בס״ד

האוניברסיטה הפתוחה
The Open University of Israel

יישום חוקי הקשר למציאת קשרי גומלין בין מיקומי גידולים סרטנים למיקומי גרורותיהם

עבודת מסכמת זו הוגשה כחלק מהדרישות לקבלת תואר
״מוסמך למדעים״ .M.Sc במדעי המחשב
באוניברסיטה הפתוחה
החטיבה למדעי המחשב

טבת התשע״ג

דצמבר 2012

צוריאל כהן

039076070

מנחה : ד״ר מיה הרמן

תוכן עניינים

תקציר

בעבודה מסכמת זו נסקר תת תחום בכריית מידע - חוקי הקשר. חוקי הקשר הינם כלי משמעותי בתחום כריית המידע. בעזרת חוקי ההקשר ניתן לבצע כריית מידע מעל בסיסי נתונים. חוקי ההקשר מאפשרים לנו למצוא ולאפיין קשרים בין שדות בבסיס הנתונים. על סמך הקשרים הנ״ל ניתן יהיה להסיק מידע חדש שלא היה ידוע קודם לכן מתוך הנתונים שלפנינו.

לעבודה זו ארבעה חלקים:

בחלק הראשון: הובאו הגדרות ומונחים בסיסיים בכריית חוקי הקשר והוסברו מושגים בסיסים בכריית מידע.

בחלק השני מוצגת סקירה השוואתית של ששה אלגוריתמים בסיסיים לכריית חוקי הקשר:

- האלגוריתם הנאיבי
- Apriori
- FP – Growth
- Eclat
- DIC
- Carma

השוני המהותי בין האלגוריתמים הנ״ל בא לידי ביטוי בעיקר הן בדרך שבה הם מנסים לכרות את חוקי ההקשר והן בסוגי מבני הנתונים בהם הם עושים שימוש. כמובן שהבדלים אלו יבואו לידי ביטוי בסיבוכיות זמן ומקום של כל אחד מהאלגוריתמים הנ״ל.

בחלק השלישי של העבודה מוצגת סקירה של שימושים בחוקי הקשר בתחום הביולוגי והרפואי – בדגש על מחלת הסרטן.

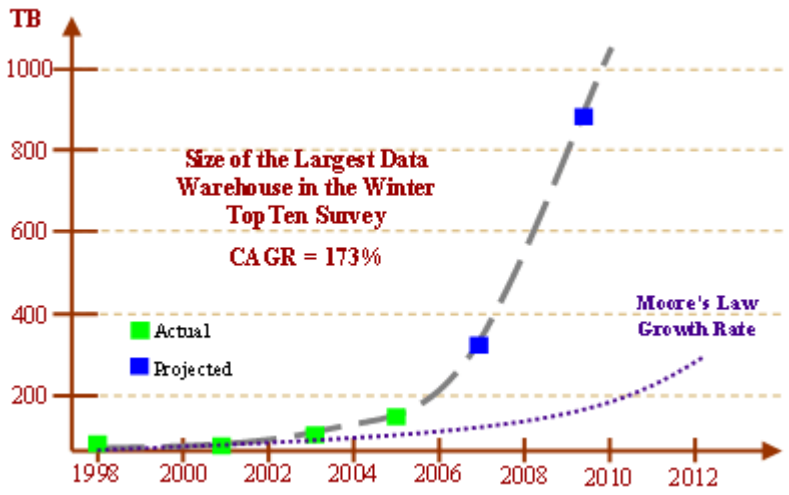
מהות הבעיה שבה העבודה עוסקת היא כריית חוקי הקשר רפואיים מתוך בסיס נתונים רפואיים המכיל מידע לגבי גרורות וגידולים סרטניים. ברצוננו למצוא בעזרת חוקי ההקשר את הקשרים בין פרמטרים בבסיס הנתונים הנ״ל, לדוגמא: קשר בין מיקום של גרורה למיקום גידול סרטני.

בחלק הרביעי מובא מימוש לפיתרון הבעיה בשימוש תוכנת SPSS – Clementine 12.0 של חלק מן האלגוריתמים הנ״ל. בהינתן לנו בסיס נתונים המכיל מידע רפואי לגבי מיקומים של גידולים סרטניים ושל גרורותיהם. נעשה שימוש באלגוריתמים הללו בכדי לכרות מידע מתוך בסיס הנתונים וננסה למצוא קשרים בין מיקומי הגידולים הסרטניים לבין מיקומי הגרורות. לאחר מכן יבוצע ניתוח השוואתי של תוצאות הכרייה שהתקבלו בשימוש בכל אחד מהאלגוריתמים שנבדקו. שאלת המחקר היא איפה, האם ניתן למצוא קשרים ברמת סבירות מספיקה בין מיקומו של גידול בגוף למיקומי גרורותיו. בסיומו של החלק הרביעי יוסקו מסקנות ויוצגו החוקים שהתקבלו.

בתור מחקר המשך לבעיה ניתן למצוא בסיס הנתונים גדול יותר, הן מבחינת כמות החולים שהוא מייצג והן מבחינת כמות המאפיינים בבסיס הנתונים (כלומר שימוש ביותר סוגי גידולים וגרורות). כמו כן ניתן להוסיף עוד פרמטרים רפואיים הקשורים לגידולים הנ״ל ולחפש קשרים ביניהם. בדיקות על בסיס נתונים מהסוגים הללו יוכלו לתקף את תוצאות המחקר שבוצע בעבודה זו.

1.מבוא

במהלך העשורים האחרונים נצפתה עליה דרמטית בכמות המידע המאוחסן בצורה דיגיטאלית כמודגם באיור 1.1 .

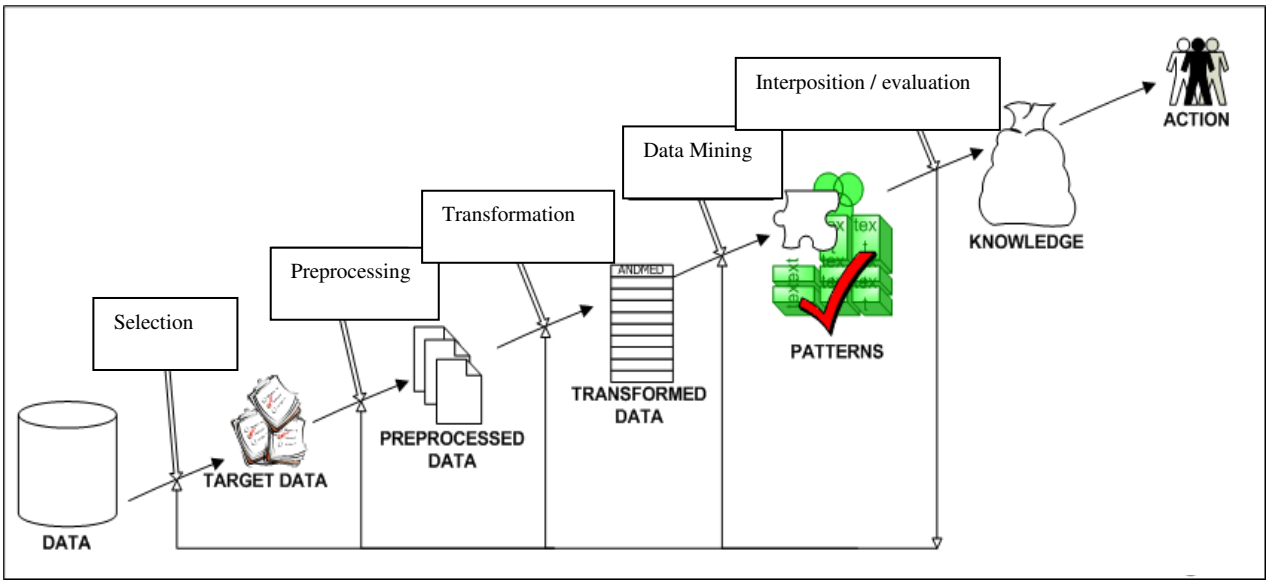


איור 1.1 - קצב גידול נפח המידע כפונקציה של הזמן[33]

ציר ה X בגרף מייצג את ציר הזמן ואילו ציר ה Y את נפחי בסיסי הנתונים באותה תקופה. במשך הזמן, אחסון מידע הפך לקל יותר, זול יותר, ובעל יכולות אחסון גבוהות יותר. עם הזמן ניתן היה להבחין שכמות כה עצומה וזמינה של מידע יכולה לשמש לצרכים שונים מלבד אחסון מידע.

הועלתה האפשרות שנוכל *לנתח* את המידע המאוחסן ולהסיק ממנו מידע חדש. ומכאן נולד המושג כריית מידע: כריית מידע הינו תהליך שבו אנו מנסים לזהות דפוסים / תבניות בנתונים הקיימים לפנינו. כמו כן כריית מידע כוללת את היכולת לנבות דפוסים ותבניות כאלו על סמך דפוסים ותבניות שנמצאו בעבר.המושג עצמו נוצר כאלגוריה לכרייה הגיאולוגית שבה כורים באדמה ע"מ למצוא אוצרות טבע.

תהליך גילוי המידע מבסיסי נתונים הינו תהליך ארוך ומורכב הכולל כמה שלבים כאשר כריית המידע הינה לב ליבו של התהליך[34]. ניתן לראות את תהליך הכרייה באיור 1.2.



איור 1.2 - תהליך גילוי המידע מבסיסי הנתונים[40]

שלבי תהליך כריית המידע הם[13][8]

1. ***הגדרת הבעיה ומטרת המחקר*** - בשלב זה נגדיר את הבעיה שאותה אנו רוצים לפתור / לחקור בעזרת כריית מידע.
2. ***איסוף המידע*** – לאחר שהוגדר מה ברצוננו לחקור, נצטרך להתחיל לאסוף את המידע הנ"ל בבסיסי נתונים. לדוגמא אם ברצוננו לחקור הרגלי קניה של לקוחות במקום מסוים. יהיה עלינו לאסוף מידע על כל הקניות שנערכו באותו מקום ועל כל הלקוחות שקנו באותו מקום.
3. ***בחירת המידע וניפויו*** – בחירת המידע הגולמי מתוך בסיס הנתונים וניפוי המידע מ 'רעשים'.
4. ***עיבוד מקדים*** – עיבוד של המידע לפני הניתוח וכריית המידע, הורדה של מידע כפול ,השלמה של מידע חסר/ לא עקבי וכדו' .
5. ***המרת המידע*** – המרה של המידע לפורמט המתאים לתהליך כריית המידע.

ניתן לאחד את שלבים 3-5 לשלב אחד הקרוי *טיוב המידע / עיבוד מקדים (Pre Procesing)*

6. ***כריית המידע*** – תהליך הכרייה עצמו, ביצוע פעולות על המידע על מנת שניתן יהיה להסיק את המידע "החבוי" במידע . קיימות מספר שיטות לכריית מידע, כל שיטה מתמודדת עם סוגים שונים של בעיות בתחום המידע. לדוגמא:
   - 6.1. בעיות סיווג – בהתאם לדוגמאות שנאספו בעבר ונאגרו בבסיס הנתונים נוכל ע"י תהליך של כריית מידע בשיטת הסיווג (Classification) לסווג ולהגדיר מקרים חדשים. לדוג' בהתאם לנתוני לקוחות קודמים יידע הבנק האם הלקוח הנוכחי מסוגל לעמוד בהחזרי המשכנתא שלו.
   - 6.2. ניתוח אשכולות – פילוח רשומות בבסיס נתונים לאשכולות של מידע, כך שבכל אשכול נמצאות רשומות בעלות מאפיינים דומים. לדוג' פילוח לקוחות של חברה.
   - 6.3. חוקי הקשר - מציאת קשרים בין פריטים שונים של נתונים – חוקיות בין שדות שונים בבסיס הנתונים, בעזרת חוקים אלו נוכל לחזות הימצאות של שדה אחד על פי השדה

האחר. דוגמא נפוצה לבעיה זו היא בעיית סידור מוצרי המכולת – על סמך שיטות של כריית מידע ניתן לקשר בין מוצרים הנקנים בחנות ביחד – לדוג' לחם וחלב. וכך ניתן לשים אותם בסמיכות אחד לשני במדף.

לכל אחת מהבעיות הנ"ל קיימים אלגוריתמים הפותרים אותם, כל בעיה נפתרת בשימוש בשיטה אחרת.

בעבודה מסכמת זו נעמיק בשיטה אחת של כריית מידע והיא ***כריית חוקי הקשר*** .

7. ***הסקת דפוסים ואגירת מידע*** – מהמידע שכרינו, ניתן להסיק דפוסי התנהגות, לאגור ידע חדש שלא היה ידוע קודם. בשלב זה ניתן לשמור את המידע ולפעול לפיו – בצורות שונות כגון : מיקוד עסקי, וכדו'.

2. חוקי הקשר – (Association Rules)

2.1 הקדמה

בחוקי הקשר, אנו מחפשים קשרים ויחסי גומלין בין עצמים בבסיס נתונים קיים. לאחר מציאת הקשרים ברצוננו להגדיר חוקים שיתארו את הקשרים הנ"ל. [32] [26] [35]

את הצורך בחוקי הקשר נבהיר ע"י הדוגמא הבאה : [13]

נניח ומנהל חנות גדולה למוצרי מזון רוצה ללמוד על הרגלי הקניה של הצרכנים שלו, בעיקר ברצונו לדעת : אילו קבוצות של מוצרים הלקוחות קונים בביקור נתון בחנות.

בכדי להשיג את המידע הנ"ל נצטרך לבצע תחקור של בסי ס הנתונים של החנות ולנתח את הרגלי הצריכה של הלקוחות, בעיקר בהיבטים של מה לקוח קונה בכל קניה בודדת. בעזרת המידע הזה נוכל לתכנן אסטרטגיות שיווקיות / פרסומיות.

ניתן יהיה ליישם בעזרת מידע זה מספר אסטרטגיות עסקיות :

מוצרים שנרכשים לרוב ביחד ימוקמו קרוב אחד לשני, בצורה זו נוכל להגדיל את כמות הקניות של שני המוצרים, לדוגמא אם גילינו שלקוחות שקונים קורנפלקס קונים לרוב גם חלב, נוכל למקם את שני המוצרים הללו סמוכים אחד לשני. כפועל יוצא מכך גם אנשים שלרוב לא קונים את שני המוצרים ביחד יקנו אותם ביחד הפעם.

כמו כן, ניתן להשתמש במידע זה בצורה שונה, מיקום של שני מוצרים שלרוב נרכשים ביחד בשני קצוות של החנות יגרום ללקוחות לעבור על פני כל החנות בחיפוש אחר המוצר. מעבר מסוג זה מגדיל את הסיכוי שלקוחות ירכשו מוצרים שלא התכוונו לרכוש.

שימוש עסקי נוסף של השיטה הזו יכול לבוא לידי ביטוי בהגדרה של מבצעים בחנות. אם נמכור שני מוצרים שממילא נרכשים ביחד ,במחיר זול (שניים במחיר אחד) סיכוי המוצרים הללו להימכר גדול יותר.

את המידע הזה נוכל להשיג בעזרת כריית מידע בשימוש בחוקי הקשר. חוקי ההקשר מגדירים את הקשר בין מוצרים ב"סל מוצרים" נתון. כלומר, בהנחה ובסל של הלקוח יש קורנפלקס יש גם (בסבירות מסוימת) חלב . ניתן בעצם לומר כי חוקי ההקשר מתארים קשרים בין עצמים במרחב עצמים נתון (במקרה שלנו בסיס הנתונים) [34].

מקובל לתאר את חוקי ההקשר ע"י מודל Market – Basket : [13]

קיימת כמות גדולה של עצמים ושל סלים – כל סל מכיל כמה עצמים.

ברצוננו למצוא קשרים בין החפצים בסל. כלומר בהינתן לנו שחפץ X נמצא בסל מה אנחנו יודעים על תוכן הסל? / מה אנחנו יודעים על חפץ Y בהינתן ש X בסל ?

החוק יסומן בצורה הבאה : $x \rightarrow y$ (בהינתן x נוכל לומר מה ההסתברות ש y יופיע גם בסל).

דוגמא :

נתונה לנו טבלא (טבלה 2.1) [32], המתארת הרגלי קניות בחנות מסוימת, לפי לקוח. בעל החנות יהיה מעוניין ליצור קשרים בין המוצרים בחנות : לדוג' מי שקונה חלב בהכרח יקנה גם קורנפלקס. (בהקשר של המודל – הסל הוא הלקוח , והמוצרים הם החפצים..)

TID	Items
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

טבלה 2.1 - דוגמא לבסיס נתונים [32]

חוקי ההקשר התואמים יראו כך :

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\}$

$\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\}$

$\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\}$

כלומר בהינתן שלקוח מסוים קנה חיתול נוכל לומר (בהסתברות מסוימת) כי הוא קנה גם בירה.
כפי שהסברנו, מוכר נבון יוכל למנף את הידע הנ"ל לצרכיו העסקיים ולמקם את מדפי הבירה בסמוך למדפי החיתולים.
כפי שכבר ציינו לעיל, חוקי ההקשר **אינם** מתארים סיבה ותוצאה, אלא מתארים קשרים בין עצמים.

2.2 הגדרות

- נניח [13] כי $j = \{i_1, i_2, i_3 \ldots, i_m\}$ הינה קבוצה של עצמים, ונניח כי D הינו אוסף של תנועות. בכל תנועה T מוכלים עצמים כך שמתקיים $T \subseteq j$. כפי שכבר הוסבר כל תנועה משויכת למספר ייחודי TID. נניח כי A הינה קבוצה של עצמים. ניתן לומר כי תנועה T תכיל את A אם ורק אם $A \subseteq T$.
  חוק הקשר הינו בעצם גרירה לוגית מהצורה $A \Rightarrow B$ כאשר $A \subset j$ וגם $B \subset j$, ומתקיים כי $A \cap B = \phi$.
- Itemset – קבוצה של עצמים (אחד או יותר).
- K-itemset – קבוצה בעלת k עצמים.
- Support Count – כמות ההופעות של קבוצה מסוימת בבסיס נתונים נתון.
- ***Support*** – כשאנו מדברים על רמת תמיכה של חוק הכוונה היא למדד שמאפשר לנו לדעת עד כמה החוק רלוונטי. לדוגמא : במידה ונתון לנו כי רמת התמיכה של חוק מסוים

היא 2% אזי הכוונה היא ש 2% מתוך כלל התנועות שנבדקו בבסיס הנתונים הנתון מקיימות את החוק הנ"ל. במילים אחרות מדד זה מאפשר לנו למדוד את רמת התפוצה של החוק הנתון. בצורה פורמלית ניתן לומר כי: חוק הקשר $A \Rightarrow B$ מתקיים עבור אוסף תנועות D עם תמיכה s. כוונת משפט זה היא שאחוז התנועות ב D המכילות את A U B[1], הינו s. ה support מהווה בעצם את "התמיכה" לחוק הנתון. ככל שהחוק יותר נפוץ ה"תמיכה" בו תעלה. בהינתן חוק הקשר $a \rightarrow b$, $support(a \rightarrow b) = P(a \cup b)$.
עבור קבוצת עצמים I התמיכה תוגדר כאחוז התנועות המכילות את העצמים המוכלים ב I. בעצם נוכל לומר כי התמיכה היא מספר התנועות המכילים את A ו B לחלק למספר התנועות הכולל[2].

Confidence – כשאנו מדברים על רמת ביטחון של חוק גם כאן הכוונה היא למדד שמאפשר לנו לדעת עד כמה החוק רלוונטי. במקרה זה הפרמטר אינו דוגם את רמת התפוצה של החוק בבסיס הנתונים אלא את מידת הדיוק של החוק. לדוגמא: במידה ונתון לנו כי רמת הביטחון של חוק מסוים (לדוג: $A \rightarrow B$) הינה 60% אזי ניתן לומר כי החוק נכון ב 60% מהמקרים.
כלומר בהינתן לי עובדת קיומו של אובייקט A בקבוצה מסויימת בבסיס הנתונים, ניתן לומר ברמת ביטחון של 60% כי גם B יופיע באותה קבוצה. בצורה פורמלית: אם נתון כי לחוק מסוים יש רמת ביטחון c (confidence c). משמעות משפט זה היא שעבור קבוצת תנועות D , c יהווה את אחוז התנועות המקיימות את החוק, כלומר בהינתן חוק הקשר R: $a \rightarrow b$, הביטחון שלנו בחוק הנ"ל הוא ההסתברות המותנה P (B|A), כלומר בהינתן שקיים A בקבוצה, מה ההסתברות שגם B קיים בה. בהינתן חוק $A \Rightarrow B$ נוכל לומר כי $confidence(a \rightarrow b) = P(b|a)$. אחוז האמון מביע לנו עד כמה החוק נכון, באיזה רמת מובהקות נוכל לומר שבמידה ותנועה מכילה את A היא מכילה גם את B.
בעצם ניתן יהיה לומר כי ה"ביטחון" עבור חוק מסוים יהיה מספר התנועות המכילות גם את A וגם את B לחלק למספר התנועות המכילות את A.
כמו כן נוכל להביע את אחוז הביטחון בשימוש בפרמטר התמיכה[3]:

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{Support(A \rightarrow B)}{Support(A)}$$

.

$$Confidence(a \rightarrow b) = \frac{Support(A \rightarrow B)}{Support(A)}$$

---

[1] הכוונה היא לתנועות המכילות **גם** את A ו**גם** את B. מכיוון שהחיתוך בין A ל B הוא קבוצה ריקה לא ניתן היה להשתמש בפרמטר החיתוך.

[2] בחלק מהמקורות שנסקרו פרמטר התמיכה נמדד ע"י מספר המופעים בבסיס הנתונים ולא באחוזים, וזאת מכיוון שמספר התנועות הכולל בד"כ ידוע.

[3] ניתן לומר כי $P(A \cap B)$ שקול ל $Support(A \Rightarrow B)$ וזאת מכיוון שהתמיכה מתארת את החוקים שגם A וגם B מצויים בהם. כפי שהוסבר בהערה 1 השימוש במושג החיתוך כאן יוצר בלבול, אבל הכוונה היא לחוקים שגם מכילים את A וגם מכילים את B.

שילוב שני החוקים הללו, מאפשר לנו לבקר את תהליך יצירת החוקים, ולהחליט בצורה מושכלת האם חוק מסוים הינו תקף או לא.

ע״י מדד ה support נוכל לדעת עד כמה נפוץ החוק שגילינו וע״י מדד ה confidence נוכל לדעת עד כמה הוא מדויק.

כמו כן מקובל להגדיר ערך מינימאלי עבור כל אחד מהמדדים הנ״ל. חוקים עם ערכי מדדים נמוכים מהערך המינימאלי לא יילקחו בחשבון. חוקים שיעמדו בשתי הדרישות של החסמים התחתונים של מדדים אלו ייקראו *״חוקים חזקים״* (Strong rules) [13]

מהקשר בין שני המדדים ניתן לראות כי קיים קשר ישיר בין רמת התמיכה לבין רמת הביטחון. ככל שרמת התמיכה עולה כך עולה גם רמת הביטחון.

יש לציין שלחוקי הקשר שנוצרו מאותו itemset תהיה אותה תמיכה אך רמת הבטחון שלהם יכולה להיות שונה.

<u>דוגמא:</u>

עבור טבלה 2.1 – עבור החוק: $\{milk, diaper\} \rightarrow \{beer\}$

$$\textbf{\textit{Support(I)}} = \text{Support}(\{milk, diaper\} \rightarrow \{beer\}) = P(\{beer\} \cup \{milk, diaper\})$$

כלומר כעת עלינו למצוא את ההסתברות של הופעת כל הפריטים האלו ביחד.

מכיוון שכל המוצרים הללו מופיעים ביחד רק ב 2 תנועות מתוך ה 5 הקיימות בטבלה 2.1, נוכל לומר כי ההסתברות להופעתן היא: $2/5 = 40\%$

$$P(\{beer\} \cup \{milk, diaper\}) = 2/5 = 40\%$$

לכן התמיכה עבור החוק הנ״ל היא 40% משמעות נתון זה היא שהחוק מתקיים רק ב 40% מהתנועות הקיימות בבסיס הנתונים. לכן ייתכן מצב שישנן תנועות שמכילות את $\{milk, diaper\}$ אך אינן מקיימות את החוק הזה (לדוג׳ תנועה 5).

$$\textbf{\textit{Confindence}}(\{milk, diaper\} \rightarrow \{beer\}) = P(\{beer\} \mid \{milk, diaper\})$$

ומכיוון שישנן 3 תנועות המכילות $\{milk, diaper\}$ ורק 2 מתוכן מכילות גם את beer ניתן יהיה לומר כי רמת האמון שלנו בחוק היא $2/3 = 66\%$

$$P(\{beer\} \mid \{milk, diaper\}) = 2/3 = 66\%$$

ניתן היה להגיע לאותה תוצאה בצורה אחרת (כפי שהוסבר בהגדרה של מדד האמון):

$$P(\{beer\} \mid \{milk, diaper\}) = \frac{P(\{milk, diaper\} \cap \{beer\})}{P(\{milk, diaper\})} = \frac{Support(\{milk, diaper\} \Rightarrow \{beer\})}{Support(\{milk, diaper\})} =$$

$$\frac{2/5}{3/5} = 2/3 = 66\%$$

כלומר החוק מתקיים רק ב 40% מהתנועות, ויש לו 66% דיוק.

אחוז התמיכה מראה כי החוק הזה התקיים ב 40% מקרב הקונים, כלומר 40% מקרב הקונים בחנות קנו גם בירה גם חיתולים וגם חלב.

כמו כן ע״י אחוז האמון נוכל לומר שבהינתן לנו כי לקוח קנה חלב וחיתולים ישנו סיכוי של 66% שהוא קנה גם בירה.

עבור החוק {milk, beer, diaper} → {milk} התמיכה תהיה בדיוק כמו החוק הקודם: {milk, diaper} → {beer}. אך רמת האמון תהיה: $\frac{2/5}{4/5} = 2/4 = 50\%$

- ***תמיכה מינימאלית (Minimum Support Count)*** - קבוצת עצמים(itemset) תקיים את פרמטר התמיכה המינימאלית אם כמות המופעים שלה בבסיס הנתונים גדולה או שווה לפרמטר התמיכה המינימאלית שהוגדר. בהמשך יוסבר השימוש במושג הנ"ל יותר בהרחבה.
- ***קבוצת עצמים תדירה- Frequent itemset*** –itemset שעומד בדרישה של "תמיכה מינימלית", כלומר ה support שלו גדול יותר מהחסם המינימלי ל support. ייקרא Frequent itemset ויסומן ב L_k כאשר k יסמן את גודל הקבוצה.

<u>דוגמא מסכמת</u> [28]

בדוגמא זו נבהיר ונדגים בצורה אחודה את כל המושגים שהוסברו לעיל.

נתון לנו בטבלה 2.2 א' בסיס הנתונים הבא [32]:

TID	A	B	C	D	E
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

טבלה 2.2 א' - בסיס הנתונים

כפי שהסברנו itemset הינה קבוצת עצמים לדוג: I = {A,B,E}

תנועה היא שורה בבסיס הנתונים (מסומנת ע"י TID - מס' סידורי של השורה בבסיס הנתונים), לרוב שורה זו מייצגת פעולה כלשהיא.

