$

SupMax1 הוא קירוב גס מאוד (מדי) ל RelSup.

המשך הגדרות- SupMax

$$SupMaxK(\alpha)=RelSup1(\alpha)-\max_{b\subseteq\alpha}(RelSup2(b)), \qquad |b|=K$$



SubMax2(P1)=6-max(2,2,2)=4

RelSup(P1)=6

SubMax2 הוא קירוב טוב יותר לRelsup.

תכונות של SupMaxK

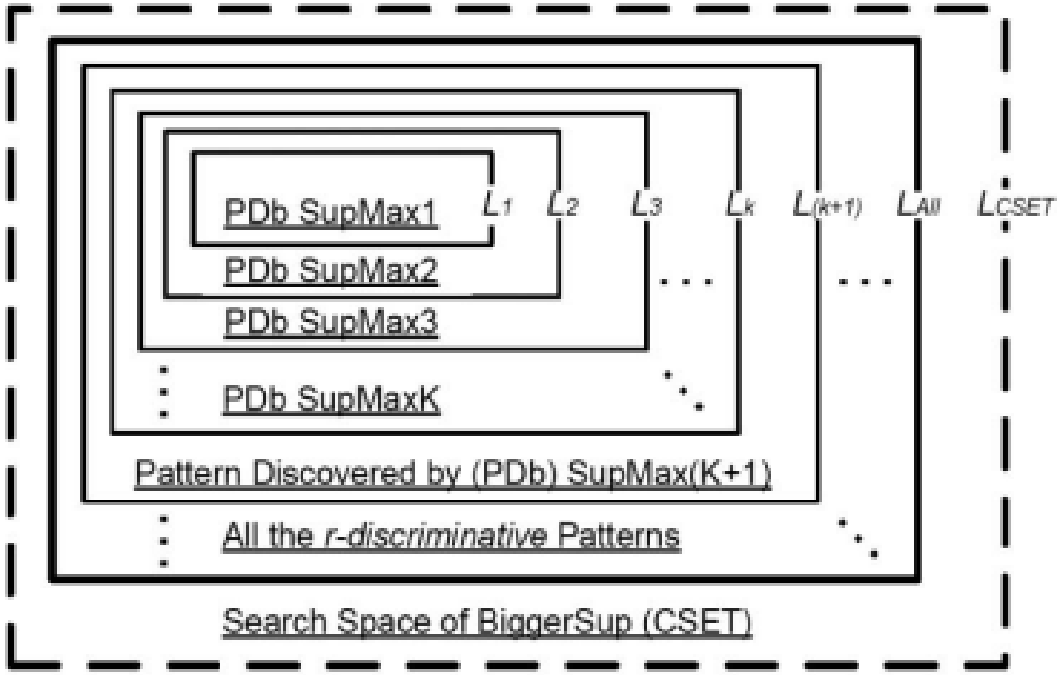
$SupMaxK(\alpha)=RelSup1(\alpha)-max_{b\underline{C}\alpha}(RelSup2(b)), \quad |b|=K$

נסמן $max_{b\underline{C}\alpha}(RelSup2(b))=MaxSup(\alpha,K)$

- מתקיים $MaxSup(\alpha,K) \leq MaxSup(\alpha,K-1)$. לכן $SupMax(K-1)\leq SupMax(K)$

- אם $K=|\alpha|$ אז $SupMaxK=Diffsup$. לכן DiffSup הוא גבול עליון לSupMax.

- כלומר ככל שנגדיל את K, נגלה יותר r-discriminative patterns.



L_{All} מתקבלת ע"י DiffSup

חישוב של MaxSup2

- ערכים שיחושבו לפני חישוב MaxSup(α, 2): (נסמן $|\alpha| = l$)
- MaxSup($\{\alpha_1, \alpha_2 \ldots, \alpha_{l-1}\}$, 2), MaxSup($\{\alpha_1, \alpha_2 \ldots, \alpha_{l-2}, \alpha_l\}$, 2), MaxSup($\{\alpha_{l-1}, \alpha_l\}$, 2) .
- *נותר לבחור את המקסימלי מבין שלושת הערכים. לכן, חישוב* MaxSup(α, 2) עבור סט α *הוא בזמן קבוע* $O(1)$.
- *מקרי בסיס שיש לחשב:*
- *MaxSup לα בגודל 0 (=0), בגודל 1 (0), ובגודל 2 (חישוב חד פעמי של* $O(l^2)$.

עקרונות מימוש האלגוריתם- מציאת תבניות מבדילות בין ActinoBacterias לFirmcutes

עקרונות מימוש האלגוריתם- מציאת תבניות מבדילות בין Firmcutes לActinoBacterias

- 1- עיבוד מקדים לData הנתון: יצירת שתי טבלאות עבור שתי המחלקות הנבדקות בלבד. טבלה לכל מחלקה.

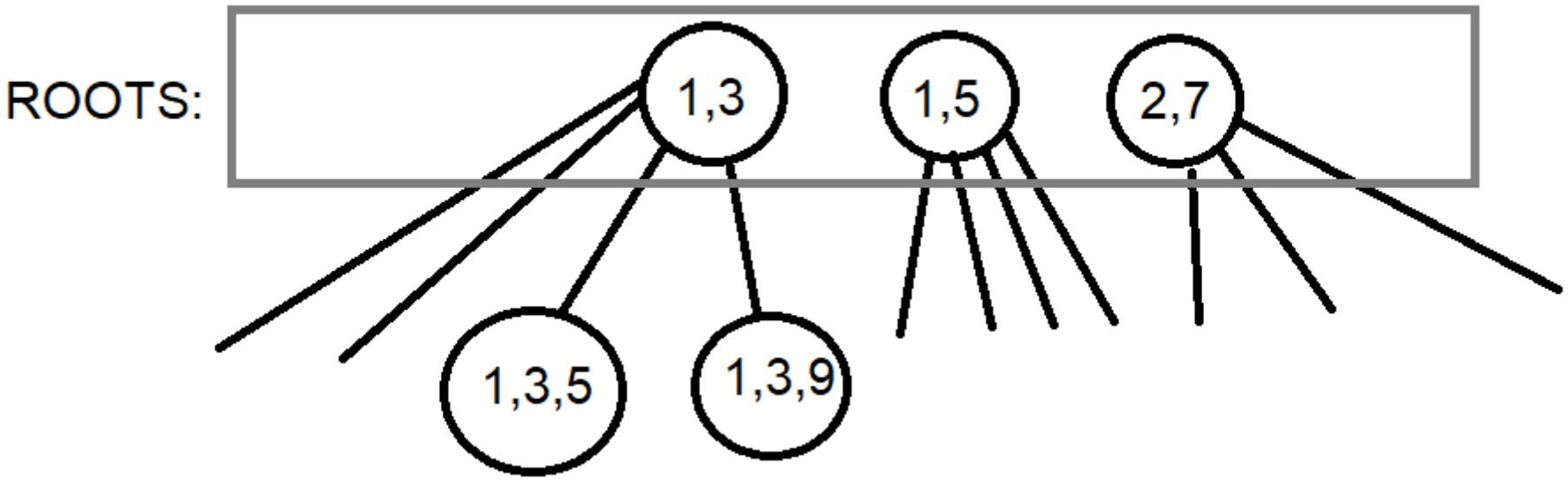
	Cog1	Cog2	…	Cog5665
Seq1	1	0		1
Seq2	0	0		0
Seq 3	1	1		0
…				

- 2- בניית מקרה הבסיס לחישוב ערכי maxSup2:
  טבלה דו-מימדית: cog מול cog.

	Cog1	Cog2	…	Cog5665
Cog1		0.5		0.25
Cog2				0.124
…				
Cog5665				

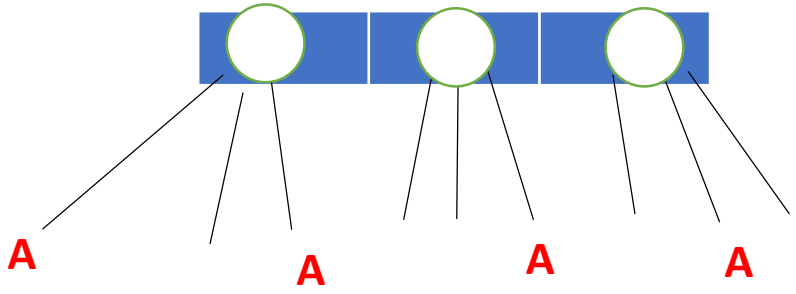
- *3- בניית עץ חישוב. כל זוג Cogs מהווה שורש של עץ. כבר כאן נבדק תנאי הגיזום ע"י חישוב relSup1-maxSup2 לכל אלפא.*

עץ חישוב: כל קודקוד בעץ הוא תת קבוצה של Cogs. כאשר כל בן מרחיב את אביו.



כל קודקוד מחזיק ערך maxSup עבור התגית שלו.