נבהיר מעט את צורת הייצוג של טבלה 2.2 א':

טבלה 2.2 א' הינה טבלה בינארית כלומר, בכל עמודה ה 0 ו ה 1 מייצגים את קיומו / אי קיומו של העצם בסל. מבחינה טבלאית הטבלה מקבילה לטבלה הבאה (טבלה 2.2 ב'):

TID	
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABC

טבלה 2.2 ב' - בסיס הנתונים

בסיס הנתונים הנ״ל מייצג בדומה לטבלה 2.1 אוסף של סלים של עצמים.

כעת בהינתן לנו ה itemset $I = \{A,B,E\}$ יש למצוא את חוקי ההקשר שעבורים התמיכה המינימלית[4] היא 2 ואחוז האמון הינו 50%.

את מס׳ החוקים האפשריים נחשב בצורה הבאה:

קיימים 3 מועמדים (A,B,E) לבחירה בצד השמאלי של החוק. בהנחה ובחרנו בתת קבוצה מתוך A,B,E הצד הימני של החוק יהיה מי שלא נבחר לצד השמאלי.

עבור כל אות קיימות 2 אפשריות או שהיא קיימת בצד שמאל של החוק או שלא. מכיוון שישנן 3 אותיות נקבל כי סך כל האפשריות הוא $2^3 = 8$ מסך כל האפשרויות נוריד את האפשרות $\{\} \rightarrow \{A,B,E\}$ שכן היא איננה חוקית. את האפשרות $\{A,B,E\} \rightarrow \{\}$ נוכל להפוך ל $\{A,B,E\} \rightarrow true$. סה״כ קיימות 7 אפשרויות.

כעת נסביר את צורת הבנייה של טבלה 2.2 ג׳:

לאחר שיצרנו את כל סוגי החוקים האפשריים והצבנו אותם בטבלה. נבדוק עבור כל חוק אפשרי את אחוזי התמיכה והביטחון.

בכדי למצוא את רמת התמיכה יהיה עלינו לחזור עבור כל חוק אל עבר בסיס הנתונים ולראות בכמה מקרים הוא מתקיים. לדוג׳ החוק $\{A,B\} \rightarrow \{E\}$

מתקיים בשני מקרים בלבד (שורה 1 ו 8 בבסיס הנתונים בטבלה 2.2 ב׳). לכן התמיכה היא 2.

את התמיכה נחשב בצורה הבאה:

נחשב את:

$$\textbf{\textit{Confindence}}(\{A,B\} \rightarrow \{E\}) = P(\{E\} \mid \{A,B\}) = 2/4 = 50\%$$

בטבלה 2.2 ג׳ מוצג סיכום רשימת החוקים שהתקבלו.

[4] כפי שהוסבר בהערה 5 התמיכה כאן מובאת בתור מספר ולא באחוזים.

A	B	E	החוק	תמיכה – support (מתוך 9)	אמון- confidence
1	1	0	{A,B}→{E}	2	2/4=50%
1	0	1	{A,E}→{B}	2	2/2= 100%
1	0	0	{A }→{B,E}	2	2/6=33%
0	1	1	{ B,E}→{A}	2	2/2=100%
0	1	0	{ B }→{A,E}	2	2/7=28%
0	0	1	{E}→{A,B}	2	2/2=100%
0	0	0	{true}→{A,B,E}	2	2/9=22%

טבלה 2.2 ג' – סיכום רשימת החוקים

מכיוון שכל החוקים נוצרו מאותו itemset רמת התמיכה שלהם זהה.

חוקים שלא עמדו בדרישת האמון המינימאלי סומנו בצבע אפור.

לקבוצת האיברים (Itemset) {A,B,E} נוכל לקרוא frequent itemset כיוון שקבוצה זו עמדה בדרישות התמיכה המינימאלית.

2.3 כריית מידע מבסיס נתונים חד מימדי בוליאני

חוקי הקשר בוליאניים[5] הם מהצורה הבאה:

Computer→Anti_Virus_Software

חוק זה נקרא בוליאני שכן הוא עוסק בקיומו /אי קיומו של עצם – אנו נבדוק האם עבור לקוחות שקנו מחשב נקנתה גם תוכנת אנטי וירוס.

חוקי הקשר חד מימדיים הם מהצורה הבאה:

Buys(X,Computer)→Buys(X, Anti_Virus_Software)

חוק זה נקרא חד מימדי מכיוון שיש לו רק פרדיקט אחד – Buys.

במקרה זה חוק זה הינו גם חוק בוליאני שכן גם פה הוא עוסק בקיומו / אי קיומו של עצם.

בסיס נתונים חד מימדי בוליאני הינו מהצורה של טבלה 2.2 א' שהוצגה לעיל. בפרק זה נעסוק בכריית חוקי הקשר מהצורה הנ"ל.

תהליך כריית המידע בשיטת חוקי הקשר מתחלק לשני חלקים:

1) מציאת כל תתי הקבוצות התדירות (frequent itemsets) מבסיס הנתונים.

2) חילול חוקי הקשר חזקים מהקבוצה הנ"ל.

בפרק הבא יוצגו אלגוריתמים שונים למימוש שיטות אלו.

[5] בעבודה זו נתמקד בסוג מסויים של חוקי הקשר – חד מימדי ובוליאני הרחבה לגבי סוגים נוספים (רב מימדי ולא בוליאני) ניתן לראות ב סמינר ב[39]. באופן עקרוני וכללי יש להמיר את המידע הקטריגוריאלי / כמותי לטווחים ואז לייצר משתנים בוליאנים עבור הטווחים הנ"ל.

3. כריית חוקי הקשר – אלגוריתמים שונים

מכיוון שבתהליך כריית חוקי הקשר עיקר הבעיה האלגוריתמית היא מציאת תתי הקבוצות התדירות. האלגוריתמים לכריית חוקי הקשר עוסקים רק בפיתרון בעיה זו.

3.1 מציאת קבוצות תדירות (*Frequent Itemsets*)– האלגוריתם הנאיבי

הדרך הנאיבית [32] [2] למציאת כל תתי הקבוצות שהן תדירות, הינה סריקה סדרתית של כל התנועות בניסיון למצוא את כל תתי הקבוצות. תהליך המציאה יתחלק לשני חלקים :

1) מציאת כל ה itemsets

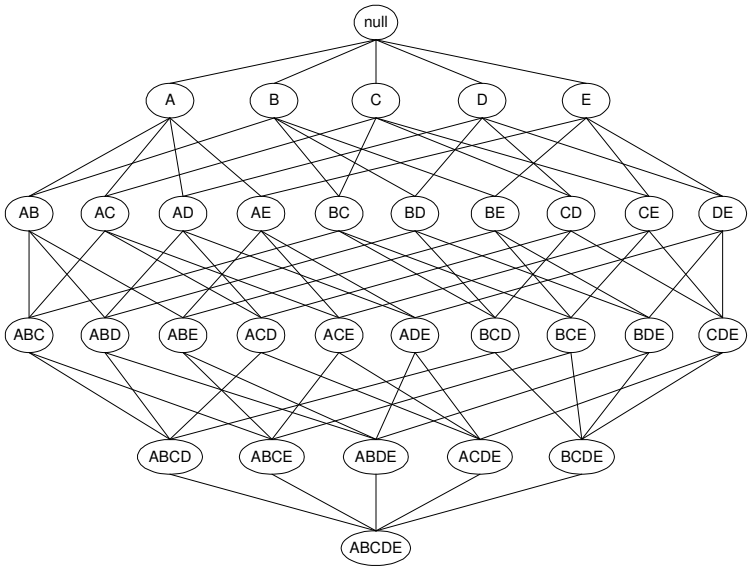
2) סינון ה itemsets שאינם עומדים בקריטריון "התמיכה המינימלית".

בהינתן d עצמים קיימים 2^d itemsets , בכדי למצוא את כולם יהיה עלינו לבנות רשת כדוגמת זו שבאיור 2.3.1 א'

לא נעשית בניה בפועל של הרשת, הרשת רק ממחישה את היקף ה Itemsets הדרושים בצורה ויזואלית נוחה לעין. הרשת מראה את היקף הגודל המעריכי של 2^d עצמים.

תהליך מציאת כל תתי הקבוצות האפשרויות (כל ה itemsets האפשריים) יתרחש בצורה הבאה :

מכיוון שישנם d עצמים ועבור כל עצם קיימת האפשרות של קיים / לא קיים יש לנו 2^d אפשרויות לתתי קבוצות, לכן הזמן שייקח לייצר את כל תתי הקבוצות הנ"ל הינו גם 2^d. לאחר שנוצרו כל תתי הקבוצות נצטרך לעבור עליהן אחת אחת ולבדוק האם כל קבוצה עומדת בתנאי התמיכה המינימאליים שהוגדרו.



איור 3.1 כל האפשריות ל תתי קבוצות(itemsets) עבור 5 עצמים [32]

באיור 3.1 א׳ ניתן לראות את אופן בניית כל תתי הקבוצות:

נדגים את אופן הבניה:

עבור כל אחד מהאיברים בקבוצה ניצור זוג עם כ״א משאר האיברים לדוג׳ : AB,AE,AC,AD

וכן הלאה, בסופו של דבר ייווצרו לנו 10 קבוצות של זוגות.

כעת עבור כל איבר בקבוצת הזוגות נצרף עוד איבר שלא נכלל מקודם ונהפוך את הזוג לשלשה.

לדוג׳ מ AB יווצרו לנו ABC ABD ABE כך נקבל 10 שלשות.

כעת שוב נעבור על כל אחת מהשלשות הללו ונוסיף לכל שלשה עוד איבר שלא היה בקבוצה לפני כן ונקבל 5 רביעיות , מהרביעיות ניצור חמישייה אחת.

סך הכל 2^5 תתי קבוצות.

כעת לאחר שיצרנו את כל תתי הקבוצות יהיה עלינו לבדוק עבור כל קבוצה האם היא עומדת במדדי התמיכה המינימאלית שהוגדרו. כלומר שוב נעבור קבוצה קבוצה, עבור כל קבוצה נספור את מספר ההופעות שלה בכל התנועות בבסיס הנתונים ונבדוק האם מספר זה גדול /שווה לתמיכה המינימאלית שהוגדרה. אם כן נוכל לומר כי תת קבוצה זו הינה תת קבוצה תדירה.

סה״כ זמן ריצה (בהנחה שיש M תתי קבוצות, N תנועות ו w עצמים בכל טרנסקציה) O(MNw)

אך מכיוון ש $M=2^d$, נקבל שכמות הזמן היא מעריכית.

מה שמוביל אותנו לחפש אלגוריתם יעיל יותר למציאת תתי הקבוצות התדירות.

3.2 מציאת קבוצות תדירות (*Frequent Itemsets*) – אלגוריתם אפריורי

באלגוריתם אפריורי [6][13][32] נעשה ניסיון לתקוף את בעיית המעריכות של האלגוריתם למציאת תתי קבוצות תדירות ע"י הקטנה של מספר המועמדים, כך לא ניתקל בסיבוכיות מעריכית בשיטה זו. [1]. כפי שציינו לעיל, תהליך כריית המידע כולל בתוכו שלב של מציאת כל תתי הקבוצות התדירות. ומכיוון שמספרן הינו מעריכי נתקלנו בסיבוכיות מעריכית.

באלגוריתם אפריורי - כבר בשלב הראשוני של כריית חוקי ההקשר(מציאת כל תתי הקבוצות התדירות) אנו מקטינים את מספר הקבוצות שיתקבלו, כך מספר הקבוצות שיעברו עיבוד ע"י האלגוריתם יקטן. וממילא סיבוכיותו של האלגוריתם תקטן גם כן.

האלגוריתם מתבסס על תכונת האפריורי הבאה[7]: כל תת קבוצה של קבוצה תדירה () היא גם תדירה. כלומר בהינתן לנו קבוצת עצמים תדירה (כלומר, קבוצה שמספר הופעותיה גדול מהמינימום שהוגדר) כל תת קבוצה של הקבוצה הנ"ל הינה תדירה גם כן.

לדוגמא: נתון כי I הינה קבוצה תדירה.

$I = \{A, B, E\}$

נוכל לומר כי כל תת קבוצה של I הינה תדירה גם כן כלומר גם הקבוצה $G=\{A,B\}$ הינה קבוצה תדירה.

כמו כן בכיוון ההפוך בהינתן לנו שקבוצת עצמים איננה תדירה כל קבוצה שמכילה אותה גם לאט תהיה תדירה כלומר:

אם נתון כי $I = \{A,E\}$ הינה קבוצה **לא** תדירה אזי יצירה של $G=\{A,B,E\}$ ע"י הוספה של העצם E לקבוצה הנ"ל לא תהפוך אותה לתדירה.

הטענה הנ"ל מסתמכת על העובדה הבאה[28][28]:

$$\forall X,Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

ובמילים: עבור כל 2 תתי קבוצות X ו Y במידה ומתקיים כי X מוכל ב Y אזי התמיכה של X גדולה או שווה לתמיכה של Y[13].

כלומר לתת קבוצה יש לכל הפחות תמיכה כמו הקבוצה המכילה אותה. נתון זה קל להוכחה:

נניח ש $X \subseteq Y$. מכיוון שב"תמיכה" של Y נכללות אך ורק קבוצות שמכילות גם את X וגם את Y, בהכרח נוכל לומר שהתמיכה של X גדולה או שווה לתמיכה של Y. מכיוון שבתמיכה של Y לא נכללו קבוצות המכילות את X. אך בתמיכה של X יכללו גם קבוצות המכילות רק את X וגם קבוצות המכילות את Y.

כעת בשימוש בידע שרכשנו מעיקרון האפריורי ניגש לאלגוריתם ונראה כיצד יישום העיקרון הנ"ל מקטין את מספר המועמדים שעלינו לסרוק בצורה ניכרת.

---

[6] לא נרחיב ממש על אלגוריתם אפריורי. אלגוריתם אפריורי נסקר בצורה מקיפה ויסודית ב [39]

[7] תכונה זו קרויה בספרות – תכונות האפריורי.

האלגוריתם [23]

מונחים:

C_k - קבוצת עצמים בגודל k , קבוצת העצמים הנ"ל מועמדת בשלבי האלגוריתם להיות קבוצת עצמים תדירה (במידה ותעמוד בתנאי התמיכה המינימאלית).

L_k – קבוצת עצמים תדירה בגודל k.

תיאור האלגוריתם באופן כללי

האלגוריתם מורכב משני חלקים צירוף (join) וגיזום (prune).

הרעיון המרכזי של האלגוריתם הוא שאת כל תתי הקבוצות התדירות נבנה באופן הדרגתי אחת מתוך השניה. כלומר קבוצה תדירה (Frequent Itemset) בגודל k תיבנה ע"י קבוצה תדירה בגודל k-1. כך נעשה במהלך האלגוריתם שימוש בתוצאות משלבים הקודמים של האלגוריתם בכדי לחסוך זמן חישוב.

כאשר בחלק ה ***איחוד*** נבנה itemset(k) בשימוש של itemset(k-1). כלומר C_k יבנה ע"י צירוף של L_{k-1} עם עצמו. בעזרת הצירוף נוכל ליצר את כל תתי הקבוצות הקיימות עבור כל קבוצת עצמים נתונה.

בחלק ה ***גיזום*** – נסיר מ itemset(k) כל itemset שהוא לא תדיר (עבור כל תתי הקבוצות האפשריות) . וזאת נעשה ע"י שימוש בתכונת האפריורי, כל תתי הקבוצות של itemset תדיר הן תדירות, לכן אם אחת מתתי הקבוצות לא מופיעה ב L_{k-1} אזי נוכל לומר כי הקבוצה המכילה אותה גם אינה תדירה, וכך חסכנו לעצמנו חיפוש ארוך וספירת support count עבור כל סוגי תתי הקבוצות. כלומר מכיוון שכל תת קבוצה מורכבת מאיחוד של תתי קבוצות אחרות. במידה ואחת מהן לא תדירה נוכל לדעת מראש שכל תת קבוצה שנוצרה ממנו תהיה בוודאות לא תדירה. כך בשלב הגיזום נדע מראש שכלל תת קבוצה שיכולה להיווצר מתת קבוצה שאינה תדירה לא תהיה תדירה. לכן אנו מסירים (גוזמים) את תתי הקבוצות שאינן תדירות ממרחב תתי הקבוצות שלנו וכך לא נוכל להשתמש בהן יותר וליצור מהן תתי קבוצות נוספות.

לכן כעת , לפני החיפוש הממצה בבסיס הנתונים אחר מופעים של itemset, נפעיל את עיקרון האפריורי על רשימת המועמדים הנוכחית.וכך נוכל לצמצם באופן משמעותי את גודל הרשימה עוד לפני התחלת תהליך החיפוש.

ניתן לומר שבעצם שימוש בתכונת האפריורי, מהווה מעין סוג של "שיפור ביצועים" עבור האלגוריתם, כי כעת ידרשו פחות צעדים ע"מ לקצץ את C_k.

Pseudo-code [23]:

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

1.L_1 = {frequent items};

2. **for** (k = 1; L_k !=∅; k++) **do begin**

2.1 C_{k+1} = candidates generated from L_k;

2.2 Prune candidate itemsets containing subsets of length k that are infrequent

2.3 **for each** transaction t in database do

2.3.1 increment the count of all candidates in C_{k+1} that are contained in t

2.4 **end**

2.5L_{k+1} = candidates in C_{k+1} with min_support

3. **end**

4. **return** $\cup_k L_k$;

כפי שצוין לעיל :

C_k - קבוצת עצמים בגודל k , קבוצת העצמים הנ"ל מועמדת בשלבי האלגוריתם להיות קבוצת עצמים תדירה (במידה ותעמוד בתנאי התמיכה המינימאלית).

L_k – קבוצת עצמים תדירה בגודל k.

שורה 1 - בשלב הראשון של האלגוריתם מחוללים באופן עצמאי את L_1 כלומר את כל ה itemsets בגודל 1 שהם תדירים. תהליך זה מבוצע ע"י מעבר על בסיס הנתונים וספירה של מס' התנועות בהן מופיע כל עצם.

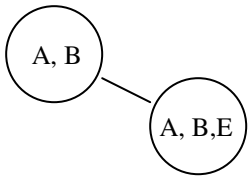
שורה 2.1 – כעת מגיע שלב ***האיחוד*** מ L_1 נייצר את C_2 ע"י פעולת איחוד טבלאית (איחוד - join[8]) בין L_1 לעצמו.

[8] אנו מניחים כי הקורא מכיר את פעולת האיחוד - Join הטבלאית, בקצרה נוכל לומר כי פעולה זו היא מעין מכפלה בין שתי טבלאות, כך שנוצרת טבלה שלישית,למידע נוסף ניתן לעיין ב : http://www.w3schools.com/sql/sql_join.asp.

שורה 2.2 - כעת מגיע שלב ***הקיצוץ*** בשלב זה נסנן (עפ״י עיקרון אפריורי) את כל ה itemsets בגודל k (בהתחלה k=1)שאינם תדירים. מובטח לנו שכל ה itemsets הנ״ל לעולם לא יהוו חלק מקבוצת עצמים תדירה, וכך אנו לא מפסידים בעצם שום מידע.

באנלוגיה לדיאגרמת העץ באיור3.1 נוכל לומר שאנו מקצצים צמתים מהעץ ואת כל הצאצאים שלהם. בניסוח אחר נוכל לומר כי אנו קוצצים תתי עצים המושרשים ע״י הצומת שאינו תדיר. כי צומת שאינו תדיר מובטח לנו שלעולם הוא ובניו לא יהיו תדירים (עפ״י עיקרון אפריורי). שוב נציין כי בפועל לא קיים עץ אלא אנו רק מציגים את המידע בצורת עץ על מנת להקל את ההבנה.

לדוגמא:



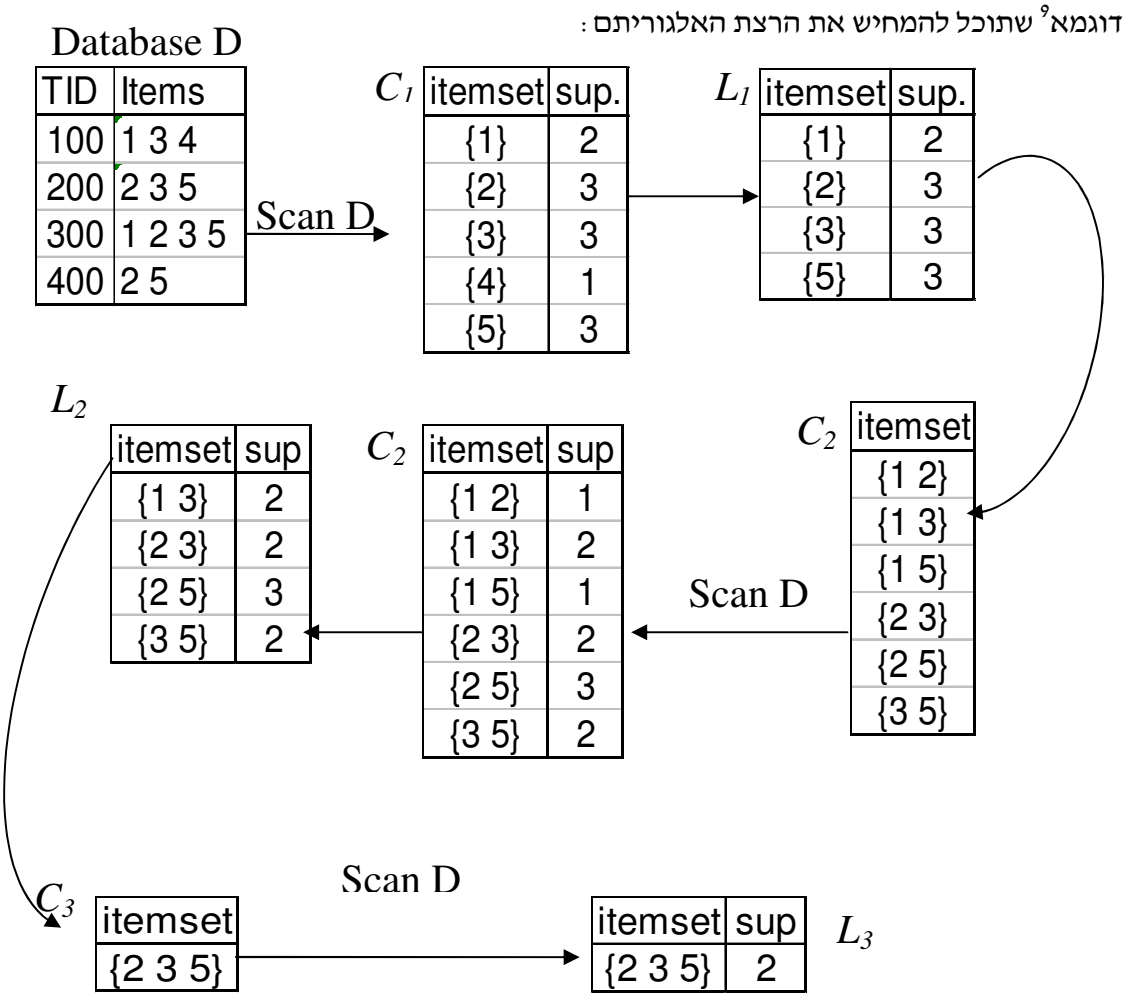
איור 3.2 א׳ דוגמא להפעלת אלגוריתם אפריורי

אם נתון לנו ש A,B הינה קבוצה לא תדירה, נוכל לומר בוודאות כי כל תת קבוצה שתיווצר ממנה תהיה לא תדירה, כלומר A,B,E אינה תדירה. בעצם אנו ״קוצצים״ את כל הצאצאים שיכולים להיווצר מ A,B. לדוגמא: בהנחה ובעץ באיור 3.1 הגענו למסקנה כי A,B אינה תדירה אזי נוכל לומר בוודאות כי {A,B,C},{A,B,D},{A,B,E} אינן תדירות וכך כל שאר הבנים שלהם בעץ הנ״ל.

שורות 2.3 - 2.4 - עבור כל תנועה בבסיס הנתונים, נעדכן את מספר המופעים של כל קבוצת עצמים ב C_{k+1}.

שורה 2.5 – מ C_{k+1} נוכל ליצור את L_{k+1}, בכך שנשאיר רק את המועמדים שעמדו בדרישות התמיכה המינימלית.

דוגמא[9] שתוכל להמחיש את הרצת האלגוריתם:



איור 3.2 ב' דוגמא להפעלת אלגוריתם אפריורי[23]

[9] איור מפורט נוסף המדגים את דרך פעילות האלגוריתם ניתן למצוא ב [13], עמ' 233.

<u>הסבר דוגמת ההרצה והאלגוריתם:</u>

בהינתן לנו קבוצה של itemsets (בהקשר של מודל הסל – מדובר על אוסף סלים של צרכנים) המצויה בתוך בסיס נתונים.

בתחילה יהיה עלינו לספור את מספר ה"סלים" שמכילים את כל אחד מסוגי העצמים, כלומר כמה מופעים בגודל 1 קיימים עבור כל סוג.

האלגוריתם מתחיל עם הקבוצה ההתחלתית C_1 שזוהי הקבוצה שמכילה את כל סוגי העצמים (כל סל / קבוצת עצמים בגודל 1)

לאחר מכן נפיק מתוך C_1 את L_1 – שזה כולל בעצם את כל העצמים ב C_1 שעומדים בתנאי התמיכה המינימלים. נוריד מ C_1 את כל העצמים שהם עצמים שאינם תדירים – וכך יורד {4}.

בכדי למצוא את C_2, נבצע Join ב L_1 ונקבל את רצף הזוגות המופיע בתרשים (מסומן ב C_2). יש לשים לב שה Join מתבצע בצורה הבאה:

עבור כל זוג מספרים (itemset) נבדוק כמה מופעים יש לזוג הנ"ל בבסיס הנתונים ונצמצם את אלו בעלי כמות מופעים נמוכה מהרצוי ({1 5} {1 2}).

כעת לאחר שחישבנו את L_2, נכפיל שוב את L_2 בעצמו ונקבל את C_3 = {2 3 5}.

C_3 מופיע פעמיים לכן הוא עומד בדרישות התמיכה המינימאליות.

וכאן מסתיים האלגוריתם, כלומר בעצם מצאנו כאן את קבוצת ה itemsets התדירה.

עבור כל C_k נוכל לדעת מה הקבוצה התדירה מכילה.

יש לציין שזהו רק השלב הראשון של האלגוריתם, עלינו עוד לחולל מנתונים אלו חוקי הקשר חזקים.

<u>ניתוח סיבוכיות לאלגוריתם אפריורי</u>

אלגוריתם אפריורי אינו שונה מהותית בתהליך חילול המועמדים שלו מהאלגוריתם הנאיבי. כל השיפור הינו רק בעובדה שמספר המועמדים אינו מעריכי הודות לשימוש בתכונות אפריורי. לכן בבואנו לנתח את הסיבוכיות של אפריורי נשתמש באותה דרך בה חושבה סיבוכיותו של האלגוריתם הנאיבי.

סה"כ זמן ריצה (בהנחה שיש M תתי קבוצות, N תנועות ו w עצמים בכל טרנסקציה) O(MNw) באפריורי , בשונה מהאלגוריתם הנאיבי M אינו מעריכי. לכן לא תתקבל סיבוכיות מעריכית. אך ישנם מקרים גרועים שבהם גם אפריורי ירוץ בסיבוכיות מעריכית.