- הרחבת העץ לרוחב. בכל איטרציה מחושבת רמה חדשה של העץ, על סמך מקרה הבסיס והרמה הקודמת בלבד.
- כל תא שמחושב ועובר את תנאי הגיזום, מסומן כACTIVE. כל תא שמחושב ולא עומד בתנאי, גם הוא נכנס לעץ אך לא מסומן כACTIVE



- בכל איטרציה, נרחיב רק את הקודקודים המסומנים כACTIVE.
- **תוצאות:** האלגוריתם מחזיר את כל התגיות של העלים הפעילים, ואת התגיות של הקודקודים הפנימיים שכל ילדהם אינם פעילים.

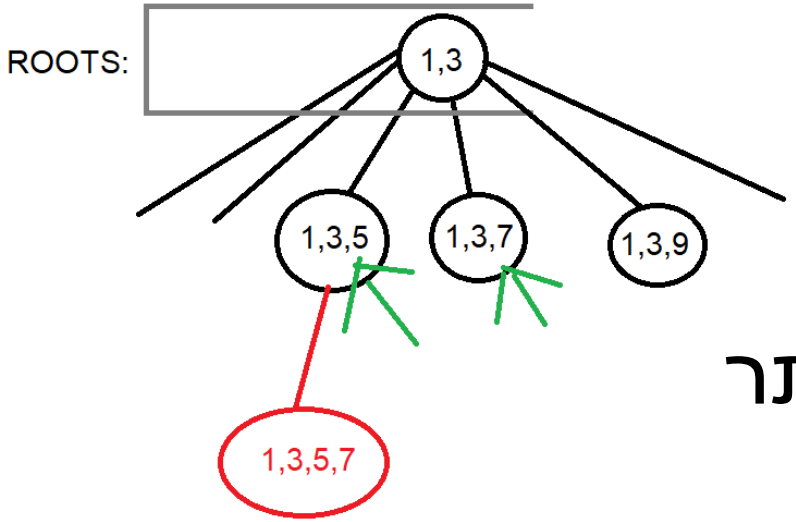
תנאי גיזום העץ *relsup1(a)-supmax2(a)<r*

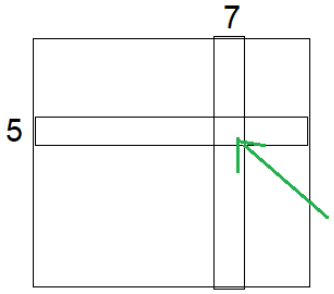
- תהי תגית $a=(a_1 \ldots a_n)$ המקיימת $score(a)=relsup1(a)-supmax2(a)<r$, מתקיים:
- עבור תגיד המרחיבה את a: $a'=(a_1 \ldots a_n, a_{n+1})$:
- $Relsup1(a) \geq relsup1(a')$, כיוון של a' לכל היותר מספר הופעות בclass1 כמספר ההופעות של a.
- $Supmax2(a) \leq supmax2(a')$, כיוון ש$supmax2(a')$ הוא ערך מקסימלי מבין שלושה ערכים, שאחד מהם הוא $supmax2(a)$.
- לכן, הציון של a' הוא לכל היותר הציון של a, וניתן לא לחשב את כל ההרחבות של a, ולגזום אותו.

חישוב SupMaxPair(a)=relSup1(a)-maxSup1(a)

- חישוב maxSup(a): מקסימום מבין הערכים:

MaxSup($\{\alpha_1, \alpha_2..., \alpha_{l-1}\}$, 2), MaxSup($\{\alpha_1, \alpha_2..., \alpha_{l-2}, \alpha_l\}$, 2),
MaxSup($\{\alpha_{l-1}, \alpha_l\}$, 2) .





<u>*זמן*</u>: ערך האב- o(1).

ערך הדוד- o(1)- חיפוש (בינארי) בין לכל היותר 5665 ילדי הדוד

ערך מטבלת מקרה הבסיס- o(1).

סה"כ: ליניארי במספר התאים הפעילים.

- חישוב relSup1- באופן נאיבי בטבלה הנתונה, ע"פ Class1.

<u>זמן:</u>

O(|alpha||class1|)*

לכל חישוב תא בעץ. בסה"כ עבור o(N) תאים פעילים (גם לילדיהם הלא פעילים). ובסה"כ חישוב ציון: O(N|alpha|*|class1|).*

כאשר אורך אלפא במקרה הגרוע הוא באורך 5665, אך במקרה ריאלי, לא יגיע למספר תלת ספרתי (במקרה שלנו לא נבדקה אלפא ארוכה מ4).