<u>חסרונותיו של אלגוריתם אפריורי</u>

למרות היעילות הרבה של אפריורי (לעומת האלגוריתם הנאיבי) קיימות [16] [13] שתי בעיות עיקריות באלגוריתם:

- נניח ויש מס' גדול מאוד של תתי קבוצות תדירות בגודל 1. אזי מספר תתי הקבוצות בגודל 2 יהיה גדול יותר (בסדר גודל מעריכי). לדוגמא: בהינתן בסיס נתונים ובו 10^4 תתי קבוצות תדירות בגודל 1. מספר תתי הקבוצות לבדיקה בגודל 2 יהיה יותר מ 10^7(!!!).

יתירה מזו בכדי לגלות עצמים תדירים בקבוצת נתונים בגודל 100 נצטרך לחולל 1- 2^{100} ~

[30] 10 תתי קבוצות ולבדוק אם הן תדירות. כלומר ניתן לראות שעדיין קיימת תלות מעריכית בין מספר המועמדים לזמן הריצה של אלגוריתם אפריורי

- תהליך בדיקה התמיכה עבור התנועות בבסיס הנתונים באלגוריתם אפריורי, לא יעיל ומצריך מעבר על כל התנועות אחת לאחת בכדי לחשב את התמיכה. מעבר סדרתי זה מהווה אבן נגף בדרכינו לשיפור הביצועים של כריית חוקי ההקשר

חסרונות אלו הובילו למציאתם של אלגוריתמים יעילים יותר, אלגוריתמים אלו יפורטו בפרקים הבאים.

FP-Growth 3.3

כפי שהסברנו, למרות השיפור שמציג אלגוריתם אפריורי בתחום חילול המועמדים. עדיין בכדי לכרות חוקי הקשר אנו נדרשים לחולל מספר רב של מועמדים [1][16]

שיטת ה FP Growth [19] משתמשת בטקטיקת "הפרד ומשול" בצורה הבאה[1]:

- נדחוס את בסיס הנתונים ונמיר את צורתו ל FP- Tree. נציין כי מדובר כאן בעץ תחיליות (Prefix tree).
- לאחר מכן נחלק את העץ הנ"ל לתתי בסיסי נתונים מותנים. מתבצע שיוך חח"ע בין כל תת בסיס נתונים לקבוצה תדירה ומתבצעת כרייה עצמאית על תת בסיס הנתונים הנ"ל.

הייחוד[1][16] של מבנה הנתונים FP Tree, הינו בקומפקטיות שלו, מכיוון שרק לעצמים בעלי תמיכה מינימלית של 1 לפחות יהיה ייצוג ע"י צומת בבסיס הנתונים. בצורה זו לא נצטרך להשקיע משאבים בעצמים שאינם יכולים להועיל[10] לתהליך הכריה. בתהליך הבניה של העץ נעשית אופטימיזציה נוספת, המוודאת כי העצמים הפחות תדירים יהיו בעלים והיותר תדירים בצמתים, כך נוכל לעשות בעצמים התדירים שימוש חוזר בכמה תנועות בבסיס הנתונים. בדיקות שנעשו העלו כי סיבוכיות המקום של העץ הינה קטנה בסדרי גודל מסיבוכיות המקום של בסיס הנתונים.

[10] כמובן, שאנו מניחים כי לרוב ברצונו של הכורה הוא בעצמים תדירים לפחות בגודל 1.

3.3.1 מבנה העץ ותהליך הבניה

בהינתן[1][16][14] רשימת עצמים $I = \{a_1, a_2, ..., a_n\}$ ובהינתן בסיס נתונים:

$DB = \{T_1, T_2, ..., T_n\}$.

ξ יסמן את התמיכה המינימלית.

בכדי לבנות את העץ ניקח בחשבון את ההנחות הבאות:

1. מכיוון שרק איברים בעלי תמיכה מינימלית של 1 יוכנסו לעץ, יש לבצע סריקה מקדימה של כל בסיס הנתונים DB בכדי לספור את כמות המופעים של כל אחד מהעצמים בבסיס הנתונים.
2. במצב של חזרות על אותו צומת בעץ, נוסיף פרמטר שיציין את מספר הפעמים שנעשה שימוש בצומת הנ"ל. שם הפרמטר *count*. ניתן כעת לומר בקלות האם שתי תתי קבוצות הינן זהות. וזאת מכיוון שכעת תתי הקבוצות ממוינות על פי התמיכה שלהן (יודגם בהמשך).
3. כפי שציינו לעיל, תנועות בבסיס הנתונים בעלות תחיליות משותפות יחלקו את אותו נתיב בעץ. נציין כי יש לעדכן את המונה המצורף לכל צומת במסלול המשותף במקרה זה.

תהליך הבניה יהיה כדלקמן:

לאחר סריקה אחת מלאה של DB נוכל לקבל רשימה של כל העצמים בצירוף התמיכה שלהם:

$\langle (i_1, \sup_1), ... (i_k, \sup_k) \rangle$. לאחר מכן נבצע מיון של כלל הזוגות הסדורים הנ"ל. כמו כן נוודא כי עבור כל רשומה בבסיס הנתונים, העצמים מסודרים בסדר הנ"ל (הממוין).

כעת ניצור עץ ששורשו הוא null ונתחיל לשבץ את הרשומות לפי הסדר. השיבוץ יעשה ע"י הוספה לעץ של איברי הרשומות. כל רשומה חדשה תתוסף כבן של השורש. נזכיר כי מכיוון שהרשומות ממוינות על סמך התמיכה המינימלית שלהן, הרשומות יחלקו בינן מסלולים משותפים בעץ. במקרה שכזה נקדם את המונה המצורף לכל צומת בעץ ב 1.[11]

בצורה זו יצרנו עץ קומפקטי בעל חיסכון בסיבוכיות מקום. בכדי להקל על סיור בעץ, ניצור טבלה המכילה כניסה עבור כל אחד מהעצמים בבסיס הנתונים. טבלה זו תהווה ראש של רשימה מקושרת לכל המופעים של האיברים מסוג זה בעץ. כך לדוג' עבור i_k יהיה מצביע שיתחיל מהטבלה ויעבור בכל הצמתים שבהן i_k מופיע.

בצורה פורמלית נוכל לומר ש עץ FP הינו עץ המוגדר כך:

1. מכיל שורש המסומן ב null, תתי עץ תחילייים בתור בנים של השורש. כמו כן מבנה הנתונים מכיל טבלה לגישה מהירה לנתונים.
2. כל צומת מכיל שלושה חלקים:
   a. שם
   b. מונה

[11] מומלץ לעבור על הדוגמא בהמשך בכדי להבין את האלגוריתם לעומק.

c. מצביע לקישור לצומת הבאה ברשימה המקושרת.

3. כל כניסה בטבלת הגישה המהירה מכילה שני שדות:

a. שם

b. ראש של רשימה מקושרת לכל המופעים (הצמתים) של העצם הנ"ל בעץ.

בהתבסס על ההגדרות הפורמליות שלעיל ניתן להציג את האלגוריתם לבניית עץ FP: [14]

Algorithm 1 (FP-tree construction)**.**

Input: A transaction database *DB* and a minimum support threshold ξ.

Output: FP-tree, the frequent-pattern tree of *DB*.

Method: The FP-tree is constructed as follows.

1. Scan the transaction database *DB* once. Collect *F*, the set of frequent items, and the support of each frequent item. Sort *F* in support-descending order as *FList*, the *list* of frequent items.

2. Create the root of an FP-tree, *T* , and label it as ''null''. For each transaction *Trans* in *DB* do the following:

Select the frequent items in *Trans* and sort them according to the order of *FList*. Let the sorted frequent-item list in *Trans* be [*p* | *P*], where *p* is the first element and *P* is the remaining list.

Call *insert tree*([*p* | *P*], *T*). The function *insert tree*([*p* | *P*], *T*) is performed as follows.

If *T* has a child *N* such that *N.item-name* = *p.item-name*, then increment *N*'s count by 1;

else create a new node *N*, with its count initialized to 1, its parent link linked to *T* , and its node-link linked to the nodes with the same *item-name* via the node-link structure.

If *P* is nonempty, call *insert tree*(*P, N*) recursively.

ניתוח סיבוכיות

ניתן לראות כי בנייתו של עץ FP, מצריכה בדיוק שתי סריקות של בסיס הנתונים[14]:

- סריקה למציאת התמיכה של כל אחד מהעצמים בכל הרשומות בטבלה.
- סריקה של בסיס הנתונים בכדי להכניס את כל העצמים אל תוך עץ ה FP.

נציין כי העלות של הכנסת רשומה מבסיס הנתונים אל תוך העץ היא O(freq(Trans)) כאשר Trans מייצג את הרשומה, והפונקציה *freq* מחזירה את רשימת האיברים התדירים ברשומה. כלומר עלות הכנסת רשומה הינה פונקציה של מספר האיברים התדירים ברשומה זו.

תכונות נוספות בעץ[14]

בהינתן בסיס נתונים DB, ותמיכה מינימלית ξ. ללא התחשבות בשורש העץ, ניתן לומר כי גודל עץ FP חסום ע"י: $\sum_{T\in DB}|freq(T)|$. והגובה שלו חסום ע"י $max_{T\in DB}\{|freq(T)|\}$, כאשר *freq(T)* מייצג את תת הקבוצה התדירה של רשומה בבסיס הנתונים.

הוכחה:

קל לראות מתהליך בנייתו של העץ, כי עבור כל רשומה בטבלה, קיים בעץ מסלול (המתחיל מהשורש) המייצג את תת הקבוצה התדירה המוכלת ברשומה הנ"ל. מכיוון שהעומק של העץ הוא

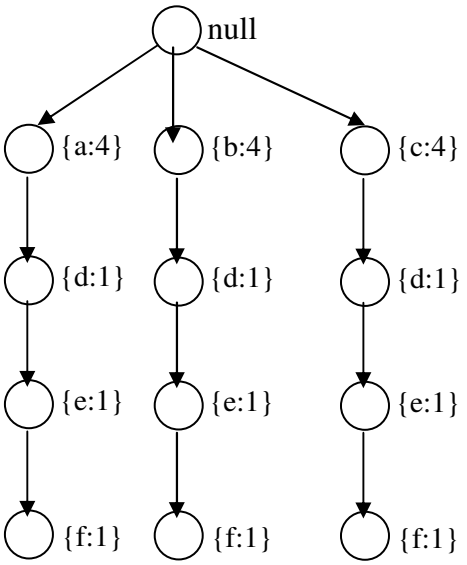
כעומקו של המסלול המקסימלי. ועומק המסלול המקסימלי ייצג בעצם את תת הקבוצה התדירה הגדולה ביותר. נוכל לומר כי עומק העץ חסום ע״י תת הקבוצה התדירה הגדולה ביותר.
גודלו של העץ (כלומר מספר הצמתים בו) חסום ע״י גודלו של בסיס הנתונים. וזאת מכיוון שכל רשומה בבסיס הנתונים תתרום ל**כל היותר** את עצמה בתור מסלול בעץ. אורכו של המסלול יהיה שווה **לכל היותר** לאורכה של הרשומה עצמה. וביתר דיוק, אורכו של המסלול יהיה שווה בדיוק לגודל תת הקבוצה התדירה המוכלת ברשומה הנ״ל. מכיוון שבמבנה הנתונים הנ״ל קיימים צמצומי מקום עקב שימוש חוזר באותם מסלולים, לרוב גודל העץ יהיה יותר קטן מחסם עליו זה. כפי שכבר צוין לעיל, מכיוון שהצמתים מסודרים בעץ בסדר יורד על פי התמיכה שלהם, הסיכוי לשיתופי צמתים גדול יותר. צמתים תדירים ימוקמו בראשית העץ וכך יגבר הסיכוי לשיתופי מסלולים בבניית מסלולים אחרים ובהקטנת נפחו של העץ. נסתייג ונאמר, כי אין זה מחייב כי תמיד במקרה זה ייבנה העץ בצורה אופטימאלית

לדוג׳ : בהינתן בסיס נתונים : $\{adef, bdef, cdef, a, a, a, b, b, b, c, c, c\}$. ונניח כי התמיכה המינימלית הינה 3. העצמים התדירים יהיו : {a:4, b:4, c:4, d:3, e:3, f :3}.

אם נבנה את עץ ה FP על סמך האלגוריתם הרגיל, סדר הקדימויות של העצמים יהיה :

a→b→c→d→e→f

כלומר על סמך סדר זה יוכנסו כל האיברים בכל הרשומות לעץ. במקרה הנ״ל העץ שיתקבל יהיה זה המוצג באיור 3.3.1 א׳ :



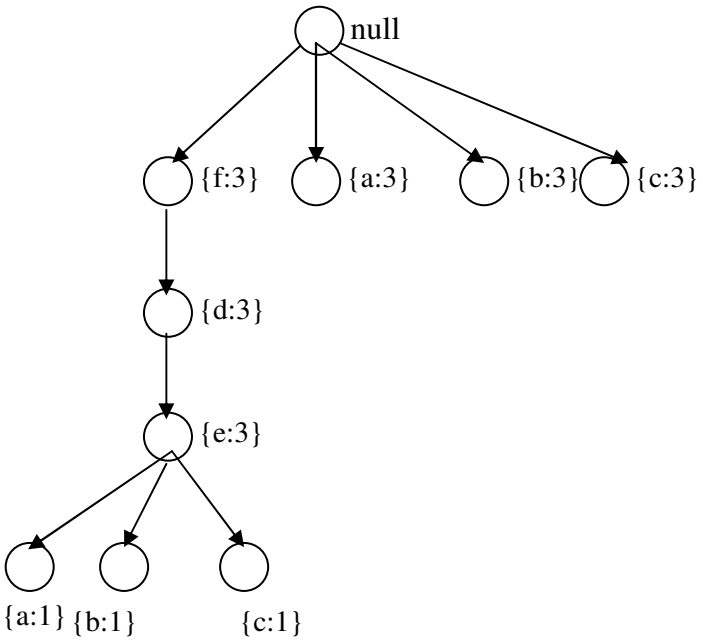
איור 3.3.1 א׳ – העץ הנוצר ממיון רשומות על פי תמיכה

ניתן לראות כי עץ זה מכיל 12 צמתים (לא כולל צומת השורש).

מאידך, אם נשתמש בסדר העדיפויות הבא:

$f \rightarrow d \rightarrow e \rightarrow a \rightarrow b \rightarrow c$

העץ שיתקבל יהיה זה המוצג באיור 3.3.1 ב׳:



איור 3.3.1 ב׳ – עץ ללא מיון רשומות לפי תמיכה

בעץ זה קיימים תשע צמתים. ראינו אם כן, שגודל העץ האופטימלי לא יתקבל תמיד במקרה של מיון הרשומות על סמך התמיכה שלהן.[12]

[12] בכדי למצוא את גודל העץ המינימלי ייתכן שיהיה עלינו למצוא את ה pattern שחוזר על עצמו המקסימלי בבסיס הנתונים. ולהשתמש בו בתור הקטע ההתחלתי של העץ ובכך לצמצם את מבנהו.

3.3.2 כריית מידע ממבנה הנתונים

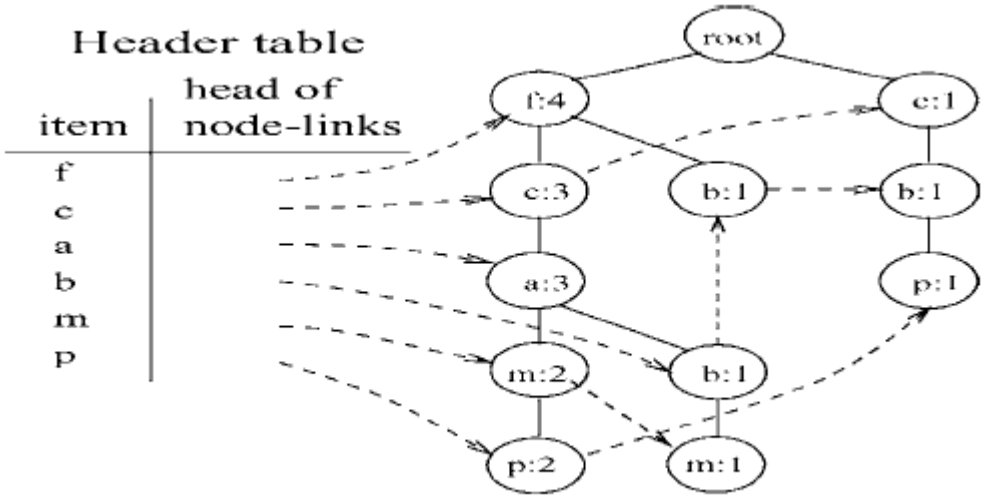
כעת, [14] לאחר שנתון לנו עץ FP הנתון בצורה קומפקטית, עלינו לוודא כי גם תהליך הכרייה יהיה שכזה. נזכיר כי ברצוננו למצוא דרך יעילה לייצר את כל תתי הקבוצות התדירות בסיבוכיות נמוכה ככל שניתן. בכדי להבהיר את תהליך הכרייה נשתמש בדוגמא הבאה[19]:

בהינתן בסיס הנתונים הבא:

TID	Items bought	(Ordered) frequent items
100	f, a, c, d, g, i, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m, n	f, c, a, m, p

איור 3.3.2 א' בסיס הנתונים[14]

עץ ה FP שיתקבל מבסיס הנתונים הנ"ל הוא:



איור 3.3.2 ב' עץ FP וטבלת הקישור לעצמים [14]

מכיוון שעבור כל המסלולים המכילים עצמים תדירים בעץ ה FP ומכילים עצם מסוים (לדוג' a_i).

ניתנים להגעה ע"י התחלה מהמקום של a_i בטבלה. (וזאת מכיוון שזו דרך בנייתה של הטבלה – קישור ישיר בין כלל המופעים של עצם מסוים בין כלל הרשומות). תהליך הכרייה יתחיל מתחתית הטבלה.

עבור צומת p – הצומת הנ״ל הינה חלק משני מסלולים קיימים בעץ (בעצם שתי רשומות בבסיס הנתונים):

{f:4,c:3,a:3,m:2,p:2} , {c:1,b:1,p:1}.

המסלול הראשון מציין כי המחרוזת fcamp מופיעה פעמיים בבסיס הנתונים (כמספר מופעי הזנב). נציין כי ניתן גם לראות מן המסלול כי המחרוזת fca מופיעה 3 פעמים בלבד.

המסלול השני מציין כי המחרוזת cbp מופיעה פעם אחת בלבד. שתי המחרוזות הללו יכונו *Conditional Pattern Base*. המסלולים ייוצגו בצורה הבאה (ללא הסיומת p).

{(*fcam*:2), (*cb*:1)}

כעת נבנה עץ FP על סמך הנתונים הללו, העץ הנ״ל ייקרא *Conditional FP – Tree*.

הענפים בעץ הנ״ל ייצגו תתי קבוצות תדירות (בשיתוף p). כך נמשיך עבור כלל העצמים[13] בבסיס הנתונים על פי הטבלה. המשך האלגוריתם יהיה הפעלה רקורסיבית מחדש על תתי עצים עד שמתקבל עצם יחיד המצורף לסופית הקיימת. בצורה זו נבנה את כל תתי הקבוצות התדירות.

כעת נציג כמה מאפיינים של עץ FP שנלמדו[14] מתהליך הכרייה:

1) עבור כל עצם תדיר a_i ברשומה נתונה, כל המסלולים האפשריים המכילים עצמים תדירים ואת a_i ניתנים להגעה ע״י מעקב אחר רשימת הצמתים של a_i.
2) עבור עץ FP בעל מסלול תדיר יחיד- תתי הקבוצות התדירות הינן כלל הצירופים האפשריים של העצמים במסלול הנ״ל.
3) בכדי לחשב את המסלולים התדירים עם סופית a_i, רק התחיליות של המסלולים המכילים את a_i יבדקו. המונה הממוקם בכל צומת יכיל את אותו מספר כמו הצומת a_i במסלול.
4) נניח כי α הינה רשומה בבסיס הנתונים. B הינה ה – *Conditional Pattern Base* של α. ו β מהווה רשומה ב B. התמיכה של α U β בבסיס הנתונים תהיה שווה לתמיכה של β ב B.
5) נניח[15] כי α הינה רשומה תדירה בבסיס הנתונים. B הינה ה – *Conditional Pattern Base* של α. ו β מהווה רשומה ב B. ניתן לומר כי α U β הוא תדיר ב DB אם״ם β הינה תדירה ב B.

בהסתמך על מאפיין - 5 , ניתן להסביר את נכונות הרקורסיה בתהליך הכרייה. כפי שהוסבר הכרייה יכולה להתבצע ע״י מציאת כל תתי הרשומות התדירות בגודל 1 בבסיס הנתונים. לאחר מכן עבור כל אחד מהעצמים התדירים נבנה *Conditional Pattern Base*. ניתן א״כ להתייחס לתהליך הכרייה כתהליך הכולל תחילה כרייה של תתי רשומות בגודל 1. ואז, בהדרגה נגדיל את

---

[13] נציין כי, בניתוח של m אין צורך להחשיב את p מכיוון שהוא נותח קודם לכן.

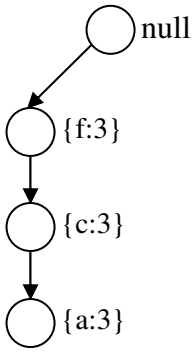
[14] ניתן לעיין בהוכחה מלאה של כל אחד מהמאפיינים ב [14]

[15] תכונה 5 נובעת ישירות מתכונה 4.

תתי הקבוצות התדירות בכך שנכרה את ה *Conditional Pattern Base* שלו. בצורה זו המרנו את בעיית כריית תתי קבוצות תדירות בגודל k, לרצף של k בעיות כרייה של תתי קבוצות תדירות בגודל 1.

דוגמא:

בכדי להבהיר תכונה זו לעומק, נציג את תהליך הכרייה עבור m מהדוגמא הקודמת (איור 3.3.2 ב').



איור 3.3.2 ג' ***Conditional FP- Tree of m*** [19]

נשים לב, כי בעץ המובא ב 3.3.2 ג' לא הצגנו את m מכיוון שהוא מהווה סופית קבועה. כעת, נכרה את העץ הנ"ל בצורה רקורסיבית כך: *mine(f:3,c:3,a:3|m)*. כלומר כעת, עלינו לבצע את תהליך הכרייה מחדש על העץ הנ"ל.

- נתחיל מ a: בעצם עלינו להתייחס ל a בהקשר של הסופית שלו m כלומר am.

ה *Conditional Pattern Base* של (am:3) יהיה (fc:3) (מה שנשאר מהשורש עד a). כעת נקרא שוב רקורסיבית לתהליך הכרייה עם (fc:3) כך: *mine(f:3,c:3|am)*.

- *mine(f:3,c:3|am)* – כעת הסופית am תודבק ל c ול f . כלומר, כעת יש לנו שני מסלולים: (cam:3) ו (fam:3).
  - עבור (cam:3) ה *Conditional Pattern Base* יהיה (f:3). לכן נקרא שוב בצורה רקורסיבית ל *mine(f:3|cam)*. ממנו נקבל את (fcam:3).
  - עבור (fam:3) – נקבל פשוט את (fam:3)
- נעבור ל c: שוב גם כאן לא נשכח את הסופית m: ה *Conditional Pattern Base* יהיה (f:3), וכעת גם לו נקרא בצורה רקורסיבית *mine(f:3|cm)*.
  - *mine(f:3|cm)* – נקבל מסלול יחיד (fcm:3)
- עבור f, העצם התדיר יהיה(fm:3) ,

אם נסכם את כל תתי הקבוצות התדירות שנבנו נקבל:

{(m:3), (am:3),(cm:3),(fm:3),(cam:3),(fam:3),(fcam:3),(fcm:3)}

וכעת אם נחזור לתכונה 5 נראה כי עבור (m:3) המהווה רשומה תדירה בבסיס הנתונים. אנו יודעים כי (a:3) הינה רשומה ב *Conditional Pattern Base* שהוא (fcam:3), נוכל לומר כי (am:3) יהיה תדיר בבסיס הנתונים עצמו.

ניתן לראות באיור 3.3.2 ד' טבלה מסכמת לכלל העצים בדוגמא.

Item	Conditional pattern-base	Conditional FP-tree
p	$\{(fcam{:}2), (cb{:}1)\}$	$\{(c{:}3)\}\|p$
m	$\{(fca{:}2), (fcab{:}1)\}$	$\{(f{:}3,\ c{:}3,\ a{:}3)\}\|m$
b	$\{(fca{:}1), (f{:}1), (c{:}1)\}$	$\emptyset$
a	$\{(fc{:}3)\}$	$\{(f{:}3, c{:}3)\}\|a$
c	$\{(f{:}3)\}$	$\{(f{:}3)\}\|c$
f	$\emptyset$	$\emptyset$

איור 3.3.2 ד' טבלה מסכמת [19]

3.3.3 הסבר ודוגמא מסכמת [13]:

בהינתן בסיס הנתונים הבא:

TID	List of items id's
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

בסיס הנתונים - איור 3.3.3 א'[13]

בתחילה נבצע סריקה התחלתית של בסיס הנתונים, הסריקה תהיה זהה לסריקה הראשונית של אלגוריתם אפריורי.כלומר, בסריקה זו ייספרו המופעים של כל איבר בבסיס הנתונים.
לאחר קבלת התוצאה, נמיין את הקבוצות על פי התמיכה בסדר יורד[16] ונקבל[17]:

$$L = \{\{I2:7\},\{I1:6\},\{I3:6\},\{I4:2\},\{I5:2\}\}$$

נניח שהתמיכה המינימלית הינה 2. במקרה שכזה שום עצם לא יסונן.

---

[16] כפי שציינו בהסבר על עץ ה FP הסיבה לשימוש בסדר יורד היא לוודא שהעצמים הפחות תדירים יהיו בעלים. ואילו העצמים היותר תדירים יהיו בצמתים וישותפו עם תנועות אחרות בבסיס הנתונים [16].

[17] המספר שמופיע לאחר האבר, מציין את מספר המופעים שלו בבסיס הנתונים

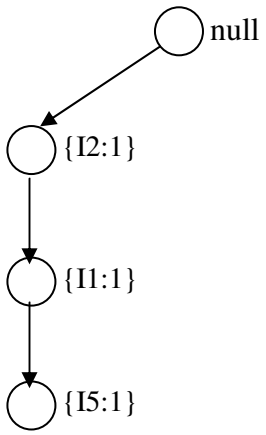
כעת נבנה את עץ ה FP. העץ ייבנה בצורה הבאה:

- שורש העץ יהיה ריק
- כעת נבצע סריקה נוספת של בסיס הנתונים, כעת יבוצע עיבוד של הנתונים בכל רשומה בבסיס הנתונים בסדר שבו הם מוינו בקבוצה L. עבור כל רשומה יווצר פיצול מהשורש בעץ. לדוגמא:

  עבור הרשומה הראשונה בבסיס הנתונים {I1,I2,I5}:

  - סדר העיבוד (על פי הסדר ב L) יהיה: {I2,I1,I5}.
  - I2 יהיה הבן של שורש העץ, I1 מחובר אליו, ו I5 מחובר ל I1.

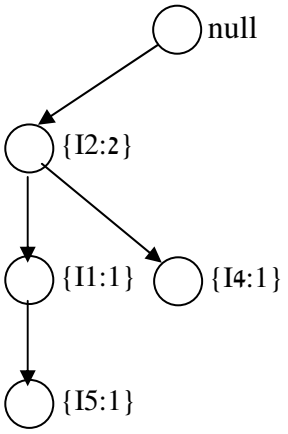
כעת העץ ייראה כך: יש לשים לב לצורה שבה מוצגים האיברים בעץ. ליד כל איבר מוצג מספר הפעמים שנעשה בו שימוש בעץ.



עץ FP איור 3.3.3 ב'

עבור הרשומה השנייה בבסיס הנתונים {I2,I4} :

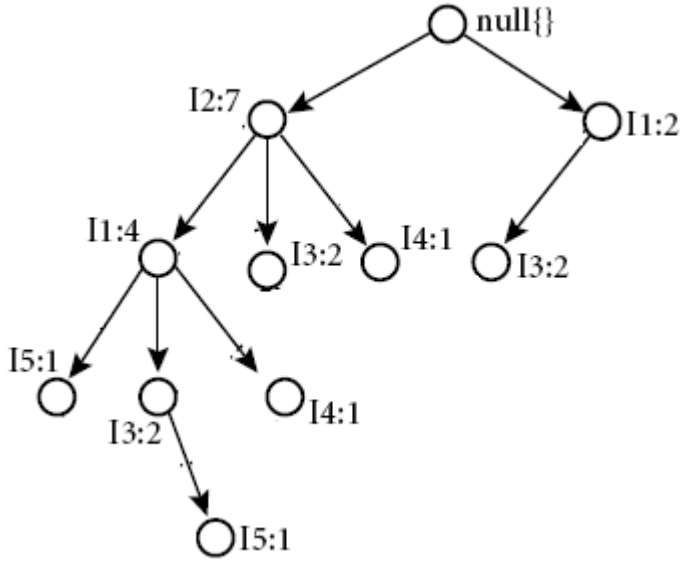
- סדר הנתונים (עפ"י L) יהיה : {I2,I4}.
- לכן תיווצר התפצלות מהשורש, I2 יהיה בן ישיר של השורש ו I4 יהיה מחובר אליו. מכיוון ש I2 כבר הינו בן של השורש לא ייווצר בן חדש לשורש. רק יתווסף בן נוסף ל I2. בכדי לציין את השימוש *העסף* שנעשה ב I2. נגדיל את המספר ליד I2 ונקבל :



עץ FP איור 3.3.3 ג'

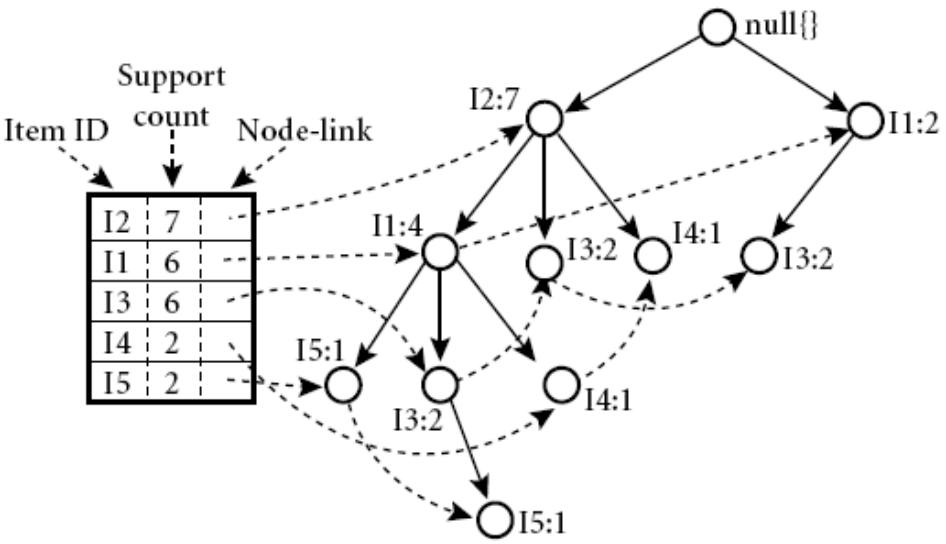
כך במהלך התהליך בבואנו להוסיף צומת לעץ. אם הצומת כבר קיימת כל שעלינו לעשות הוא להגדיל ב 1 את המונה הממוקם סמוך לצומת הנ"ל.

העץ הסופי ייראה כך :



עץ FP -איור 3.3.3 ד'[13]

בכדי להקל על הסריקה של העץ שנוצר ניצור טבלה שתכיל מצביע עבור כל עצם בבסיס הנתונים אל מיקומו הראשון בעץ. העצם הנ"ל יכיל גם מצביע לאיברים הבאים שהינם כמותו. כך לדוג' עבור העצם I1, יווצר מצביע ל {I1:4} וממנו יהיה מצביע ל {I1:2}. כך ניתן לראות כי עבור כל עצם תיווצר "מעל העץ" כמין שכבה של רשימה מקושרת של מיקומי העצם בעץ.



עץ FP וטבלת ההכוונה -איור 3.3.3 ה'[13]

לאחר ההמרה של בסיס הנתונים לצורת העץ נוכל לעבור לשלב השני של האלגוריתם והוא כריית המידע מעץ ה FP.

בנקודה זו של האלגוריתם ניתן לומר כי ביצענו סוג של רדוקציה מבעיית כריית קבוצות תדירות בבסיסי נתונים לבעיית כריית קבוצות תדירות בעץ FP.

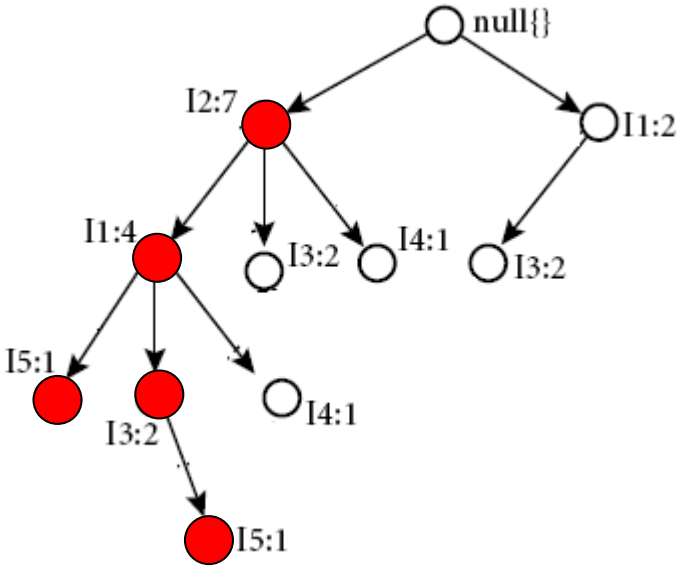
הרעיון המרכזי של המשך האלגוריתם הינו שימוש בשיטת "הפרד ומשול" על העץ בכדי למצוא את תתי הקבוצות התדירות. לאחר שניצור תת עץ נחזור ונכרה אותו בצורה רקורסיבית.

כך בעצם, נוריד את הצורך בחילול כולל של כל תתי הקבוצות האפשריות.

<u>המשך הדוגמא:</u>

כעת, נתחיל לעבור על העץ מהעצם בעל התמיכה הנמוכה ביותר - I5.

I5 ניתן להגעה ע"י שני מסלולים בעץ (צבועים באדום).



עץ FP -איור 3.3.3 ו'[13]

כפי שניתן לראות מדובר על : {I2:7, I1:4, I3:2, I5:1}, {I2:7, I1:4, I5:1}.

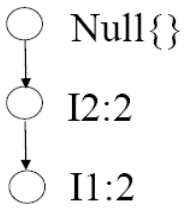
מכיוון שמדובר רק חלק זה של העץ יש לעדכן את המונה ליד כלל הצמתים בשני המסלולים, למספר הפעמים האמיתי שעברו בהם עד להגעה ל I5. לכן המסלולים האמיתיים ייראו כך :

{I2:1, I1:1, I3:1, I5:1}, {I2:1, I1:1, I5:1}

מכיוון ש I5 ישמש כזנב של המסלול ניתן לומר כי שני המסלולים התחיליים המובילים אליו הם :

{I2:1, I1:1, I3:1}, {I2:1, I1:1}

שני מסלולים אלו מכונים – ***Conditional Pattern Base***. מתוך המסלולים הללו נבנה את תת העץ שעליו תתבצע כריה. העץ שייבנה יהיה עץ FP ממש כמו עץ האב ויבנה באותה צורה. העץ שיתקבל כאן יהיה :

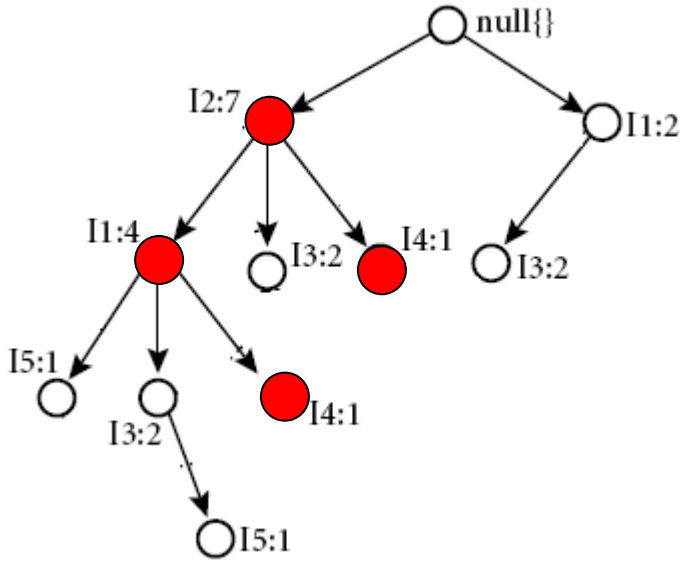


עץ FP -איור 3.3.3 ז' [1]

נציין כי I3, לא מופיע בעץ וזאת מכיוון שהתמיכה שלו קטנה מהתמיכה המינימאלית שהוגדרה (2). ניתן לראות כי הקבוצה שהתקבלה היא : {I2:2,I1:2}. נצרף אליהם את I5 בכל הקומבינציות האפשריות ונקבל :

{I2:2, I5:2}, {I1:2 ,I5:2}, { I2:2, I1:2 ,I5:2}

כעת, נמשיך בתהליך הכרייה עבור I4. המסלולים המובילים אליו(מסומנים באדום) הם :

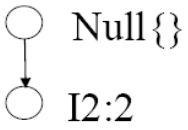


עץ FP -איור 3.3.3 ח'[18]

המסלולים עבור I4, הם {I2:7,I1:4,I4:1} ו {I2:7,I4:1}. מכיוון ש I4 הינו סוף קבוע של המסלול נוריד אותו מתיאור המסלול, כמו כן מכיוון שאנו מתייחסים רק למונה שקשור ל I4 יתקבל :

{I2:1}, {I2:1,I1:1}

אם נבנה את תת העץ FP מהעצמים הנ"ל נקבל :



עץ FP -איור 3.3.3 ט' [1]

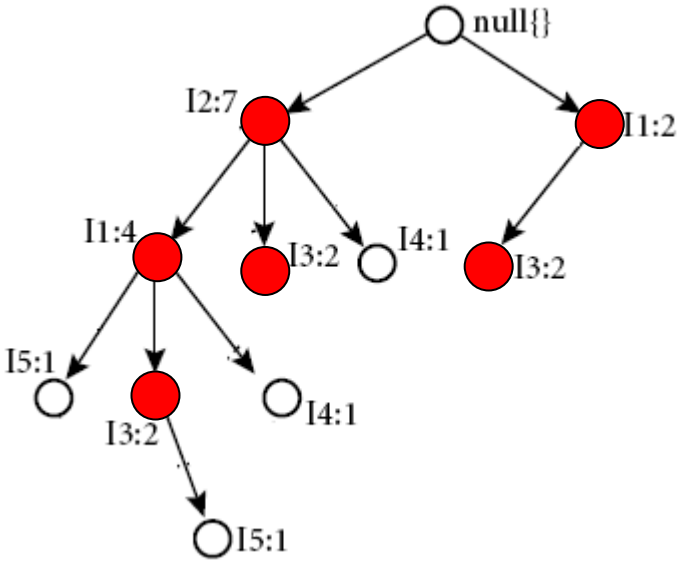
יש לשים לב כי I1 לא נכלל בעץ ה FP וזאת מפני שבתהליך הבניה של העץ מסננים עצמים שאינם עומדים בדרישות התמיכה המינימאלית (במקרה הזה 2).

אם נצרף ל {I2:2} את I4, נקבל :

{I2:2, I4:2}

נציין כי למרות ש I5 מופיע אחרי I4 בעץ הוא לא נכלל בתהליך החישוב וזאת מכיוון שכבר בוצע ניתוח של I5 מקודם.

כעת נמשיך את הניתוח עבור I3 :



עץ FP -איור 3.3.3 י'[18]

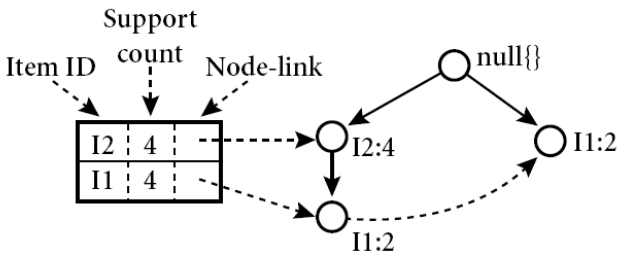
עבור I3 קיימים 3 מסלולים :

{I2:7,I1:4,I3:2}, {I2:7,I3:2}, {I1:2, I3:2}

שוב, גם כאן ניתן להוריד את I3 ולעדכן את המונה בכל אחד מהעצמים במסלול[18].

{I2:2, I1:2},{I2:2},{I1:2}.

תת עץ ה FP שייבנה יהיה :



עץ FP -איור 3.3.3 יא'[18]

[18] ניתן לשים לב, כי בעצם במונה יהיה תואם למונה של העצם שעליו אנו מבצעים את הכריה (במקרה זה I3) . וזה מפני שאנו מונים את מס' הפעמים שהגענו לעצם **זה**.

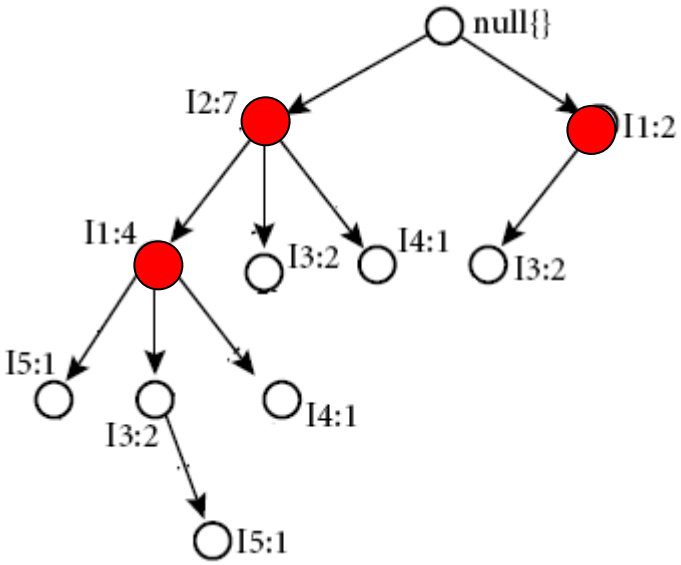
כלומר תתי הקבוצות שהתקבלו:

{I2:4, I1:2}, {I1:2}

אם נצרף את I3 לכל אחת מהקבוצות נקבל:

{I2:4, I3:4}, {I1:4, I3:4}, {I2:2, I1:2, I3:2}

בצורה דומה נבצע את אותו תהליך ל I1.



עץ FP -איור 3.3.3 יב'[18]

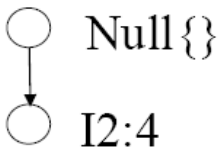
המסלולים הם:

{I2:7, I1:4}, {I1:2}

אם נתעלם מ I1, ונעדכן את המונים:

{I2:4}

העץ שיתקבל:



עץ FP -איור 3.3.3 יג' [1]

כלומר הקבוצה התדירה שתתקבל היא:

{I2:4, I1:4}

באיור 3.3.3 יד' מוצגת טבלה מסכמת לכלל הקבוצות התדירות שהתקבלו בשילוב הדרך לקבלתן.

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

טבלה מסכמת -איור 3.3.3 יד'[13]

בעצם נוכל לומר כי הקבוצות התדירות שהתקבלו הן:

{I2,I5},{I1,I5},{I2,I1,I5},{I2,I4}, {I2,I3}, {I1,I3},{I2,I1,I3}, {I2,I1}

כפי שצוין לעיל בשיטה הנ"ל, במקום לחפש תתי קבוצות תדירות גדולות, נחפש תתי קבוצות קטנות בצורה רקורסיבית. יש לציין כי כאשר מדובר על בסיס נתונים גדול, לעיתים אין זה מציאותי לבנות עץ FP עקב מגבלות זיכרון במחשב. במצב כזה ניתן לחלק את העץ לכמה תתי עצים שונים ולבצע את פעולת הכרייה עבור כ"א בנפרד. ניתן לחזור על תהליך זה בצורה רקורסיבית במקרה הצורך. כמובן שלאחר מכן נצטרך לבצע תהליך של איחוד המידע שנובע מכלל תתי העצים שנכרו בתהליך. אך למרות זאת עצם העובדה שהתהליך הינו רקורסיבי מאפשרת לנו לחלק את העצים לתתי עצים נפרדים (במידת הצורך) ולכרות כ"א מהם בנפרד. תהליך איחוד המידע יכלול חוץ מחיבור בפועל של העץ למקומו המקורי בעץ הראשי, שלב של עדכוני אינדקסים בעץ הראשי שיכילו את שינוי מספר הפעמים שבוצע ביקור בצומת זה. תהליך האיחוד יתבצע מס' קבוע של פעמים (כתלות במס' החלוקות). כמו כן רצף השינויים יבוצע לאורכו של העץ (כלומר, מנקודת החיבור כלפי מעלה). נוכל לומר כי מבחינת סיבוכיות במקרה של חלוקת העץ עקב בעיות זיכרון עלות החיבור מחדש תהיה o(log(n)). למרות כל זאת מחקרים מראים [1] כי לעומת אלגוריתם אפריורי, ל FP Growth יש יתרון של כמעט סדר גודל שלם.

סיבוכיות הזמן והמקום של האלגוריתם הוצגה לעיל בסעיף 3.3.1לכן לא תפורט מחדש.

לסיכום ניתן לציין באלגוריתם FP – Growth מספר יתרונות ברורים: [16]

- דחיסת בסיס הנתונים למבנה של עץ – מקטינה את סיבוכיות הזיכרון הנדרשת (שימוש חוזר במסלולים בעץ).
- צורת היצירה של תתי הקבוצות הינה בשימוש העץ (חיבור צומת נוספת לענף) ולכן היא אינה כוללת מכפלות של קבוצות וייצור של תתי קבוצות לבדיקה – פעולה שעלותה יקרה מאוד.
- נעשה בשימוש בשיטת הפרד ומשול בכדי להקטין סיבוכיות זמן ומקום.
- מצריך רק 2 מעברים על בסיס הנתונים המקורי. כפועל יוצא מכך סיבוכיות האלגוריתם עצמו מבחינת מעבר על העץ היא פונקציה של גודלו של בסיס הנתונים בלבד.

מאידך, ניתן לשים לב למספר חסרונות בולטים באלגוריתם[25] :

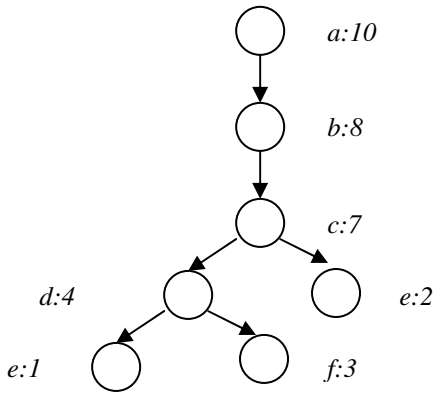
- יתכן מצב של בעיתיות בסיבוכיות מקום עקב גודלו של העץ. מצב זה ייתכן במקרים של בסיסי נתונים גדולים.
- תהליך בניית העץ הינו יקר מבחינה חישובית ומצריך 2 מעברים **מלאים** על כל בסיס הנתונים.[19]
- התמיכה יכולה להיות מחושבת אך ורק לאחר שכל תת הקבוצה הוכנסה בצורה מלאה לעץ.

[19] קיים Trade off בין הזמן שלוקח לבנות את העץ מצד אחד לבין העובדה שלאחר שבנינו אותו כל המידע הנדרש נמצא בהישג יד ללא מעבר נוסף על בסיס הנתונים.

3.3.4 מסלול תחיליות יחיד (Single Prefix Path)

למרות שהשיטה הכללית של עץ FP עובדת עבור כל סוג של עץ. ניתן עדיין למצוא אופטימיזציות לאלגוריתם עבור מבני נתונים מסוימים ביניהם: *Single prefix- path FP – tree*. זהו עץ FP לכל דבר, עץ זה מכיל רק מסלול אחד או מסלול תחילי יחיד מהשורש לצומת הראשונה בעץ המכילה יותר מבן אחד[14].

<u>דוגמא:</u>



איור 3.3.4 א' - Single Prefix Path [14]

במקרה שלנו, המסלול יהיה a→b→c, עבור עצים מהסוג הזה קיימת שיטת כרייה יעילה יותר מהשיטה המקובלת ב FP. עדיף יהיה *לחלק* את העץ לשני מקטעים:

- Single Prefix – Path: (a:10)→(b:8)→(c:7) יסומן ב P
- Multipart path – מה שנשאר מהעץ (d,e,f) (מוסיפים שורש מדומה - R)

ניתן לכרות את שני העצים בנפרד ולחבר את התוצאות.

את תוצאות הכרייה עבור single prefix – path ניתן לחשב בקלות, כפי שהוסבר לעיל במקרה כזה משתמשים בכל הקונפורמציות של המסלול העומדות בתנאי התמיכה המינימלית.

(a:10),(b:8),(c:7),(ab:8),(ac:7),(bc:7)(abc:7)

נניח ו Q הינו המקטע השני שבעץ (בעל השורש המדומה) תוצאת הכרייה על Q תהיה:

{(d:4),(e:3),(f:3),(df:3)}

כעת נוכל לומר כי עבור כל תת קבוצה תדירה ב Q, נוכל להתייחס ל R כ Conditional Frequent pattern – base. כמו כן, כל תת קבוצה ב Q בצירוף כל תת קבוצה מ R יכולה להיחשב מסלול תדיר.

דוגמא:

עבור (d:4) שהינו חלק מ Q, נוכל להתייחס ל P כ Conditional Frequent pattern – base של Q, ולכן כל מסלול שמחולל מ P כגון : (a:10) ייצור תת קבוצה תדירה חדשה לדוגמא : (ad:4). לכן עבור (d:4) הקבוצה של תתי הקבוצות התדירות תהיה[20] :

$$(d:4)Xfreq\_pattern\_set(P) = \{(ad:4),(bd:4),(cd:4),(abd:4),(acd:4),(bcd:4),(abcd:4)\}$$

נציין כי התמיכה המינימלית של הקבוצה המשותפת תהיה הנמוכה מבין שתי התמיכות של שתי הקבוצות. נוכל לומר כי הקבוצה השלמה הנובעת מחיבורן של שתי קבוצות המכילות תתי קבוצות תדירות היא :

Freq_pattern_set(Q) X freq_pattern_set(P)

כאשר התמיכה המשותפת שווה לתמיכה המינימלית של Q, שאינה גדולה יותר משל P.

הוכחת נכונות חלקית ותכונות עיקריות

בחלק זה [14] נעסוק, בהוכחת נכונות האלגוריתם שתואר לעיל, ונציג תכונות עיקריות של מבנה הנתונים הנ״ל. בכדי להוכיח את נכונות נשתמש במספר הנחות :

1. בהינתן עץ FP – T , המכיל מסלול יחיד – P. הקבוצה המלאה של כל תתי הקבוצות התדירות של T, ניתנות ליצירה ע״י שימוש בכלל הפרמוטציות האפשריות של תתי המסלולים ב P, בתנאי שהם עומדים בתנאי התמיכה המינימלית.

הוכחה:

נניח[14] כי נתון המסלול P :

$$(a_1{:}s_1 \rightarrow a_2{:}s_2 \rightarrow \ldots \rightarrow a_k{:}s_k)$$

מכיוון שעץ ה FP הנ״ל מכיל מסלול יחיד. נוכל לומר כי התמיכה s_i של כל עצם a_i (עבור $1 \le i \le k$). הינה התדירות של הופעתו עם המחרוזת התחילית שלו. לדוג׳ נוכל לומר כי הרצף a_1a_2 מופיע s_2 פעמים. לכן עבור a_i נוכל לומר כי עבור כל תת קבוצה של עצמים במסלול (בעצם תת מסלול) לדוג׳ $a_j \ldots a_n$ כאשר $j \ge 1, n \le k$, מתקיים כי תת הקבוצה הנ״ל גם תדירה. מכיוון שכל עצם ב P הינו יחודי לא יווצר מסלול נוסף שכזה. כמו כן אף תת קבוצה תדירה לא תוכל להיווצר מאיברים שמחוץ לעץ הנ״ל (מכיוון שאין כאלו).

[20] סימון ה X כאן נועד לציין שכל איבר מקבוצה A מצורף עם כל איבר מקבוצה B (עבור A X B). התמיכה הנבחרת היא התמיכה המינימלית מבין שתי הקבוצות המצורפות.

2. נניח ונתון עץ T – FP. ונניח כי עץ זה מורכב משני מסלולים P- Single Prefix Path ו Q- Multipath. בדומה לאיור 3.3.4 א'. כלל תתי הקבוצות התדירות יתקבלו באמצעות שלושת החלקים הבאים:

- כלל המסלולים התדירים שנוצרו מ P (כמובן, בתנאי שכלל העצמים במסלול הנ"ל הם בעלי תמיכה מינימלית לפחות).
- כלל המסלולים התדירים שנוצרו מ Q.
- כלל המסלולים התדירים שנוצרו ע"י מכפלה של המסלולים התדירים ב P ו Q. כאשר התמיכה המשותפת תהיה זו המינימלית מבין התמיכות של שני המסלולים.

<u>הוכחה:</u>

בהסתמך[14] על תהליך בניית העץ, כל צומת a_i ב Single Prefix Path של עץ FP מופיעה פעם אחת בלבד בעץ. מכיוון שלא קיימים צמתים משותפים בין P ו Q נוכל לכרות אותם בנפרד. כעת, כפי שהוכחנו מקודם (ב סעיף 1) כל מסלול שנוצר מ P הינו תדיר ויחודי. כמו כן כל תתי המסלולים שנוצרו מ Q הינם גם תדירים, וזאת מפני שמסלולים אלו קיימים כפי שהם ב Conditional Database של העץ המקורי ללא ערבוב של עצמים זרים. עבור P X Q, נוכל גם לומר כי תתקבל קבוצה ייחודית ותדירה וזאת מכיוון שכל תת קבוצה תדירה שנוצרה מ P יכולה להיחשב כתת קבוצה תדירה ב Conditional Pattern Base של העצמים התדירים ב Q. כמו כן התמיכה המינימלית תהיה המינימילית מבין התמיכות של P ו Q. וזאת מכיוון שאנו מעוניינים במספר הפעמים ששניהם מופיעים **ביחד.** מכיוון שהוכחנו מקודם כי כל מסלול שנוצר מ T יווצר או מ P או מ Q או מהאיחוד. נוכל לומר בוודאות כי בשיטה זו נקבל את כלל תתי הקבוצות התדירות שניתן להוציא מ T.