תוצאות וזמן חישוב

זמן ריצה

- Class1=ActinoBacteria
- Class2=Firmcutes
- *עיבוד מקדים ובניית העץ: כ- 5665^2 תאים.*
- *כמות הקבוצות הנבדקות ללא גיזום: אקספוננציאלי (2^{5665})*
- *כמות הקבוצות הנבדקות עם גיזום, עבור bound=0.6 (מספר תאים פעילים בעץ שחושבו עבור המידע הספציפי שלנו):*

*תאים פעילים- **205**. חישוב על כל תא פעיל- כ- 5665.*

סה"כ

- 5665^2+(**205***5665)
- פולינומי בכמות הקוגים (א"ב).
- ללא עיבוד מקדים ובניית העץ- **o(N*|cogsAmount|)**.
- עבור bound=0.08 למשל, נקבל **N=5606** תאים פעילים.

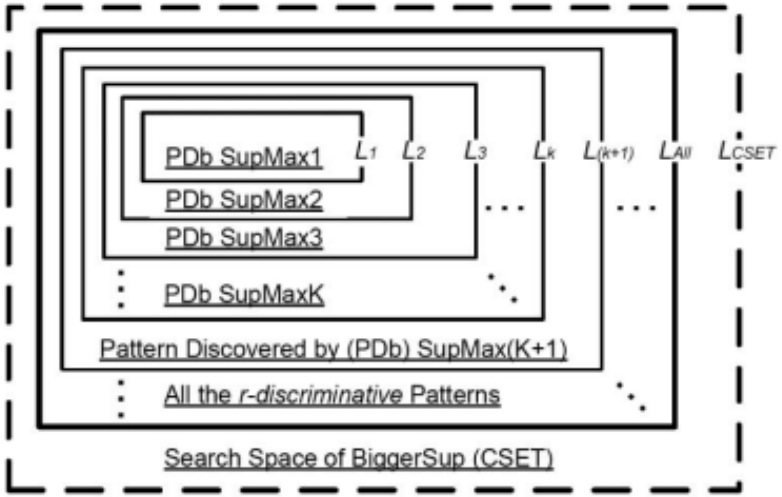
תוצאות עבור תבניות מבדילות עם ערכי bound שונים

- Class1= ActinoBacteria
- Class2=Firmcutes
- עבור bound=0.8 אנו דורשים שלכל הפחות 80% מהתבניות בClass1 יכילו את התבנית המבדילה, לכן תוצאות אלו הן High Support. תוצאות אלו יאפיינו מאוד את Class1 לעומת Class2.
- עבור bound=0.08 למשל, אנו דורשים שלכל הפחות 8% מהתבניות בClass1 יכילו את התבנית המבדילה ולכן תוצאות אלו הן Low Support. תוצאות אלו מבדילות בין שתי המחלקות על פי supMaxPair, אך לא נוכל להגיד שהן מאפיינות רצפים בClass1, אלא שהן מבדילות בלבד.
- אם נריץ את האלגוריתם כך ש Class1=Firmicutes ו Class2=ActinoBacteria נקבל תוצאות שונות עם מידע רלוונטי שיצביע הפעם על תבניות שמאפיינות את Firmicutes.

דוגמאות לתוצאות HighSupport וLowSupport

- מתוך תוצאות Discriminatives-0.8: [789, 817, 1716]. נוכל לומר שתת קבוצה של קוגים אלו מאפיינת ActinoBacterias, ומבדילה אותם מFirmicutes. (High support).
- מתוך Discriminatives-0.08: נבחר תת קבוצה של קוגים שציונה הוא לכל הפחות 0.08, אך נבחר אחת שציונה לא עולה על 0.5 (אפשר להפעיל בדיקה על קבצי התוצאות השונים). למשל, התבנית [16, 396, 515] היא כזו, וניתן לתייג אותה כLowSupport. תבנית זו תופיע בActinoBacterias אך לא תופיע, בדר"כ (ע"פ supMaxPair), במחלקת Firmicutes.

יתרונות וחסרונות מרכזיים לחישוב באמצעות SupMaxPair



- חסרון מרכזי לשיטת החישוב באמצעות SupMaxPair

- אנו מפספסים תבניות מבדילות בחישוב. למשל:

אם קוגים (1,3,5) נפוצים במחלקה 1, ובמחלקה 2 לא קיים השילוב הנ"ל באף רצף, אך הקוגים (1,3) למשל כן נפוצים במחלקה 2, אנו לא נזהה את תת הקבוצה (1,3,5). (אולי יש ביולוגים שלא יתייחסו לזה כחיסרון).

- יתרון מרכזי לשיטת supMaxPair:

- השיטה מאפשרת גיזום וכך חישוב בזמן סביר של תבניות מבדילות בין שתי מחלקות רצפים.

נובע מהתכונה $SupMax(K-1) \leq SupMax(K)$ *(שקופית 9).*