בהסתמך על התכונות שהוכחו לעיל נוכל להציג את האלגוריתם הבא לכריית FP- Tree[14]:

Input: Database DB representing FP Tree and minimum support threshold ξ

Output: The Complete set of Frequent Patterns.

First Call: FP-growth(Tree, null)

Procedure FP-growth(Tree, α)

{

(1) if Tree contains a single prefix path // Mining single prefix-path FP-tree

(2) then {

(3) let P be the single prefix-path part of Tree;

(4) let Q be the multipath part with the top branching node replaced by a null root;

(5) for each combination (denoted as β) of the nodes in the path P do

(6) generate pattern β U α with *support = minimum support of nodes in β*;

(7) let freq pattern set(P) be the set of patterns so generated;

}

(8) else let Q be Tree;

(9) for each item a_i in Q do { // Mining multipath FP-tree

(10) generate pattern $\beta = a_i$ U α with *support = a_i.support;*

(11) construct β's conditional pattern-base and then β's conditional FP-tree $Tree_\beta$;

(12) if $Tree_\beta = \varnothing$

(13) then call FP-growth($Tree_\beta$, β);

(14) let freq_pattern_set(Q) be the set of patterns so generated; }

(15) return(freq_pattern_set(P) U freq_pattern_set(Q) U (freq pattern set(P)

X freq pattern set(Q)))

}

ניתוח סיבוכיות:

האלגוריתם סוקר את בסיס הנתונים פעם אחת ומייצר בסיס נתונים $B_{a(i)}$, עבור כל עצם תדיר – a_i. מבנה נתונים זה יכיל עבור כל עצם את התחיליות שלו בעץ. כעת בצורה רקורסיבית נבצע כרייה עבור מבנה הנתונים הנ"ל. כפי שהסברנו לעיל עץ FP המייצג בסיס נתונים לרוב, קטן בהרבה מבחינת סיבוכיות מקום מבסיס הנתונים עצמו. כמו כן $B_{a(i)}$ קטן בהכרח מהעץ עצמו וזאת מכיוון שהוא מכיל חלק קטן מהמידע הקיים בעץ. לכן תהליך הכריה מראש עובד על כמות מידע קטנה יותר מאשר כל העץ. כמו כן נניח וקיים בסיס נתונים בעל 100 עצמים, סיבוכיות המקום הנדרשת לאלגוריתם תהיה רק 100 מכיוון שנאלץ לאחסן בזיכרון רק את הרשימה של 100 העצמים. למרות זאת האלגוריתם יצור בסביבות 10^{30} מסלולים תדירים אך העץ עדיין יכיל אך ורק 100 עצמים ולא נצטרך אפילו לבנות Conditional FP Tree בכדי למצוא את כלל המסלולים. סה"כ סיבוכיות מקום נדרשת תהיה $O(n)$ כאשר n הינו גודלו של בסיס הנתונים. מבחינת סיבוכיות זמן – אין שינוי בהיקפים של סדרי גודל. השיפור שמציג האלגוריתם הינו אך ורק בהיבטי סיבוכיות המקום.

3.3.5 הטלה של בסיסי נתונים

למרות השיפור הרב [14] שמציג אלגוריתם FP בסיבוכיות הזיכרון, במקרים בהם בסיס הנתונים יהיה ממש גדול או במקרים של תמיכה מינימלית נמוכה במיוחד יהיה זה בלתי מציאותי לצפות כי מבנה נתונים מבוסס עץ FP יוכל להכיל את המידע הנדרש במגבלות הזיכרון הקיימות.

בחלק זה נציג שיטה להטלה (projection) של בסיס הנתונים בכדי להקטין את גודלו ולאחר מכן נבנה עבור בסיס הנתונים הנ"ל (המוקטן) עץ FP. נשתמש בבסיס הנתונים ובעץ מהדוגמא הקודמת (איור 3.3.2 א' וב'). נניח ועץ זה לא יכול להיות מאוחסן מטעמי חיסרון במקום בזיכרון המחשב.נפעיל על בסיס הנתונים את ההטלה בצורה הבאה:

נתחיל מסוף רשימת העצמים התדירים – מ p (נזכיר כי הרשימה היא: f,c,a,b,m,p).

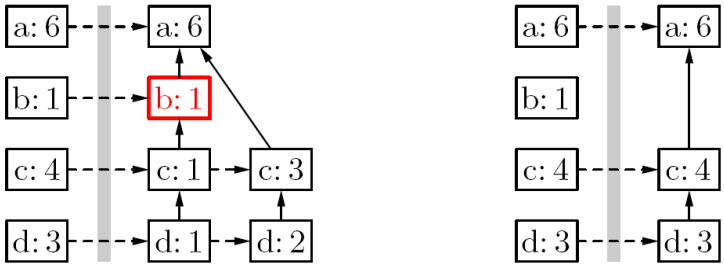
אוסף התנועות בבסיס הנתונים המכילות את p. יקובצו לבסיס נתונים אחד בשם p – projected DB. כל העצמים שאינם תדירים ו כן p עצמו הינם מיותרים שכן אינם תורמים לתהליך הכריה, לכן יורדו מבסיס הנתונים החדש (p יכול להיות מוסר מפני שברור ש p נמצא בבסיס הנתונים של p). ביצוע פעולה דומה על כלל האיברים בבסיס הנתונים ייתן את התוצאה הבאה:

Item	Projected database	Conditional FP-tree
p	$\{fcam, cb, fcam\}$	$\{(c{:}3)\}\|p$
m	$\{fca, fcab, fca\}$	$\{(f{:}3,\ c{:}3,\ a{:}3)\}\|m$
b	$\{fca,\ f,\ c\}$	$\emptyset$
a	$\{fc, fc, fc\}$	$\{(f{:}3,\ c{:}3)\}\|a$
c	$\{f,\ f,\ f\}$	$\{(f{:}3)\}\|c$
f	$\emptyset$	$\emptyset$

טבלה 3.3.5 א' [14]

ניתן להבחין כי בסיס הנתונים המוטל הינו בעצם בדיוק ה Conditional Pattern Base של העצם הנ"ל ממש כמו בטבלה 3.3.2 ד'. בצורה דומה נבצע את אותו תהליך על m. נבצע הטלה לבסיס נתונים יעודי עבור m. נשים לב כי מלבד m עצמו ועצמים בלתי תדירים סיננו מבסיס הנתונים של m גם את p. מכיוון שכלל קשריו עם m כבר תוארו קודם לכן בבסיס הנתונים של p. כך בכל פעם נצטרך לייצג פחות ופחות עצמים מבסיס נתונים. לעצים אלו נבנה את ה Conditional FP Tree. ונמשיך את תהליך הכריה כרגיל. במידה וגם העץ המצומצם הינו גדול מדי נחזור על תהליך ההטלה גם עבור ה Conditional FP Tree בצורה רקורסיבית. קיימות שתי שיטות[16] שונות לביצוע הטלה של בסיס נתונים: Partition Projection , Parallel Projection [14].

במאמר [1] המתאר מימוש של אלגוריתם FP – Growth[21]. מתואר שלב נוסף בשרשרת השלבים של האלגוריתם. מדובר על שלב של קיצוץ. מיקומו של שלב זה הינו לאחר ביצוע ההטלה. לאחר ההטלה, ניתן לקצץ מהעץ המוטל צמתים שאינם תדירים ולאחד צמתים בנים דומים. קיצוץ שכזה מקטין עוד יותר את נפחו של העץ ומקל על מלאכת הכרייה.
דוגמא לקיצוץ שכזה ניתן לראות באיור 3.3.5 ב׳.



איור 3.3.5 ב׳ [1]

ניתן לראות באיור את שני שלבי קיצוץ העץ:

- בשלב הראשון b מוסר כיוון שאינו תדיר. לכן לא נשתמש בו לביצוע כריה בבסיס הנתונים. לכן נסיר את הצומת המייצג את b מתוך העץ המוטל.
- לאחר מכן נאחד את שתי המסלולים שהיוו צאצאים של b. באיחוד זה יחוברו שני מופעים של צמתים זהים לצומת אחת בלבד.

ניתן לומר כי הקריטריון לקיצוצו של צומת הינו מידת נחיצותו להמשך תהליך הכריה. צומת שאינו תדיר לא יהיה חלק מתת קבוצה תדירה. לכן ניתן יהיה לקצצו. הקיצוץ מבוצע ע״י הסרה של הצומת מהעץ וחיבור בניו לאביו של הצומת שהוסר. במילים אחרות: רמת התמיכה של הצומת הינה הקריטריון הכמותי שיקבע האם הצומת יוסר מהעץ או לא.

[21] נציין כי במאמר המקורי מדובר על α – pruning, כאן התייחסנו לפעולה כאל קיצוץ רגיל.

3.3.6 הטלת עץ (Tree Projection)

Tree Projection הינה שיטה נוספת[22] לכריית תתי קבוצות תדירות [1]
בשיטה זו נבנה עץ ממומין בצורה לקסיקוגרפית ונטיל את בסיס הנתונים הגדול אל תוך סדרה של תת בסיסי נתונים בגודל קטן יותר. השיטה דומה בחלקה ל FP Growth אך לשתי השיטות יש הבדלים משמעותיים בשיטה שבה בונים את העץ ובשיטה שבה מבוצעת הכרייה בפועל.

<u>דוגמא:</u>

עבור בסיס הנתונים הבא:

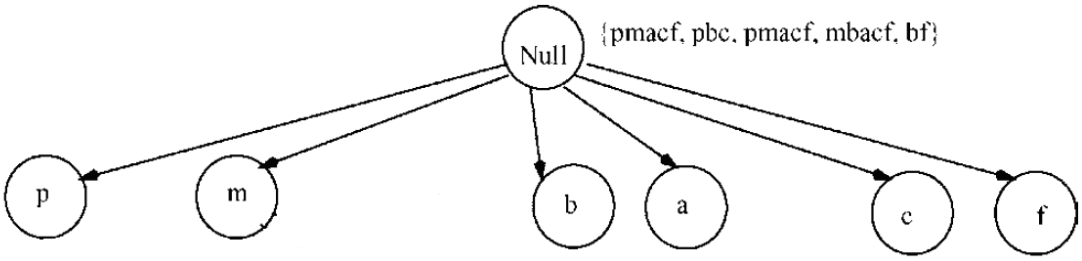
TID	Items bought	(Ordered) frequent items
100	f, a, c, d, g, i, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m, n	f, c, a, m, p

איור 3.3.6 א' [14]

נבנה עץ לקסיקוגרפי בשיטה שתוארה ב [1][23] ראשית עלינו לסרוק את בסיס הנתונים ולזהות את כל תתי הקבוצות התדירות בגודל 1. הסדר בו יסודרו הקבוצות יהיה על פי פרמטר התדירות בסדר עולה. לכן סדר העיבוד של הנתונים יהיה[24] p – m – b – a – c – f. כעת ניצור צומת בן של שורש העץ (כאן הינו null), עבור כל אחד מהעצמים כאן. נציין כי כל צומת יכיל שני נתונים:

- המידע הפנימי של הצומת (האות אותה הוא מייצג במסלול).
- מידע לגבי העצמים שעלולים לייצר מסלולים חוקיים ארוכים ע"י צירוף לצומת הנ"ל.(כמובן עם חשיבות לסדר שהוגדר לעיל).

נקבל את השורש והעצמים בעלי תתי קבוצות תדירות (בדוגמא זו התמיכה המינימלית הינה 3) \ לפחות בגודל 1:



איור 3.3.6 ב' [14]

---

[22] למרות שמדובר בשיטה נפרדת שלכאורה ראויה לתת פרק משלה, החלטנו להביאה תחת FP – Growth מכיוון שאלגוריתם זה דומה מאוד ל FP- Growth.

[23] לא נדון במסגרת זו בפרטיו המלאים של האלגוריתם. הרחבות ניתן לראות ב [1]

[24] נציין כי סדר זה הינו **הפוך** לסדר של עץ ה FP

ניתן לראות כי העץ שהתקבל אינו מכיל שום מידע לגבי תתי קבוצות בגודל 1 שאינן תדירות. כך מתקבל עץ המכיל תתי קבוצות תדירות בלבד.

כעת, בכדי למצוא את תתי הקבוצות התדירות בגודל 2 נבצע את הפעולות הבאות:

נבנה טבלה (מטריצה) בכדי לסכם את כלל התדירויות של כלל תתי הקבוצות בגודל 2. הטבלה שתתקבל תהיה:

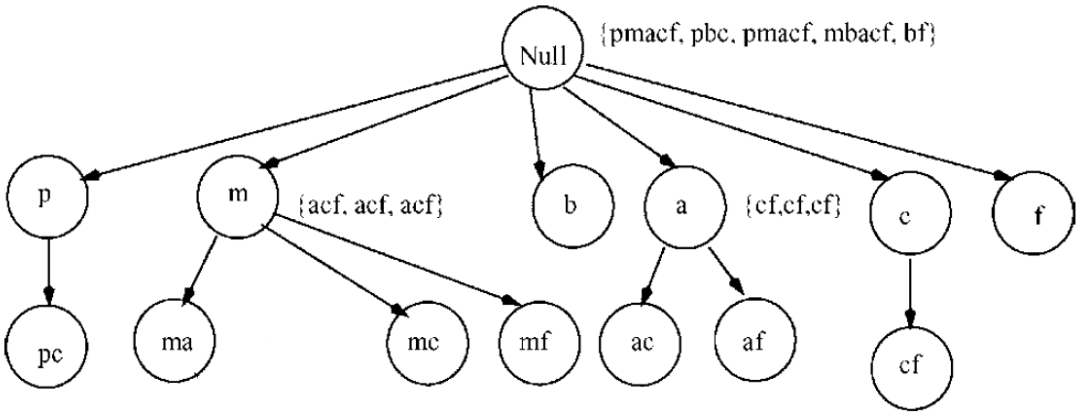
	p	*m*	*b*	*a*	*c*	*f*
p						
m	2					
b	1	1				
a	2	3	1			
c	3	3	2	3		
f	2	3	2	3	3	

איור 3.3.6 ג'[14]

מהמטריצה ניתן לראות שתתי הקבוצות התדירות בגודל 2 שנמצאו הן:

{pc, ma, mc, mf, ac, af, cf}, לכלל הקבוצות יש תמיכה מינימלית של 3.

הצמתים הנ"ל יוכנסו לעץ במקום המתאים. העץ שיתקבל יהיה:



איור 3.3.6 ד'[14]

נשים לב כי כעת מופיע ליד m ו a מידע נוסף, תתי הקבוצות שבהם נעשה שימוש בכדי ליצור את הבנים של m. בעזרת המידע הנ"ל ניתן יהיה בהמשך ליצור תתי קבוצות גדולות יותר. בכדי להבהיר את הנקודה נעמוד על ההבדל בין p ל m.

עבור p: נסתכל על בנו של p - pc, לא קיים בבסיס הנתונים עצמו שום רצף שכולל את p ואחריו את c ואחריהם עוד עצם (כמובן שהסדר מוגדר על פי התדירות). לכן אין ל p מידע נוסף בתוך הצומת. במילים אחרות ניתן לומר כי p אינו פעיל.

עבור m: עבור m ועבור בניו ma, mc, mf. יכולנו למצוא את הרצף המתאים בבסיס הנתונים (בתנועות 100, 200, 500). הרצף המתאים הינו acf. רצף זה מופיע שלוש פעמים בבסיס הנתונים ויהיה ניתן לעשות בו שימוש, ולכן מצוין בסמוך ל m כי m ישתמש ב acf בכדי לייצר את תתי הקבוצות הבאות שלו. גם כאן נציין, כי acf מגיע אחרי m בסדר שהוגדר (סדר התדירויות). לכן

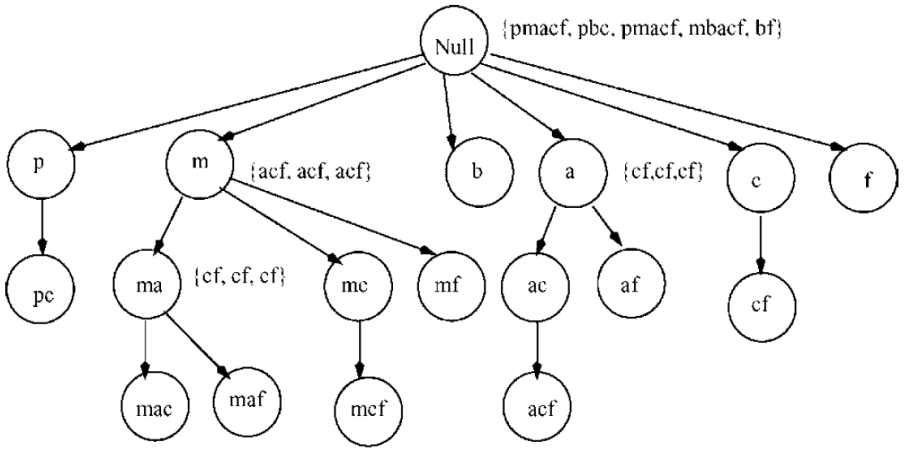
עבור m ניתן לומר כי m הינו פעיל. נציין, כי בזמן בניית המטריצה מתבצעת הטלה של תנועות מבסיס הנתונים אל תוך העץ בצורה הבאה:

נניח וקיימת תנועה בבסיס הנתונים $t = a1a2 \ldots a_n$. ונניח כי התנועה ממוינת על סמך סדר תדירות ההופעה של העצמים בבסיס הנתונים. הטלתה של התנועה הנ"ל תהיה אל תוך צומת a_i תהיה: $t'_{a_i} = a_{i+1}a_{i+2}...a_n$. ניתן לומר כי אנו מתעלמים בעצם מכלל האיברים התנועה שמופיעים לפני הצומת הרלוונטית. וזאת מכיוון שהם אינם רלוונטיים לתהליך הכריה. כיוון שבכלל תהליך הכריה אנו לוקחים בחשבון אך ורק את העצמים שלפנינו.

כעת נעבור לתתי קבוצות תדירות בגודל 3, לאחר בניית המטריצה נקבל את הקבוצות:

mac, maf, mcf, acf

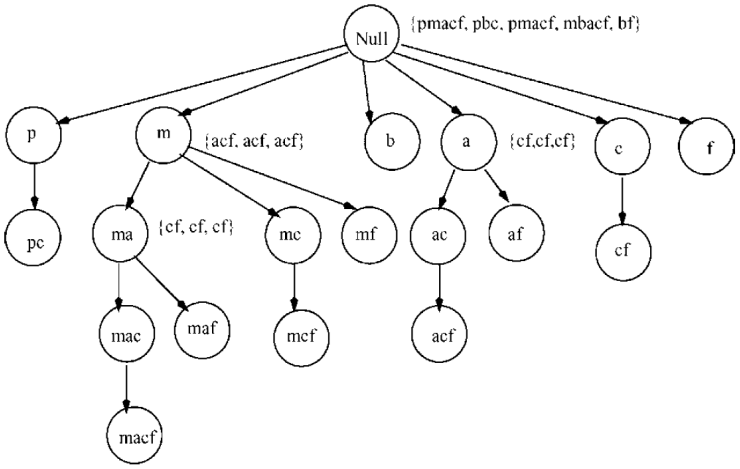
הקבוצות הנ"ל יצורפו לעץ ונקבל:



איור 3.3.6 ה'[14]

נשים לב כי כעת רק ma הינו פעיל. וזאת כיוון שבתנועות 100, 200 ו 500 נוכל למצוא רצף שכולל בתוכו את m ולאחריו a ולאחריהם עצמים נוספים (c ו f).

כעת נעבור לתתי קבוצות בגודל 4:



איור 3.3.6 ו'[14]

כעת העץ בנוי במלואו. מספר הצמתים בעץ לקסיקוגרפי הוא בדיוק מספרם של תתי הקבוצות התדירות. כעת על סמך ההטלה לעץ בבסיס הנתונים ועל סמך תתי הקבוצות שמיוצגות ע״י צמתים בכל רמה בעץ, נוכל לדעת מי הן כל תתי הקבוצות התדירות בבסיס הנתונים הנתון.

יעילותו של האלגוריתם באה לידי ביטוי בשני גורמים:

- ביצוע ההטלה של התנועות מתוך בסיס הנתונים אל העץ מקטינה את מרחב הבעיה שבו אנו עוסקים. הצמצום התמידי בגודלו של העץ מאפשר להתייחס אך ורק לתנועות הרלוונטיות.
- הסדר הלקסיקוגרפי של העץ מקל על ניהולו ועל ספירת המועמדים ועל תהליך הכריה בכלל.

מאידך בהשוואה ל FP – Growth האלגוריתם הנ״ל סובל ממספר חסרונות:

- TreeProjection עלול להתקשות בחישובי המטריצות כאשר מדובר בבסיס נתונים גדול, או כאשר ישנן הרבה תנועות בעלות תתי קבוצות תדירות רבות. כמו כן, יהיה קושי בחישוב המטריצה עבור תמיכה נמוכה במיוחד. מקרים כאלו יגרמו ליצירה של מטריצות ענקיות. עובדה שתקשה על ביצוע האלגוריתם[25]. לעומת זאת FP – Growth לא מצריך כלל בניה של מטריצות כלשהן. מכיוון שבמוצהר אנו נמנעים במהלך האלגוריתם לייצר כל תת קבוצה תדירה שהיא. החישוב היחיד שנעשה הוא של תתי קבוצות בגודל 1.
- מכיוון שתנועה אחת בבסיס הנתונים עלולה להכיל תתי קבוצות תדירות רבות, ייתכן כי תנועה מסוימת בבסיס הנתונים תוטל מספר פעמים למקומות שונים בעץ. כמו כן כאשר קיימות תנועות ארוכות בעלות מספר תתי קבוצות תדירות ההטלה עצמה עלולה לצרוך זמן עיבוד יקר ולהעלות את סיבוכיות הזמן של האלגוריתם.
- מבנה הנתונים של האלגוריתם מייצר צומת אחת בלבד עבור כל תת קבוצה תדירה. במבט ראשון זה נראה חסכני מאוד, אך מבט מעמיק יגלה כי במקרה של תתי קבוצות תדירות רבות וארוכות יגיע העץ למימדים גדולים מאוד[26]. עבור אותה קבוצה FP יזדקק ל 100 צמתים בלבד.

---

[25] קיימת שיטה להקטנת המטריצות במקרים כאלו, לא נעסוק בה כעת.

[26] לדוג׳ עבור תת קבוצה תדירה באורך 100, מספר תתי הקבוצות התדירות(= צמתים) שנקבל יהיה:

$$\binom{100}{50} = \frac{100!}{50! \times 50!} \approx 1.0 \times 10^{29}$$

3.3.7 ביצועים

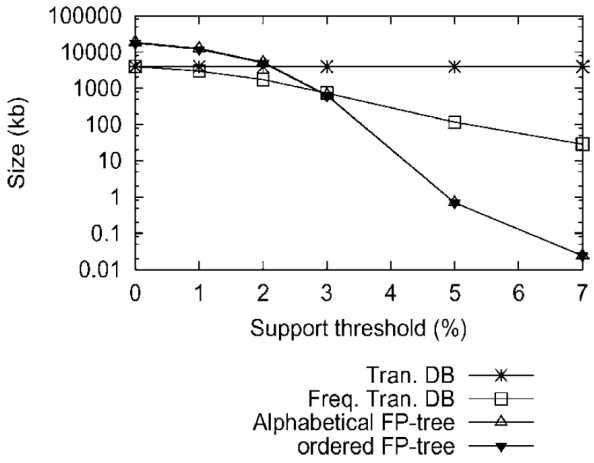
בחלק זה יוצגו בצורה השוואתית תוצאות של מחקרים ובדיקות שנעשו על מנת להבדיל מבחינת ביצועים בין האלגוריתמים שהוצגו עד כה.

<u>סיבוכיות מקום</u>

בבדיקה נבדקו ארבעה סוגים שונים של מבני נתונים:

- Alphabetical FP - Tree - עץ FP רגיל אך הסדר הפנימי של העץ אינו נקבע על סמך תדירות ההופעה אלא על סמך סדר אלפאביתי.
- Ordered FP – Tree – עץ FP רגיל
- Transaction DB – מבנה הנתונים עצמו
- Frequent Transaction DB – תת קבוצה של מבנה הנתונים הקודם. כולל הורדה של כל העצמים שאינם תדירים.

מבדיקות השוואתיות שנעשו[14] בכדי לבחון את מידת הקומפקטיות של מבני הנתונים הנ"ל התקבלו התוצאות הבאות[27]:



איור 3.3.7 א'[14]

ניתן להסיק מגרף זה כמה מסקנות:

- עצי FP משיגים ברוב המקרים תוצאות דחיסה טובות, וזאת הודות ליכולת של שיתוף המידע (תתי קבוצות תדירות) בתוך העץ.
- כאשר התמיכה נמוכה במיוחד עצי FP נעשים סביכים וגדולים, וזאת מכיוון שרמת העצמים המשותפים בעץ קטנה ולכן גודלו של העץ גדל. במקרים כגון זה מומלץ לשקול להשתמש בהטלות של בסיסי נתונים. לאחר ביצוע מספר מחזורים של הטלות ניתן יהיה לבנות עץ FP, ובמקרה הנ"ל לעץ זה שוב תהיה סיבוכיות מקום נמוכה.
- ניתן להסיק[55] גם בצורה ברורה כי סיבוכיות המקום של FP – Growth קטנה בהרבה מזו של אפריורי (עקב מבנה העץ החסכני והשימוש בהטלות).

---

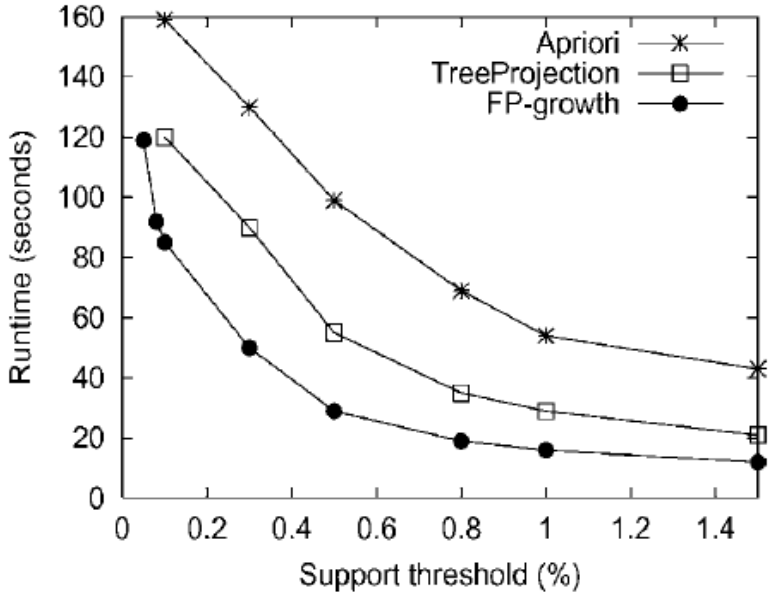
[27] קיימים עוד גרפים נוספים דומים ב [14] . המציגים תוצאות שונות שהורצו על בסיסי נתונים שונים. בכולם התוצאה דומה לכן הבאנו גרף מייצג.

סיבוכיות ריצה

בחלק זה נבדקו בצורה השוואתית זמני הריצה של שלושה אלגוריתמים:

Apriori, Tree Projection, FP – Growth.

הבדיקה נעשתה בתמיכה משתנה מ 0.15% ל 0.01%. באיור 3.3.7 ב' ניתן לראות את תוצאות ההשוואה[28].



איור 3.3.7 ב'[14]

ניתן לראות בצורה ברורה כי FP – Growth הינו האלגוריתם המהיר מבין שלושת האלגוריתמים שנבחנו. כמו כן ככל שהתמיכה יורדת אנו רואים את הפער בין FP – Growth לשאר האלגוריתמים הולך וגדל. שכן מספר העצמים גם הולך וגדל. כפי שניתן לראות מהאיור Apriori מתקשה להתמודד עם תמיכה נמוכה שכן מצב זה מניב מספר רב של עצמים תדירים ולאלגוריתם הנ"ל יש קושי להתמודד עם מספר עצמים רב.

נציין כי מסקנה זו הינה הגיונית וצפויה מכיוון שלעומת אפריורי המבצע n+1 סריקות של בסיס הנתונים (כאשר n הוא אורכה של התנועה הגדולה ביותר) [37]. FP – Growth מצריך, רק שתי סריקות של בסיס הנתונים. ולכן נוכל לומר כי בהינתן כי n הינו גודלו של בסיס הנתונים סיבוכיותו של FP –Growth תהיה O(n), מכיוון ששלב הכרייה בעץ עצמו הינו זניח יחסית למעבר על כלל בסיס הנתונים.

[28] כמובן, שמדובר בהשוואה ספציפית לבסיס נתונים מסוים: במקרה זה מדובר על בסיס נתונים המכיל תנועות בגדלים מעורבים חלק גדולים מאוד וחלק קטנים מאוד. במידה ומדובר רק על בסיס נתונים בעל תנועות גדולות הפער גדול יותר (לטובת FP – Growth).

3.3.8 סיכום

בחלק זה של העבודה הוצג בצורה מעמיקה ויסודית אלגוריתם FP – Growth לכריית תתי קבוצות תדירות מבסיס נתונים. יתרונותיו של האלגוריתם הם[14] [3] :

- בניה של מבנה נתונים קומפקטי וקטן בהרבה מבסיס הנתונים המקורי.
- שימוש בשיטה של pattern growth המונעת יצירה של תתי קבוצות תדירות (מהלך יקר מבחינה חישובית). השימוש בעצים המותנים (Conditional FP Tree) מוודא כי לעולם לא נייצר תתי קבוצות לנתונים שאינם רלוונטיים (לדוג׳ : לא עומדים בדרישות התמיכה המינימלית). בשונה מאלגוריתם אפריורי שבו אנו מחוללים כמות של תתי קבוצות ואז בודקים האם הם עומדים בדרישות התמיכה המינימלית. כאן קבוצות שכאלו פשוט לא ייוצרו.
- נעשה שימוש בשיטת ״הפרד ומשול״, ניתן כך להקטין בצורה ניכרת את גודלם של העצים בהם אנו עושים שימוש. ולהריץ את האלגוריתם כל פעם על תת העץ הרלוונטי.

חוקרים רבים עדיין מנסים לשפר את ביצועי האלגוריתם[29]. לדוג׳ ב [24] בוצעו שני שיפורים שהאיצו את מהירות הריצה של האלגוריתם פי שישה :

- השיפור הראשון כלל שימוש במבנה נתונים – עץ FP משופר התומך ב Caching. שיפור מבנה הנתונים אפשר לעשות שימוש בטכנולוגיות Caching מתקדמות ברמת החומרה.
- שיפור שני הרחיב את האלגוריתם להרצה על מעבדים מרובי ליבות, ע״י הוספה של מנגנון מקביליות ללא נעילות לעץ.

חוקרים רבים גם הרחיבו את יכולותיו של האלגוריתם ושילבו עקרונות בסיסיים ממנו באלגוריתמים אחרים לדוגמא : FGP המשלב בין FP ל GP – Close[30], בשיטה זו משתמשים ב FP לכרייה של חוקי הקשר בשילוב קטגוריות. [8]

---

[29] מימוש משופר נוסף ניתן לראות ב [1]

[30] להרחבה על כריית חוקי הקשר בשילוב קטגוריות ראה [31]

***ECLAT* 3.4**

3.4.1 מבוא

אלגוריתם Eclat (**E**quivalence **CL**Ass **T**ransformation) הוצג לראשונה ב [36] כחלק מתוך שישה אלגוריתמים חדשים לכריית חוקי הקשר.

קיימים ארבעה מאפיינים לאלגוריתמים הללו (ו Eclat בתוכם):

- שימוש בבסיס נתונים מסוג vertical tid-list database, במצב זה אנו משייכים לכל קבוצת נתונים (itemset), רשימה של תנועות בבסיס הנתונים בהן היא מופיעה.
- נעשה שימוש בגישת רשת תיאורטית (Lattice theoretic approach) בכדי לפצל את מרחב החיפוש למספר תתי רשתות בגודל קטן יותר. תתי רשתות אלו יעובדו במקביל. יוצגו להלן שתי שיטות לביצוע הפירוק לתתי רשתות:
  - Prefix Based
  - Maximal Click Based
- בעיית החלוקה לתתי רשתות תופרד לחלוטין מבעיית החיפוש בבסיס הנתונים. לבעיית החיפוש יוצגו שלושה פיתרונות שונים:
  - Bottom Up
  - Top Down
  - Hybrid Search
- השיטה תצריך רק מספר מועט של סריקות בבסיס הנתונים.

3.4.2 תיאורית הרשת

<u>הגדרות ומינוחים</u>

בכדי להבהיר את המינוחים בהם נעשה שימוש בתיאור האלגוריתם יש צורך בהגדרות ומשפטים שישמשו אותנו בהמשך. לכן, נגדיר את ההגדרות הבאות [36]:

1. נניח כי P הינה קבוצה. סדר חלקי (Partial Order) על P הינו היחס $\leq$ כך שעבור כל $X,Y,Z \in P$ היחס יהיה:
   - רפלקסיבי: $X \leq X$
   - אנטי סימטרי: $X \leq Y$ וגם $Y \leq X$ גורר ש X = Y.
   - טרנזיטיבי: $X \leq Y$ וגם $Y < Z$ גורר ש X≤Z.

   הקבוצה P עם היחס $\leq$ נקראת קבוצה סדורה.
2. נניח ו P הינה קבוצה סדורה, ו $X,Y,Z \in P$. ניתן לומר כי X *מכוסה ע"י* Y. (יסומן ב: $X \lessdot Y$). אם X < Y ו $X \leq Z < Y$ גורר Z = X. כלומר לא קיים שום איבר Z מתוך P המקיים $X < Z < Y$.
3. נניח ו P הינה קבוצה סדורה, ו $S \subseteq P$. איבר $x \in P$ יהיה *גבול עליון* של S אם $s \leq X$ עבור כל $s \in S$ (ובהתאמה ההפך עבור *גבול תחתון*). הגבול העליון הקטן ביותר ייקרא צירוף (join) של S, ויסומן ב $\vee s$. הגבול התחתון הגדול ביותר ייקרא meet של S ויסומן ב $\wedge s$. האלמנט המקסימלי ב S יסומן ב T, ויכונה האלמנט העליון (top element). האלמנט המינימלי יכונה האלמנט התחתון (bottom element) ויסומן ב $\perp$.
4. נניח ו L הינה קבוצה סדורה. L תיקרא[31] *רשת צירוף למחצה* (join semilattice), אם $X \cup Y$ קיים עבור כל $X,Y \in L$.[32] L תיקרא *רשת* אם היא גם רשת צירוף למחצה וגם meet semilattice. כלומר אם $X \cup Y$ ו $X \cap Y$ קיימים עבור כלל הזוגות $X,Y \in L$. L תיקרא *רשת שלמה*, אם $\wedge s$ $\vee s$ קיימים עבור כלל תתי הקבוצות המקיימות $S \subseteq L$. *קבוצה סדורה* $M \subset L$ תיקרא *תת רשת* של L אם $X,Y \in M$ גורר כי $X \cup Y \in M$ ו $X \cap Y \in M$
5. עבור קבוצה S, הקבוצה הסדורה (power set of s) – P(s). הינה רשת מלאה שבה join ו meet נתונים ע"י איחוד וחיתוך בהתאמה:

$$\bigvee\{A_i \mid i \in I\} = \bigcup_{i \in I} A_i \qquad \bigwedge\{A_i \mid i \in I\} = \bigcap_{i \in I} A_i$$

---

[31] ההגדרה ניתנת גם בצורתה ההפוכה עבור meet semilattice, כאשר כלל האופרטורים הפוכים כלומר: $x \cap y$.

[32] אנו דורשים כאן בעצם סגירות תחת פעולת ה meet (או ה Join)

האיבר העליון של P(S) יהיה T = S, האיבר התחתון יהיה $\perp = \{\}$. עבור כל $L \subseteq P(S)$. ניתן לומר כי L נקראת *רשת של קבוצות*. אם L סגורה ע"י מספר סופי של חיתוכים ואיחודים.

<u>דוגמא:</u>

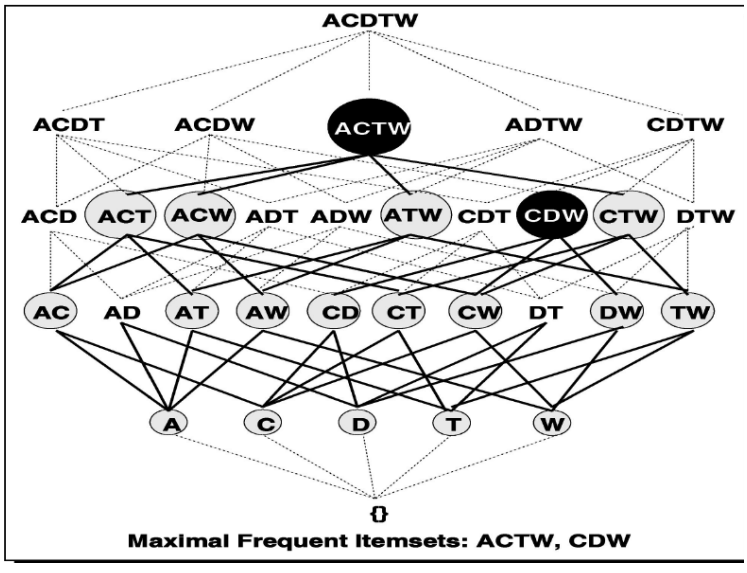
בהינתן בסיס הנתונים $I = \{A, C, D, T, W\}$. בסיס הנתונים מופיע באיור 3.4.2 א'

DATABASE

Transaction	Items
1	A C T W
2	C D W
3	A C T W
4	A C D W
5	A C D T W
6	C D T

איור 3.4.2 א' [36]

P(I) תיראה כך[33]:



איור 3.4.2 ב' [36]

[33] בהמשך נסביר את דרך הבניה כעת נתמקד בהגדרות של הקבוצות.

נציין כי כלל הקבוצות התדירות צבועות באפור והקבוצות התדירות המקסימליות צבועות בשחור. ניתן לראות כי הקבוצה של כל תתי הקבוצות התדירות יוצרת רשת למחצה מסוג meet. וזאת מכיוון שרשת זו סגורה תחת פעולת ה meet. כלומר חיתוך של שני איברים השייכים לקבוצת תתי הקבוצות התדירות יוצר תוצאה שגם היא שייכת לקבוצת תתי הקבוצות התדירות. מאידך זוהי לא רשת צירוף למחצה מכיוון שאין כאן סגירות תחת פעולת הצירוף(join). כלומר בהינתן ש X ו Y הינם תתי קבוצות תדירות לא נוכל לדעת בוודאות כי $X \cup Y$ הינן גם תדירות. ניתן לומר כאן כי העצמים הלא תדירים מקיימים ביניהם רשת צירוף למחצה. כלומר, תתי הקבוצות הלא תדירות סגורות תחת פעולת ה join. ניתן אם כן לומר כי איחוד של שתי תתי קבוצות לא תדירות יצור גם קבוצה לא תדירה. ומאידך חיתוך של תתי קבוצות לא תדירות לא יצור בהכרח קבוצה לא תדירה.

<u>משפטים נוספים[34]:</u>

1. כל תתי הקבוצות של תתי קבוצות (וקבוצות) תדירות הם תדירות.
   משפט זה נובע ישירות מהסגירות תחת פעולת ה meet עבור הקבוצה של תתי הקבוצות התדירות (כפי שהסברנו מקודם). כמו כן, נוכל לומר כי כל תתי הקבוצות של קבוצה שאינה תדירה הינם גם כן לא תדירים. הנחה זו היא בעצם תכונת אפריורי הידועה לנו מאלגוריתם אפריורי. עד היום נעשה שימוש בתכונה זו לסינונים של תתי קבוצות תדירות במהלך חיפוש בצורת bottom up (כדוגמת אפריורי). בהמשך נציג שימוש נוסף של תכונה זו.
2. תתי הקבוצות התדירות המקסימליות מייצגות בצורה חח"ע את כל תתי הקבוצות התדירות.
   משפט זה מוביל אותנו לרצון למצוא שיטת חיפוש שמוצאת בצורה מהירה את תתי הקבוצות התדירות המקסימליות.

3.4.3 חישוב תמיכה

גם בחלק זה נתחיל בהגדרות בסיסיות שישמשו אותנו בהמשך:

<u>הגדרות ומינוחים</u>

1. רשת L תיקרא "ניתנת לפילוג" (distributive) אם מתקיים עבור כל $x, y, z \in L$: .
   $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$.
2. נניח ו L הינה רשת עם איבר תחתון $\perp$. במצב כזה, $X \in L$ יקרא *אטום*(Atom) אם - $X \neq \perp$ - x) "מכסה" את $\perp$) קבוצת האטומים של L תסומן ב A(L).
3. רשת L תכונה רשת בוליאנית אם:
   a. היא ניתנת לפילוג
   b. בעלת איבר תחתון ועליון.
   c. לכל איבר ב L קיימת קבוצה משלימה.

---

[34] במסגרת זו לא נוכיח את המשפטים בצורה מלאה אלא נספק להם הסבר.

נציין כי הרשת באיור 3.4.2 א׳ הינה רשת בוליאנית מהסיבות הבאות:

- מתקיים כי עבור כל $X \in L$ קיימת קבוצה משלימה – $I \setminus X$. (כלל c)
- קל לראות כי הרשת ניתנת לפילוג. האיברים מקיימים את חוק הפילוג.(כלל a)
- האיבר התחתון והעליון יוגדרו על סמך התמיכה המינימילית.(כלל b).

נציין כי כי רשימת האטומים לרשת הנ״ל תואמת לרשימת העצמים בבסיס הנתונים. כלומר : $A(P(I)) = I$. כאשר I הינה רשימת הנתונים. נשייך לכל אטום (עצם בבסיס הנתונים) רשימת tid לציון התנועות שבהן הוא מופיע בבסיס הנתונים. רשימה זו תסומן ב $\mathcal{L}(X)$ באיור 3.4.3 א׳ ניתן לראות דוגמא לשיוך הנ״ל.

A	C	D	T	W
1	1	2	1	1
3	2	4	3	2
4	3	5	5	3
5	4	6	6	4
	5			5
	6			

איור 3.4.3 א׳ [36]

כלומר, העצם A מופיע בתנועות 1,3,4,5. העצם T מופיע ב 1,3,5,6 וכדו׳.

4. עבור רשת בוליאנית סופית L, עבור $X \in L$ מתקיים[35]: $X = \bigvee \{Y \in \mathcal{A}(L) \mid Y \leq X\}$
כלומר, כל אלמנט ברשת בוליאנית, נתון כצירוף (join) של תת קבוצה של קבוצת אטומים. מכיוון ש P(I) הינה רשת בוליאנית נקבל :
5. עבור כל $X \in P(I)$ נגדיר $J = \{Y \in A(P(I)) \mid Y \leq X\}$. נוכל לומר כי : $X = \bigcup_{Y \in J} Y$
אזי $\sigma(X) = \mid \bigcap_{Y \in J} \mathcal{L}(Y) \mid$.[36] בעצם אנו אומרים כי אם קבוצת עצמים נתונה כאיחוד של קבוצת עצמים מ J. אזי התמיכה[37] של הקבוצה הנ״ל נתונה כחיתוך של כלל רשימות ה tid של העצמים מ J. בעצם, ניתן יהיה להגדיר את התמיכה של כל k – itemset רק ע״י ביצוע חיתוך של רשימות ה – tid של שנים מתוך תתי קבוצותיו שהן באורך של k-1. בדיקה של העוצמה של קבוצת התוצאה תוכל בקלות לומר לנו אם קבוצת העצמים

---

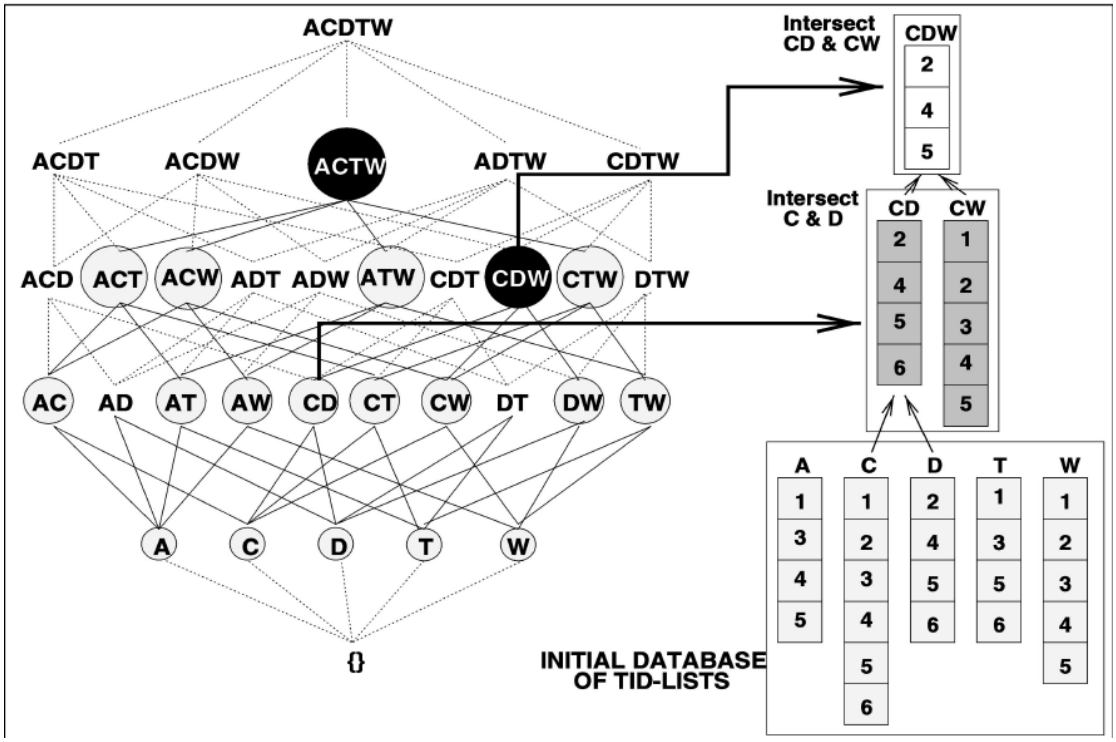
[35] הוכחה של משפט זה חורגת ממסגרת עבודה זו.

[36] ניתן להכליל את המשפט גם עבור קבוצות של itemsets : עבור כל $X \in P(I)$ נגדיר $X = \bigcup_{Y \in J} Y$ אזי $\sigma(X) = \mid \bigcap_{Y \in J} \mathcal{L}(Y) \mid$

[37] במקרה הנ״ל התמיכה מסומנת ב $\sigma(x)$

החדשה היא תדירה או לא. וזאת מכיוון שהעוצמה מעידה על מספר התנועות שעצם זה מופיע בהן בבסיס הנתונים. בחישוב של מספר המופעים ביחס למספר התנועות הכולל נקבל את התמיכה.

6. נניח ש X ו Y הינם שני קבוצות נתונים(itemsets). ונניח מתקיים כי $X \subseteq Y$. אזי ניתן לומר כי $\mathcal{L}(X) \supseteq \mathcal{L}(Y)$. ניתן להוכיח משפט זה בקלות מכיוון שעובדה זו נובעת ישירות מהגדרת התמיכה. בכל מקום ש X מופיע גם Y יופיע אך לא ההפך. מסקנה חשובה הנובעת ממשפט זה היא שהעוצמה של רשימת ה tid קטנה ככל שאנו מתקדמים במבנה הרשת. (מכיוון שככל שהקבוצה גדולה יותר ה tid list שלה קטן. וזאת מכיוון שככל שהקבוצה גדלה יש לה פחות מופעים בבסיס הנתונים. ניקח לדוגמא את C אורך הרשימה שלו הינה 6 , כי C מופיע ב 6 תנועות בבסיס הנתונים. לעומת זאת אם ניקח את ACTW נקבל שאורך הרשימה יהיה 3 (כי הצירוף הנ״ל מופיע רק 3 פעמים בבסיס הנתונים).



איור 3.4.3 ב'[36]

באיור 3.4.3 ב' ניתן לראות את תהליך בניית הרשת. ניתן להבחין בפינה הימנית התחתונה בבסיס הנתונים (בדומה ל איור 3.4.3 א'). כעת עלינו לחשב עבור כלל התמורות האפשריות של בסיס הנתונים את התמיכה. כפי שניתן לראות כלל האפשרויות לסידור של האטומים בבסיס הנתונים בקבוצות מובאות כאן בצורת רשת. אפשרויות אלו התקבלו ע"י ביצוע של צירופים של קבוצות שונות. לדוגמא: את AT נקבל מתוך A ו T. מתוך צירוף של 2 תתי קבוצות בגודל X נקבל קבוצה בגודל X+1. כעת עלינו לחשב את התמיכה של כל אחת מתתי הקבוצות - נדגים עבור חלק מהן:

נראה כיצד יוצרים רשימת tid עבור CD (עפ"י משפט – 5). כפי שאמרנו CD יתקבל ע"י חיתוך של C ו D. באותה דרך CDW יתקבל ע"י חיתוך של CD עם CW.

על פי זה נחשב גם את התמיכה: התמיכה של CD תורכב מחיתוך התמיכה של C ושל D, כלומר: $\{1,2,3,4,5,6\} \cap \{2,4,5,6\} = \{2,4,5,6\}$ - כלומר 4.

לגבי CDW ניתן לראות כי עוצמת רשימת האינדקסים בבסיס הנתונים הינה 3 $\{2,4,5\}$, לכן, התמיכה של CDW תהיה 3.

א"כ התמיכה (באחוזים) תהיה העוצמה של קבוצת החיתוך מתוך מספר התנועות הקיים. מספר זה ייתן לנו בעצם את מס' התנועות בבסיס הנתונים (והתמיכה תחושב ע"י: מס' התנועות בהן מופיע מתוך סך התנועות הכולל).

באופן תיאורטי ניתן היה להשתמש ברשת זו, אך עקב מגבלות זיכרון ויעילות לא מומלץ להשתמש בה. מה שמוביל אותנו הישר לסעיף הבא.

3.4.4 פירוק הרשת – מחלקות מבוססות תחילית

נניח והייתה לנו כמות די גדולה של זיכרון במחשב, ניתן היה לבנות רשת כמו בסעיף 3.4.3, ולבצע חיתוכים על מנת לחשב את התמיכה של תתי קבוצות שונות. מכיוון שאנו מוגבלים בכמות הזיכרון כלל מבני הנתונים שהוצגו לעיל לא יהיו שימושיים (tid list, רשת).

אנו רוצים לנסות לפרק את הרשת לתתי רשתות שניתנות לכרייה באופן בלתי תלוי. לצורך מענה לשאלה זו נגדיר את ההגדרות והמשפטים הבאים[32]:

1. נניח ו P הינה קבוצה. *יחס שקילות* על P הינו יחס בינארי ≡ כך שמתקיים כי עבור: $x, y, z \in P$ היחס הבא:
   a. רפלקסיביות: $x \equiv x$
   b. סימטריה: $x \equiv y$ גורר $y \equiv x$
   c. טרנזינטיביות: $x \equiv y$ ו $y \equiv z$ גורר $x \equiv z$

יחס השקילות מחלק את P לתתי קבוצות מנותקות שיקראו *מחלקות שקילות*. מחלקת שקילות של $x \in P$ היא $[X] = \{Y \in P \mid X \equiv Y\}$

נגדיר את הפונקציה הבאה: $p : P(I) \times N \mapsto P(I)$[38] כאשר p(X, k)= X[1:k], כלומר התחילית באורך k של X. נגדיר יחס שקילות θ_k על הרשת P(I):

$$\forall X, Y \in P(I), X \equiv_{\theta_k} Y \Leftrightarrow p(X,k) = p(Y,k)$$

נוכל לשייך שני תתי קבוצות של עצמים (itemsets) לאותה קבוצת שקילות. אם הם יחלקו את תחילית משותפת באורך k. נוכל לכנות את θ_k – יחס שקילות מבוסס תחיליות.

באיור 3.4.4 א' ניתן לראות את הרשת לאחר הפעלת החלוקה לקבוצת שקילות בשימוש ביחס θ_1 על P(I). כלומר בעצם אנו נכווץ את כל תתי הקבוצות בעלות תחילית משותפת באורך 1 אל תוך קבוצת שקילות אחת. קבוצות השקילות יהיו :[A,C,D,T,W,{}].

2. כל קבוצת שקילות $[X]_{\theta_k}$ הנוצרת ע"י יחס שקילות θ_k הינה תת רשת של P(I)

<u>הוכחה:</u>

נניח[32] כי U ו V, הינם שני אלמנטים ממחלקת השקילות [X]. לכן אנו יודעים שלשניהם יש את אותה תחילית – X. ולכן שניהם שייכים ל [X].

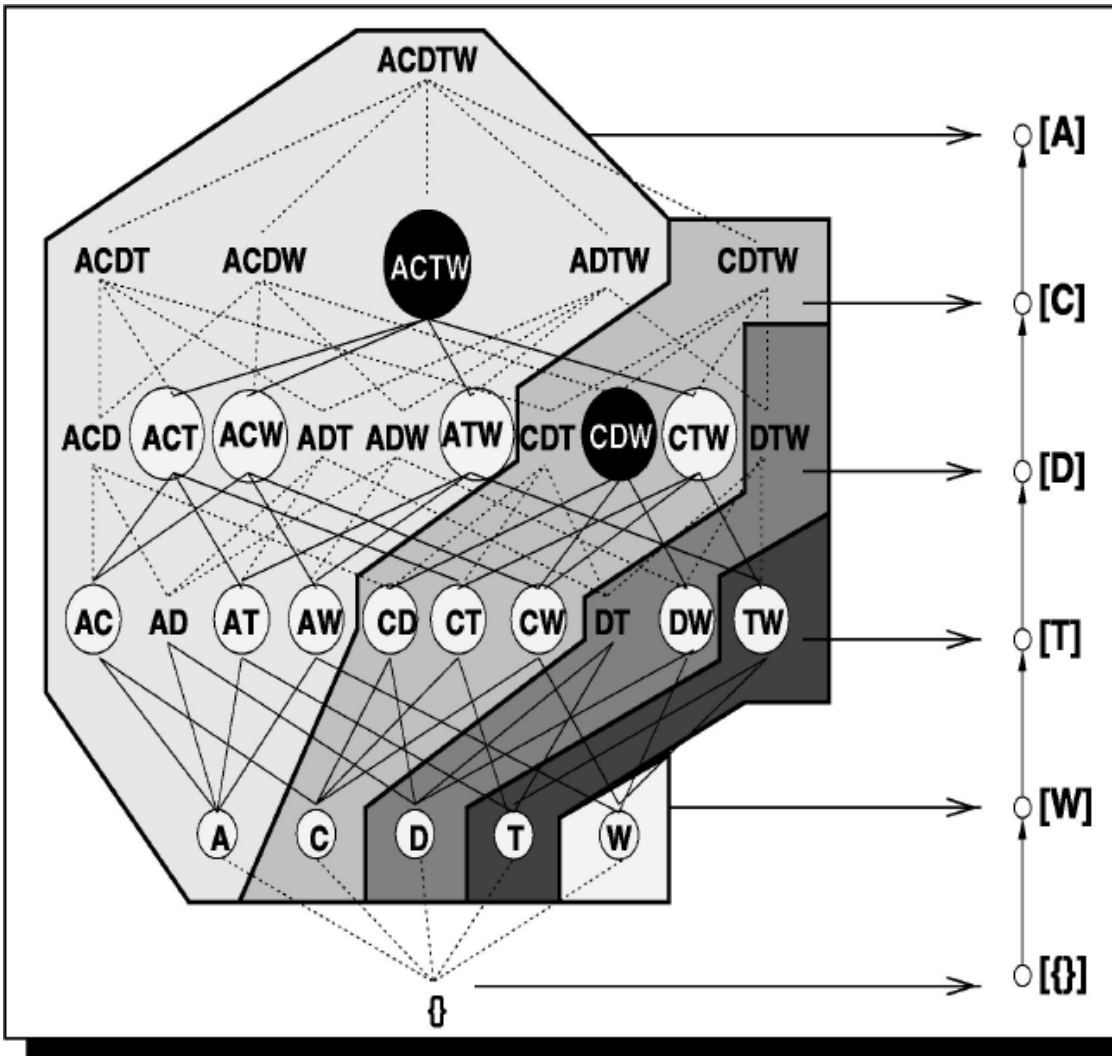
מכיוון ש[39] $U \cup V \supseteq X \Rightarrow U \cup V \in [X]$ וגם $U \cap V \supseteq X \Rightarrow U \cap V \in [X]$

על סמך משפט 4 – הגדרת תת רשת בפרק 3.4.2. נוכל לומר כי $[X]_{\theta_k}$ הינה תת רשת של P(I).

---

[38] לצורכי הבהרה של הפונקציה: נעשה מיפוי בין P(I) X N ל P(I).

[39] איחוד של שתי קבוצות עם תחילית זהה ייתן קבוצה שלישית בעלת אותה תחילית. כנ"ל לחיתוך, מכיוון שבמקרה זה תמיד תהיה התחילית שתהיה שווה בין שניהם, גם במקרה שהיא האיבר היחידי הזהה, עדיין היא תישאר בשניהם ולכן התוצאה תהיה שייכת ל [X].



איור 3.4.4 א'[36]

נציין, כי כל $[X]_{\theta_1}$ הינו רשת בוליאנית בעצמו עם קבוצת אטומים עצמית. לדוג' האטומים של $[A]_{\theta_1}$ יהיו {AC, AD, AT, AW}. האיבר המקסימלי והמינימלי יהיו T = ACDTW ו $A = \perp$.

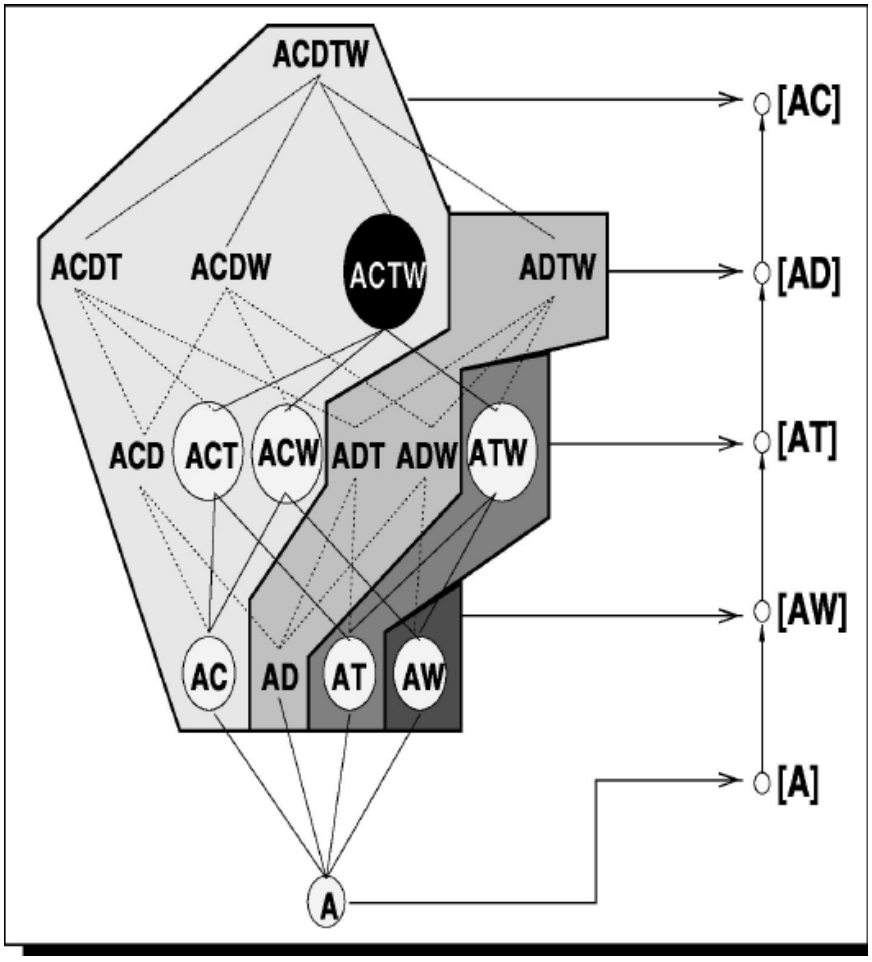
ניתן על פי משפט 5 ב 3.4.3 למצוא את התמיכה של כלל תתי הקבוצות בכל קבוצת שקילות (תת רשת). ע"י חיתוך של ה tid lists של כל עצם (כפי שתואר לעיל) או של שני תתי קבוצות המצויות רמה אחת מתחת הרמה הנבדקת. אם היה לנו מספיק זיכרון היינו יכולים לשמור עבור כל קבוצה רשימות tid זמניות רלוונטיות וכך ניתן היה לחשב תמיכה עבור כל $[X]_{\theta_1}$ בצורה בלתי תלויה לשאר הקבוצות. באיור 3.4.4 א' ניתן לראות כיצד הרשת מתחלקת לקבוצות שקילות שונות בהתאם לעצמים אותם כל קבוצת שקילות מייצגת. לדוגמא: אם נסתכל בחלק השמאלי של איור 3.4.4 א' נוכל לראות את קבוצת השקילות A. קבוצה זו מתחילה את בנייתה מA. לאחר מכן נוכל לראות ברמות מתחת ל A את כלל הצירופים בכל גודל הקיימים עבור כל רמה בתת הרשת של A. כנ"ל לגבי C. כפי שכבר צוין לעיל כל תת רשת שכזו ניתנת לכרייה באופן בלתי תלוי ברשת השניה.

ניתן לראות כי קשרים בין מחלקות שקילות מציינים גם תלויות. כלומר אם נרצה לסנן קבוצת עצמים שמכילה תת קבוצה שאינה תדירה, יהיה עלינו לעבור על קבוצות השקילות מהסוף להתחלה. כלומר מזו שבעלת פחות קשרים לזו שיותר. ובדוגמא שלנו מ[W] עד [A]. צורה זו מבטיחה שכלל המידע הרלוונטי לסינון יהיה נגיש. כך תמיד שנבוא לבדוק האם תת קבוצה היא תדירה יהיה בידינו המידע משאר הקבוצות שעובדו קודם לכן שיסייע לנו בהחלטה .

בסופו של דבר, ברוב המקרים חלוקה לרמה אחת של הרשת ע"י θ_1 תספיק. במקרים בהם הקבוצות שיתקבלו יהיו עדיין גדולות מדי עבור הזיכרון הנתון נבצע חלוקה מחודשת בצורה רקורסיבית לתת הרשת הבעייתית. נניח ובאיור 3.4.4 א' [A] הינה גדולה מדי עבור הזיכרון הקיים. מכיוון ש [A] בעצמה הינה רשת בוליאנית, ניתן יהיה לפרקה ע"י שימוש ב θ_2. כלומר, נקבץ את כלל העצמים בעלי תחילית משותפת באורך שתים ביחד. הקיבוץ יעשה ע"י בחירה מהרשת הקודמת (3.4.4 א') את כלל הצמתים בגודל 2 בעלי התחילית A. החלוקה תימשך עד שמגבלות הזיכרון יוסדרו – ואז ניתן יהיה להריץ את האלגוריתם ישירות.

הקבוצות שיתקבלו יהיו:

{[AC], [AD], [AT],[AW]}. באיור 3.4.4 ב' ניתן לראות את תת החלוקה.

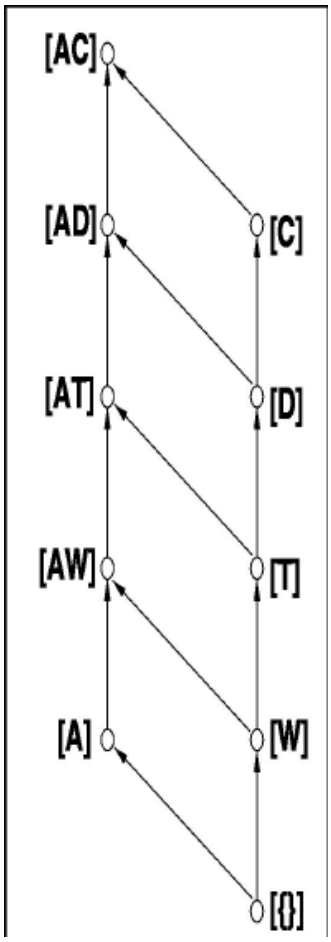


איור 3.4.4 ב'[36]

התלויות בין הקבוצות בצורה הסופית של הרשת מובאות באיור 3.4.4 ג'

באיור זה ניתן לראות את הקשרים בין הקבוצות של מחלקות השקילות. כפי שהסברנו לעיל אם כל [D] הייתה לא תדירה אזי כל האיברים ב [AD] היו לא תדירים. כמובן שגם כעת ניתן להמשיך לחלק את הרשת לתתי רשתות עד שהגודל יתאים לגודלו של הזיכרון.

`



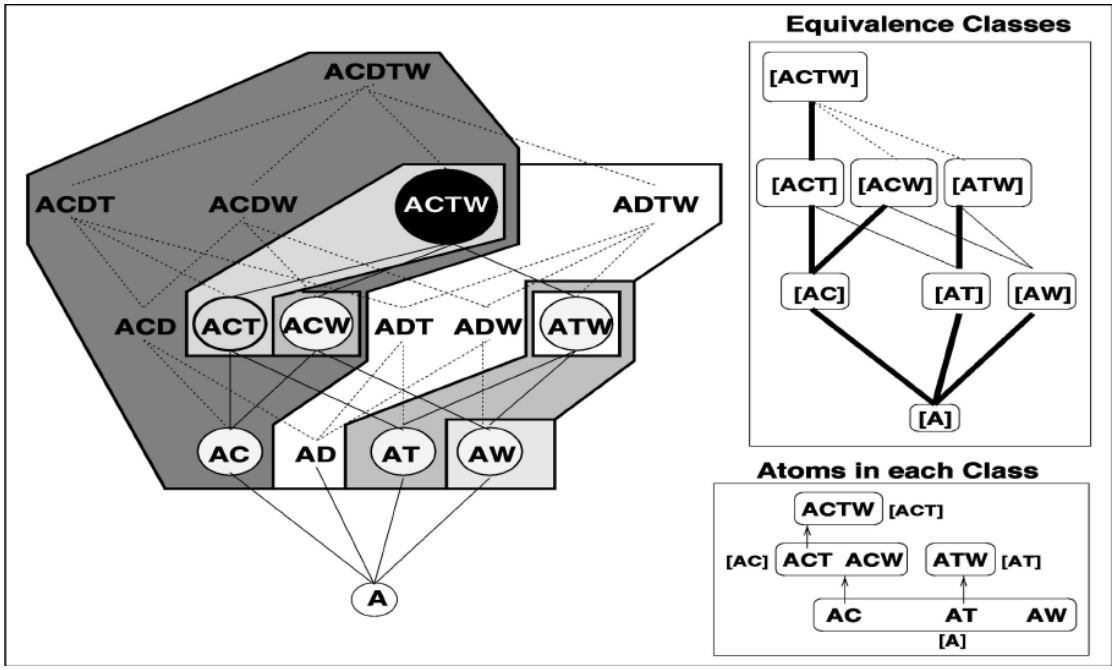
איור 3.4.4 ג'[36]

3.4.5 חיפוש תתי קבוצות תדירות

בחלק זה נדון ונציג שיטות יעילות לחיפוש תתי קבוצות תדירות בכל מחלקת שקילות.

<u>חיפוש מלמטה למעלה (Bottom – Up Search)</u>

חיפוש מלמטה למעלה מבוסס על פירוק רקורסיבי של כל מחלקה לכמה תתי מחלקות שנוצרו ע"י יחס השקילות θ_k . באיור 3.4.5 א' ניתן לראות את החלוקה של $[A]_{\theta_1}$ לתתי מחלקות.[40] כמו כן באיור מוצגת החלוקה לקבוצת אטומים על פי כל תת מחלקת שקילות.ניתן לסרוק את הרשת שהתקבלה בשני אופנים: DFS(Depth First Search) או BFS (Breadth First Search). במאמר זה נעשה שימוש רק ב BFS[41].



איור 3.4.5 א'[36]

משמעות השימוש באופי חיפוש זה תבוא לידי ביטוי בסדר הביקור בצמתים בעץ. הסדר יהיה מלמטה למעלה אך לרוחב בכל רמה:

[A] לאחר מכן [AW][AT][AC] לאחר מכן [ATW][ACW][ACT] ולבסוף, [ACTW]. (נשים לב שאנו מבקרים רק את תתי הקבוצות התדירות). כפי שכבר צוין לעיל, בכדי לחשב את התמיכה עבור כל תת קבוצה, יהיה עלינו לבצע חיתוך של רשימות ה tid עבור שני תתי קבוצות מהקבוצה הקודמת לקבוצה הנוכחית. מכיוון שאנו עוברים על כלל הצמתים ב BFS , בעצם נמצא כך את התמיכה עבור כל תתי הקבוצות.

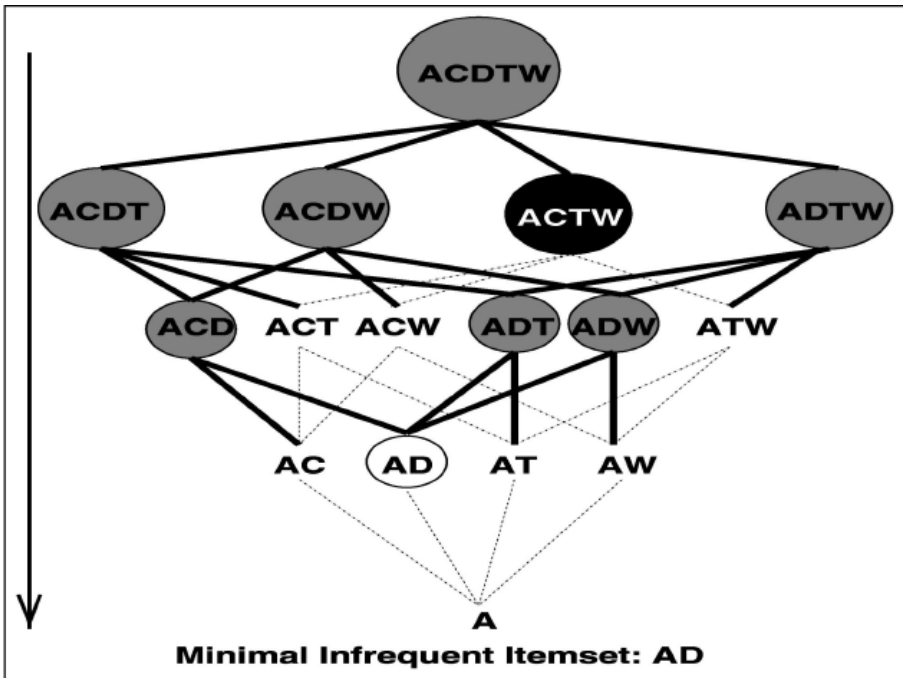
---

[40] מעין המשך לאיור 3.4.4 ב'. נציין כי כאן נתייחס אך ורק לתתי קבוצות המכילות איברים תדירים. תתי קבוצות ללא איברים תדירים לדוג' [AD], לא יופיעו באיור.

[41] בהמשך נראה כי אחד ממאפייני Eclat הינו עובדת היותו אלגוריתם DFS, למרות זאת כאן נעשה שימוש ב BFS.

<u>חיפוש מלמעלה למטה (Top Down)</u>

החיפוש בשיטה זו מתחיל עם האיבר העליון של הרשת. התמיכה שלו מחושבת על ידי חיתוך של רשימות ה tid של האטומים המרכיבים אותו. כמובן שבמידה וגודלו יהיה k יהיה עלינו לבצע k חיתוכים. אם האיבר העליון הינו תדיר אזי סיימנו את החיפוש מכיוון שכל בניו הם גם תדירים. אם לא מדובר באיבר תדיר, נבדוק כל תת קבוצה ברמה הבאה. תהליך זה יימשך עד אשר נזהה את כל תתי הקבוצות הלא תדירות המינימליות. באיור 3.4.5 ב׳ ניתן לראות דוגמא לתהליך החיפוש. באיור יסומנו תתי הקבוצות המקסימליות[42].החיפוש יתחיל ב ACDTW מכיוון שהוא אינו תדיר, נעבור לבנים שלו, מתוכם רק ACTW תדיר, לכן נסמן גם את כל צאצאיו כתדירים. כך נמשיך בחיפוש בעומק העץ. עד שנזהה את AD – תת הקבוצה הלא תדירה בגודל מינימלי.



איור 3.4.5 ב׳[36]

החץ באיור 3.4.5 ב׳ מציין את כיוון החיפוש

[42] נציין כי תת קבוצה מקסימלית בתוך תת רשת אינה מקסימלית בהכרח ברשת כולה. וזאת מכיוון שיתכן שבשאר הרשת קיימות תתי קבוצות הגדולות יותר מרותה תת קבוצה.

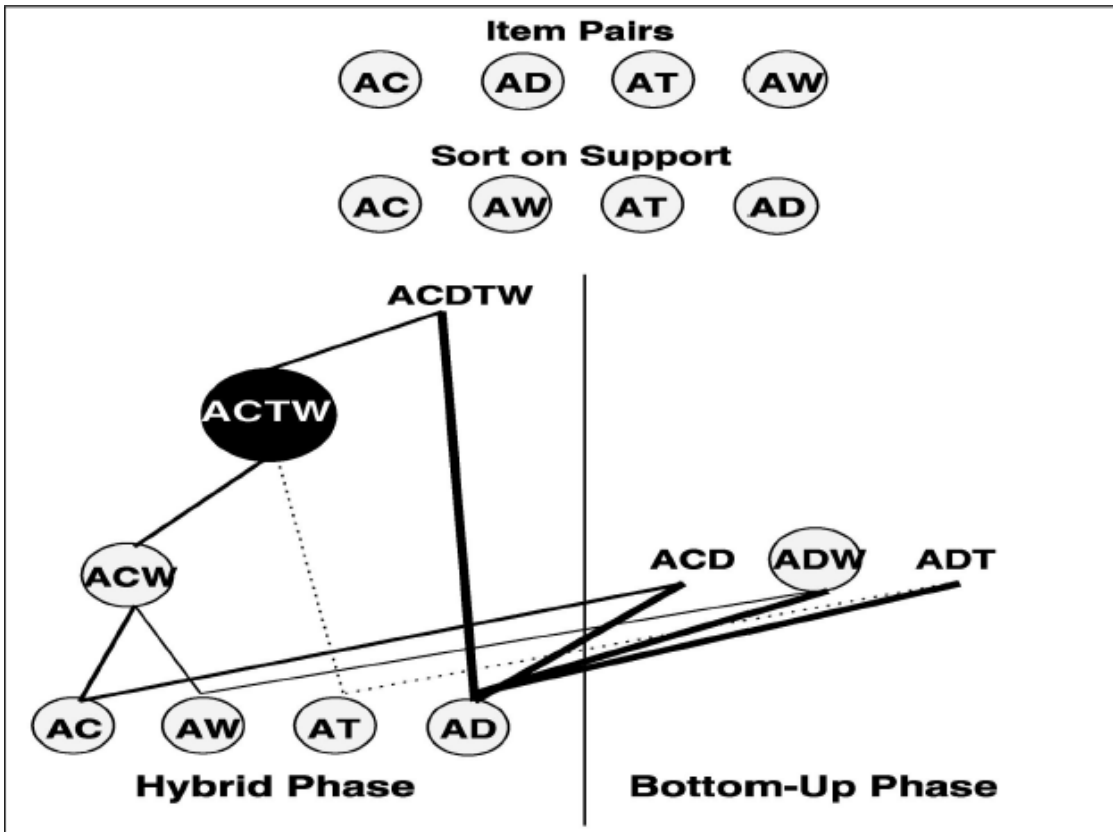
<u>חיפוש היברידי (Hybrid Search)</u>

גישה זו מבוססת על האינטואיציה שככל שהתמיכה של תת קבוצה תדירה גדולה יותר, כך הסיכויים שקבוצה זו היא חלק מתת קבוצה תדירה גדולה יותר, גדולים יותר.

ישנם שני שלבים עיקריים לגישה זו:

נתחיל עם האטומים של המחלקה ממוינים בסדר יורד על סמך התמיכה שלהם.

- בשלב הראשון נתחיל לבצע צירופים של האטומים, אחד בכל פעם. כך נקבל קבוצות תדירות גדולות יותר ויותר(בכל פאזה). התהליך יפסק כאשר התוספת הופכת ללא תדירה.
- בשלב השני: נעבור לשלב ה bottom up: האטומים שנשארו יחוברו עם האטומים של הקבוצה הראשונה (יוסבר באיור 3.4.5 ג') וכך נוכל לייצר את כל שאר תתי הקבוצות התדירות.



איור 3.4.5 ג'[36]

נציין כי באיור 3.4.5. ג' לצורכי הבהרה הנחנו כי AD ו ADW הינן תדירות.

כפי שציינו לעיל, החיפוש מתחיל בסידור של תתי הקבוצות בגודל 2 על פי התמיכה שלהן.כאשר הכי תדיר ראשון(כלומר, מדובר בסדר יורד).

לאחר מכן נתחיל לחפש את תת הקבוצה התדירה המקסימלית ע"י שימוש בצירופים: אנו משתמשים בעובדה שאומרת שאם קבוצה תדירה אזי גם כל תתי קבוצותיה תדירות.

AC יצורף עם AW וכך נקבל את ACW שהוא גם תדיר. נרחיב את התוצאה עם האיבר הבא AT ונקבל ACTW. כאשר נוסיף את AD התהליך ייפסק מכיוון ש ACDTW אינה קבוצה תדירה.

המסקנה היא שהשלב ההיברידי מצא את תת הקבוצה התדירה המקסימלית ACTW.

כעת נעבור לשלב ה Bottom Up. בשלב זה AD יצורף עם כלל הזוגות הקודמים שהוזכרו לעיל בכדי שנוודא את שלמות תהליך החיפוש, כך שלא דילגנו על אף צירוף. כאן יתקבלו כתוצאה מהצירוף: ACD ADW ו ADT. חיפוש על ADW ו ADT ניתן יהיה להגדיר בתור חיפוש במחלקת השקילות [AD], הליך שכבר הראנו כיצד היא ניתנת לפיתרון בחיפוש bottom up.

3.4.6 פירוק הרשת - גישת הקליקה המקסימלית

בשיטה זו נייצר (בשונה מ 3.4.4) תתי רשתות (ומחלקות שקילות) קטנות יותר ממקודם. ככל שתת הרשת קטנה יותר היא מכילה פחות אטומים ופחות חיתוכים[43]. הקטנת מספר האטומים מקטינה באופן ישיר את מספר החיתוכים במהלך חיפוש bottom up. כנ"ל לגבי מספר החיתוכים בסכמה ההיברידית. כמו כן במהלך חיפוש top – down יוקטן גודלו של האלמנט המקסימילי. בכדי להבהיר את שיטת הפירוק בחלק זה נגדיר מס' הגדרות:

1. נניח ו P הינה קבוצה. *יחס שקילות מדומה* (pseudo equivalence relation) על P הוא היחס הבינארי ≡, כך שמתקיים עבור כל $X.Y \in P$:
   a. x≡x רפלקסיבי
   b. x≡y גורר y≡x

   יחס השקילות המדומה מחלק את P לתתי קבוצות שייתכן שיש ביניהן אף חיתוך. קבוצות אלו נקראות *מחלקות שקילות מדומות*.
2. גרף המכיל קבוצת איברים – V (שיקראו קודקודים). וקבוצה של קווים המחברים זוגות של קודקודים שיקראו צלעות. הגרף יקרא *שלם* אם יש צלע בין כל זוגות הקודקודים, כלומר לא קיים זוג שאין עבורו צלע תואמת. תת גרף שלם של גרף יקרא *קליקה*.

נניח כי F_k מייצג קבוצה של תתי קבוצות תדירות בגודל k. נגדיר גרף שייקרא k – association graph, גרף זה יסומן ב $G_k=(V,E)$. כאשר $V = \{X \mid X \in F_1\}$ ו

$$E = \{(X,Y) \mid X,Y \in V, and \exists Z \in F_{(k+1)} \; such..that..X,Y \subset Z\}$$

כלומר אנו דורשים כי הקודקודים יהיו שייכים לקבוצת תתי הקבוצות התדירות בגודל 1. ולגבי הקשתות – אנו דורשים כי יהיו שייכים לקבוצת תתי הקבוצות התדירות בגודל k+1.

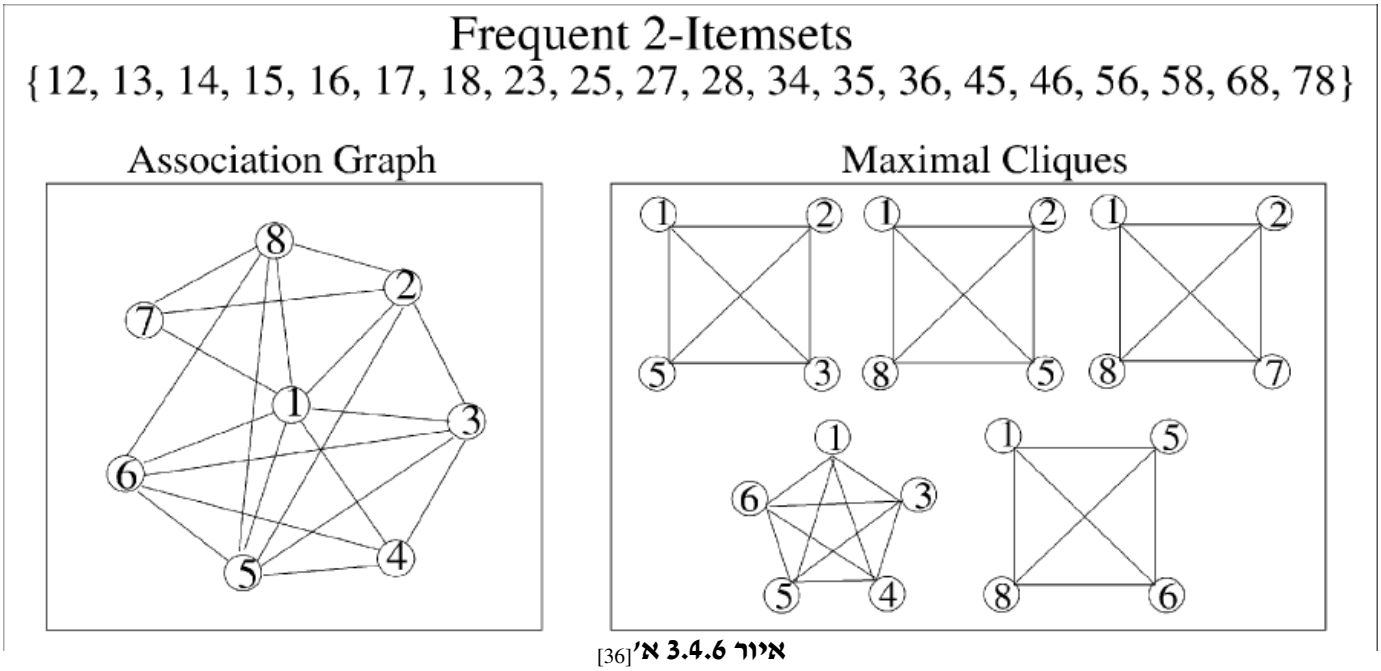
כעת נניח כי M_k מייצג את הקבוצה של הקליקות המקסימליות ב G_k.

---

[43] לדוג' עבור k אטומים נבצע $\binom{2}{k}$ חיתוכים עבור השלב הבא בעץ.

באיור 3.4.6 א' ניתן לראות את הגרף G_1 הקשתות מוגדרות ע"י הקבוצה F_2.

M_1={1235,1258,1287,1568}.



איור 3.4.6 א'[36]

ניתן לראות באיור 3.4.6 א' את כלל ההגדרות שהוגדרו לעיל:

הגרף הינו G_1 ולכן קבוצת הקודקודים שייכת ל F_1= {1,2,3,4,5,6,7,8}. הקשתות על פי ההגדרה שייכים ל F_2, ואכן ניתן לראות בבירור כי האיברים של F_2 מייצגים בצורה חח"ע את הקשתות.

לדוגמא: האיבר 17 – מיצג את הקשת בין קודקוד 1 ל 7.

הקבוצה M_1 מייצגת את קבוצת הקליקות המקסימליות – כלומר אם נסתכל על G_1 נוכל "לגזור" מתוכו כמה קליקות, בקבוצה M_1 ניתן לראות את הקליקות המקסימליות שהוצאנו מתוך G_1.

כעת נגדיר יחס שקילות מדומה ϕ_k על הרשת P(I) כך:

$$\forall X,Y \in P(I), X \equiv_{\phi_k} Y \Leftrightarrow \exists C \in M_k \text{ כך ש: } X,Y \subseteq C..and, p(X,k) = p(Y,k)$$

כלומר, שני תתי קבוצות יהיו באותה קבוצת שקילות מדומה אם הן תתי קבוצה של אותה קליקה מקסימלית, והן חולקות תחילית שווה באורך k. נקרא ל ϕ_k*יחס שקילות מדומה מבוסס קליקה מקסימלית*.(נשתמש בהגדרה זו בהמשך).

3. כל מחלקה מדומה $[X]_{\phi_k}$ שנוצרה ע"י יחס השקילות המדומה ϕ_k הינה תת רשת של P(I).

הוכחה: כמו 3.4.4 משפט 2.

ניתן כעת לומר, כי כל מחלקה מדומה $[X]_{\phi_k}$ הינה רשת בוליאנית. התמיכה של כלל האיברים יכולה להתקבל ע"י שימוש ב 3.4.3 משפט 4 על האטומים. כמו כן נציין כי ניתן להשתמש בכל אחת מאסטרטגיות החיפוש שהובאו לעיל.

4. נניח כי $\aleph_k$ מייצגת קבוצה של מחלקות מדומות של יחס השקילות המדומה מבוסס הקליקה המקסימלית ϕ_k. כל מחלקה מדומה $[Y]_{\phi_k}$ שנוצרה ע"י יחס התחילית ϕ_k היא תת קבוצה של מחלקה אחרת $[X]_{\theta_k}$ שנוצרה ע"י θ_k. ולהיפך, כל $[X]_{\theta_k}$ הוא איחוד של סט של מחלקות מדומות Ψ. מחלקות אלו נתונות ע"י המשוואה הבאה: $[X]_{\theta_k} = \bigcup\{[Z]_{\phi_k} \mid Z \in \Psi \subseteq \aleph_k\}$.[44]

משפט זה טוען בעצם כי כל מחלקה מדומה של ϕ_k הינה קטנה יותר מ θ_k.

לכן, אם נשתמש ביחס ϕ_k במקום θ_k, נוכל לייצר תתי רשתות קטנות יותר. תתי רשתות אלו יצרכו פחות זיכרון ויוכלו להיות מעובדות בצורה מקבילית ומהירה יותר מהמקבילות להן ביחס θ_k.

באיור 3.4.6 ב' ניתן לראות את המחלקות (תתי הרשתות) שנוצרו ע"י θ_1 ו ϕ_1. ניתן בבירור לראות כי ϕ_1 מייצר מחלקות קטנות יותר.

Maximal-Clique-Based Classes	Prefix-Based Classes
[1] : 1235, 1258, 1278, 13456, 1568 [2] : 235, 258, 278 [3] : 3456 [4] : 456 [5] : 568 [6] : 68 [7] : 78	[1] : 12345678 [2] : 23578 [3] : 3456 [4] : 456 [5] : 568 [6] : 68 [7] : 78

איור 3.4.6 ב'[36]

לדוגמא: 12345678=[1] גדולה בהרבה מקבוצת הקליקה המקסימלית:

{1235,1258,13456,1568}=[1] (כמובן שאנו מסתכלים על גודל של כ"א מתתי הקבוצות הנ"ל).

קיומן של המחלקות הקטנות משתלם וזאת מכיוון שחישובן של קליקות מקסימליות עבור גרפים רגילים הינה בעיית NP- Complete. למרות זאת, לרוב הגרף הינו דליל והקליקות המקסימליות יכולות להיות מחושבות ביעילות ובקלות. נשתמש אם כן ב ϕ_k רק במידה ו G_k איננו צפוף מידי.

חלק מהגורמים המשפיעים על צפיפות הצלעות כוללים תמיכה נמוכה וגודל תנועה גדול[45].

[44] הוכחה מלאה ניתן לראות ב [36]. ההוכחה חורגת ממסגרת העבודה ולכן לא תובא כאן.

[45] סקירה מקיפה של השפעת גורמים אלו על תוצאות האלגוריתם תיסקר בהמשך.

3.4.7 יצירת קליקה מקסימלית

עבור מחלקה [x] נאמר כי $y \in x$ מכסה תת קבוצה של [x], נסמן $\text{cov}(y) = [y] \cap [x]$.

עבור כל מחלקה C, נגדיר ראשית את קבוצת הכיסוי:

$$\{y \in C \mid \text{cov}(y) \neq 0.and.\text{cov}(y) \not\subseteq \text{cov}(z)\}$$

עבור כל $z \in C, z < y$.

לדוג': המחלקה [1] מאיור 3.4.6 ב'. מכיוון ש 2 הינו איבר ששיך ל [1]. נקבל:

$$\text{cov}(2) = [2] \cap [1] = \{2,3,5,7,8\} \cap \{1,2,3,4,5,6,7,8\} = \{2,3,5,7,8\} = [2].$$

בדוגמא שהובאה לעיל ניתן לומר כי cov(y) = |y|. עבור כל $y \in [1]$

בצורה דומה נקבל כי: $\text{cov}(3) = \{4,5,6\}$ ו $\text{cov}(4) = \{5,6\}$.

קבוצת הכיסוי של [1] תהיה $\{2,3,5\}$. בעצם ניתן לומר כי קבוצת הכיסוי הינה קבוצה של כלל האיברים שה cov שלהם יכול "לבנות" את קבוצת האב. ולכן נוכל לראות כי cov(2) ו cov(3) ו cov(5) מכילים בעצם את כלל האיברים של [1] . כמו כן עלינו לוודא כי כל אחת מהקבוצות המצויות בקבוצת הכיסוי אינן מכילות אחת את השניה. לכן קבוצת הכיסוי מורכבת דווקא מ: 2,3,5. מכיוון שקבוצה [4] מוכלת בקבוצה [3]. כמו כן קבוצה [6] מוכלת בקבוצה [5]. וקבוצה [7] מוכלת ב [2].

במהלך תהליך היצירה של קליקות מקסימליות מתחשבים אך ורק באיברים המוכלים בקבוצות הכיסוי השונות הנסקרות במהלך האלגוריתם. באיור 3.4.7 א'[46] מובא האלגוריתם עצמו:

```
1:for (i = N; i >= 1; i − −) do
2:    [i].CliqList = ∅;
3:    for all x ∈ [i].CoveringSet do
4:        for all cliq ∈ [x].CliqList do
5:            M = cliq ∩ [i];
6:            if M ≠ ∅ then
7:                insert ({i} ∪ M) in [i].CliqList such that
8:                ∄X or Y ∈ [i].CliqList, X ⊆ Y, or Y ⊆ X;
```

איור 3.4.7 א'[36]

ניתן לראות כי במהלך האלגוריתם, מחוללים בצורה רקורסיבית את הקליקות המקסימליות עבור איברים בקבוצות הכיסוי של כל מחלקה.

[46] לא תתבצע כאן סקירה מעמיקה של האלגוריתם הנ"ל.

<u>קליקה מקסימלית חלשה</u>

עבור חלק גדול מבסיסי הנתונים ,צפיפות הקשתות עדיין תהיה גבוהה ולא מספקת. במקרה זה קיימות הרבה קליקות עם חפיפות ביניהן.במצב שכזה לא רק שהכרייה איטית בהרבה, לעיתים אף עלולים לקבל בכל תת רשת תתי קבוצות תדירות משותפות, כך בעצם איבדנו את כל היתרון של השימוש בחלוקה לתתי רשתות.

בכדי לפתור את הבעיה נציג את שיטת *קליקה מקסימלית חלשה*[36] :

בהינתן שתי קליקות X ו Y. ניתן להגיד שקיים ביניהם קשר מסוג α אם : $\frac{|X \cap Y|}{|X \cup Y|} \geq \alpha$ כלומר, היחס של האיברים המשותפים בשניהם לעומת כלל האיברים גדול מהסף α שהוגדר מראש.

קליקה רפויה מקסימילית : $Z = \{X \cup Y\}$ תיווצר ע"י כיווץ של שתי הקליקות לקליקה אחת.

במהלך חילול הקליקה המקסימלית נתייחס אך ורק לקליקות בעלות ערך סף α הגדול מהערך שנקבע מראש. נציין כי עבור $\alpha = 1$ נחזור למצב חיפוש של קליקה מקסימלית הרגיל שהוצג לעיל. ובמצב של $\alpha = 0$ נקבל אך ורק קליקה אחת. מחקרים מראים שניתן להימנע מהתופעה שהנ"ל ע"י שימוש בסף של 0.5.

3.4.8 הצגת אלגוריתמי הכרייה

בחלק זה נציג את האלגוריתמים לכריית חוקי הקשר. האלגוריתמים הנ"ל ישתמשו בוריאציות שונות של כלל השיטות שהוצגו לעיל. כמובן שאנו נתמקד ב Eclat [5][36]

- בשלב הראשון יחושבו תתי הקבוצות התדירות בגודל 1 ובגודל 2.[47]

בשלב הבא יחושבו תתי הרשתות (בשימוש באחת מהשיטות שהוצגו לעיל: (תחיליות – Prefix Based, קליקה - Clique Based). הרשתות יווצרו מתוך קבוצת תתי הקבוצות התדירות בגודל 2.

- כל תת רשת תעובד באופן נפרד, אחת בכל פעם בשימוש אחד משלושת סוגי החיפוש שהוצגו (Bottom Up, Top Down, Hybrid).

ניתן לראות סיכום כלל המאפיינים של השיטות שהוצגו במאמר בטבלה 3.4.8 א'

שיטת חלוקה / שיטת חיפוש	Prefix Based	Clique Based
Bottom Up	***Eclat***	Clique
Top Down		Top Down
Hybrid Search	Max Eclat	Max Clique

טבלה 3.4.8 א'

בטבלה 3.4.8 א' אנו רואים שכלל האלגוריתמים המוצגים במאמר הינם בעצם וריאציות שונות של שיטות החלוקה והחיפוש השונות שהוצגו במאמר. כל שורה מייצגת שיטת חיפוש בעץ ואילו העמודות מייצגות שתי שיטות חלוקה שונות של העץ.

כפי שצוין לעיל אנו נתמקד ב Eclat. שיטה זו עושה שימוש בחלוקה לתתי רשתות בשימוש בשיטת התחיליות. לאחר מכן נעשה שימוש ב Bottom Up בכדי לנתח כל תת רשת בנפרד.

בכדי להציג את הדברים בצורה ברורה יותר נרחיב מעט לגבי אלגוריתם החיפוש bottom up שעושים בו שימוש בשיטת Eclat.

באיור 3.4.8 ב' ניתן לראות את הפסודו – קוד לחיפוש bottom up שאנו מבצעים ברשת.

[47] ב [36], מוצגת שיטה לגילוי מהיר של תתי קבוצות תדירות בגודל 1 ו 2. לא נתמקד בה כאן.

```
Bottom-Up(S):
for all atoms A_i ∈ S do
    T_i = ∅;
    for all atoms A_j ∈ S, with j > i do
        R = A_i ∪ A_j;
        L(R) = L(A_i) ∩ L(A_j);
        if σ(R) ≥ min_sup then
            T_i = T_i ∪ {R}; F_|R| = F_|R| ∪ {R};
    end
end
for all T_i ≠ ∅ do Bottom-Up(T_i);
```

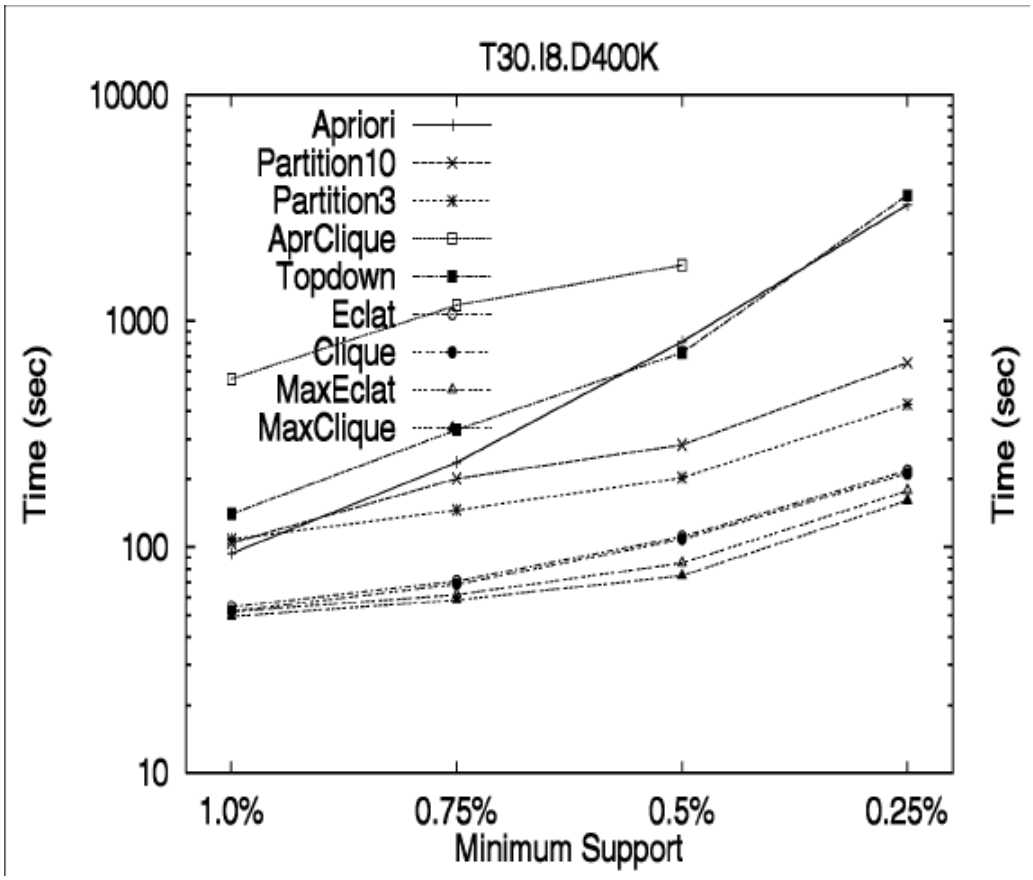
איור 3.4.8 ב'[36]

הקלט לפונקציה יהיה רשימה של אטומים של תת רשת כלשהיא- S. כפי שכבר צוין מספר פעמים במקרה זה התמיכה המשותפת של זוג אטומים מחושבת ע"י חיתוך של רשימות ה tid של כל אחד מהם. כל זוג שנמצא תדיר יעבור בצורה רקורסיבית לשלב הבא של הפונקציה. התהליך יימשך עד שיימצאו כל תתי הקבוצות התדירות. מבחינת סיבוכיות זיכרון נדרש בכל שלב זיכרון רק לשתי רמות בעץ. זו שמעובדת וזו שאחריה. נוכל לומר בעצם כי סיבוכיות הזכרון במקרה הזה הינה קבועה: $O(1)$.

באיור 3.4.8 ג' ניתן לראות תוצאות של הרצה השוואתית של כלל האלגוריתמים השונים שנסקרו כולל אפריורי. באיור מוצגים זמני הריצה של האלגוריתמים השונים, כאשר רמת התמיכה המינימלית יורדת. ניתן לראות כי ככל שתמיכה המינימלית (ציר ה X) יורדת, זמן הריצה של אלגוריתם אפריורי עולה בצורה משמעותית, וזאת מכיוון שכמות תתי הקבוצות התדירות עלתה ולכן גם מספר האיברים שיש לסרוק.

ניתן לראות בצורה ברורה כי Eclat עוקף בביצועיו את מרבית האלגוריתמים שנסקרו[48] כולל Apriori בסדרי גודל. וזאת מכיוון ש Eclat עובר רק מספר מועט של פעמים על בסיס הנתונים. כמו כן, הוא אינו דורש שימוש בטבלאות גיבוב ומשתמש בפעילות חיתוך פשוטות. יתירה מכך, Eclat מסוגל להתמודד עם רמת תמיכה נמוכה בצורה טובה יותר מ Apriori ודומיו.

[48] המאמר משתמש לצורכי השוואה גם באלגוריתם partition. אלגוריתם זה מהווה שיפור לאלגוריתם אפריורי ע"י שימוש בשיטת הפרד ומשול. אלגוריתם זה לא יסקר במסגרת עבודה זו. ניתן לקרוא עליו ב [36] סקירה נוספת על אלגוריתם זה ניתן לראות ב [39] – 2.4.3.



איור 3.4.8 ג'[36]

אם מתחשבים בארבעת האלגוריתמים העיקריים שנותרו בבדיקה, ניתן לראות כי Max Clique הינו בעל הביצועים הטובים ביותר מהסיבות הבאות:

- עקב השיפור בשיטת החלוקה, שגורם ליצירת מחלקות קטנות יותר. Clique מהיר קצת יותר מ Eclat. כמו כן, מספרי החיתוכים ברשימות tid ש Clique מבצע הינם נמוכים יותר.
- Max Clique מהיר יותר מ Max Eclat, שוב בגלל יצירה של מחלקות קטנות יותר. עובדה שמקצרת את זמן החיפוש בכל מחלקה. נציין כי כאשר יש תמיכה נמוכה הביצועים של שיטות מבוססות Clique יורדים בצורה ניכרת עקב צפיפות וחפיפות בקליקות.
- היתרונות של החיפוש ההיברידי[49] מעניקות ל Max Qlique את הבכורה בצורה כוללת על פני שאר שיטת הכריה.

ניתן לראות אם כן, בצורה ברורה את יתרונות השיטה של החלוקה למחלקות שקילות שבה משתמשים Eclat והאלגוריתמים הדומים לו על פני השיטה של אפריורי.

[49] החיפוש ההיברידי מזהה מראש itemsets ארוכים ונמנע מלזהות את כל ה Itemsets. מסיבה זו עבור itemsets גדולים, השיטה ההיברידית הינה היחידה שמסוגלת לתפקד.

3.4.9 שיפורים והרחבות ב Eclat

3.4.9.1 מיקבול האלגוריתם – שימוש בעץ מושרש (radix tree)

בחלק זה נציג אלגוריתם המממש את Eclat בצורה מקבילית. המימוש המקבילי מתאפשר בעיקר בשל השימוש במבנה הנתונים המיוחד ולכן נתמקד בו בעיקר. שינויים נוספים והרחבות בנושא מאפייני האלגוריתם ניתן לראות ב [29]

נציג כעת בצורה פורמלית כוללת[50] את מבנה Eclat [7]

The Eclat algorithm

Initialization phase:

- Scan local database partition
- Compute local counts for all 2-itemsets
- Construct global L2[51] count

Transformation phase:

- Partition L2 into equivalence classes
- Schedule L2 over the set of processors P
- Transform local database into vertical form
- Transmit relevant tid-lists to other processors

Asynchronous phase:

- for each equivalence class E2 in local L2

  ComputeFrequent(E2)

Final Reduction phase:

- Aggregate Results and Output Associations

[50] מכיוון ש[7] עוסק במיקבול האלגוריתם הנ"ל, הסכמה הכללית של Eclat מוצגת בצורה מקבילית.

[51] L2 הינה רשת הזוגות, כלומר הרמה השניה ברשת הכוללת של כלל האטומים.

ניתן לראות כי האלגוריתם כולל בתוכו מספר שלבים:

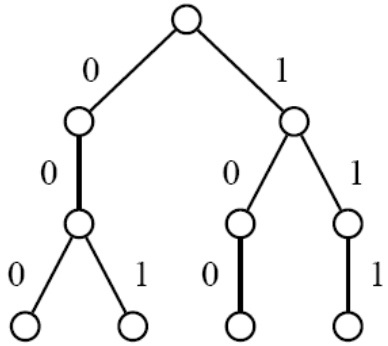
בשלב הראשון אנו מחשבים את תתי הקבוצות התדירות בגודל 2. לאחר מכן מתבצעת החלוקה למחלקות שקילות. בכל מחלקה מתבצע כרייה בצורה נפרדת. ולבסוף מאחדים את התוצאות.

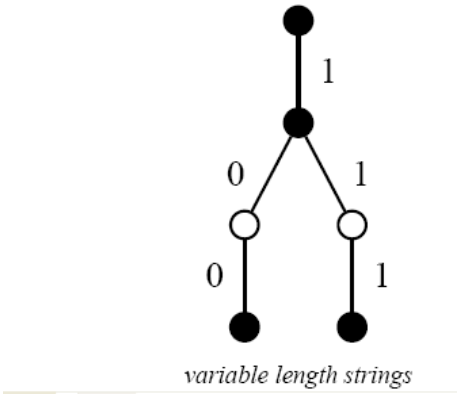
ניתן לראות בצורה ברורה כי שלב ההמרות (transformation phase) הינו המסיבי והמאט את זמני הריצה(גם בגרסא המקבילית). בכדי לשפר את הביצועים בגרסא המקבילית של האלגוריתם נעשה שימוש [7] בעצים מושרשים.

במקרה שלנו נאחסן בצמתים את הספרות הבינאריות 0 ו 1 . שייצגו מספר[52] המייצג אינדקס של תנועה בבסיס הנתונים. קיימות שתי שיטות לייצוג מילים בעץ:

- ייצוג מחרוזות בארוך משתנה .
- ייצוג מחרוזות קבועות

דוגמא ניתן לראות באיור 3.4.9 א'





איור 3.4.9 א' [7]

נסביר את השיטה - עבור העץ בגודל משתנה (התחתון). צמתים לבנים מסמנים כי לא מצויה שום מילה בצומת הנ"ל. צמתים שחורים – ההפך.

לדוגמא: אם אנו מחפשים את המילה 10 נבצע את הצעדים הבאים:

- נתחיל מהשורש, נרד למטה (1)
- נפנה שמאלה(0)
- נבדוק את צבע הצומת

[52] 1 יסמן ימינה ו 0 שמאלה

- אם מדובר בצבע לבן – המילה לא קיימת בעץ
- אם מדובר בצבע שחור המילה קיימת

כאשר מדובר בעץ בעל מחרוזות באורך קבוע, לא נצטרך להגדיר צבעים עבור הצמתים. כיוון שסוף מילה מסומן ע״י עלה.

נראה כעת כיצד ניתן לייצג בסיס נתונים בשימוש עצים מהסוג הנ״ל:

בהינתן לנו בסיס הנתונים הבא:

Client Id	Contract Date	Max Amount	Seller	Kind of Cont.	Min Ref.	Acc. Id
1	12-21-1992	450,000	2	House	900	1
2	02-24-2000	12,000	17	Car	830	2
3	11-28-1996	230,000	11	House	1,350	3
4	05-30-2001	780,000	2	House	2,400	4
1	07-17-1992	27,500	3	Car	912	5
1	04-13-1998	1,000,000	2	Family	100	6
2	09-11-1999	830,000	2	House	11,000	7

איור 3.4.9 ב׳ [7]

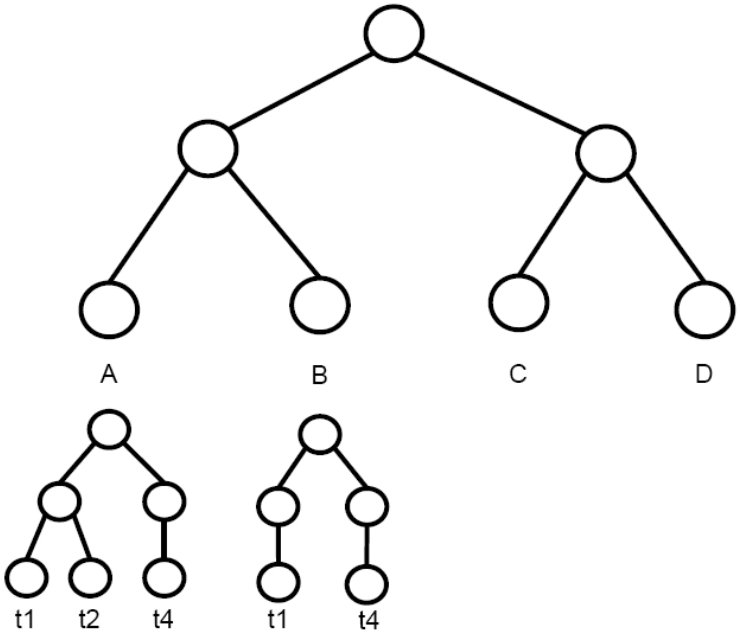
בסיס הנתונים מכיל מידע המחולק לפי עמודות.

עבור כל אחת מהעמודות בטבלה נבנה מילון ערכים לעמודה ועבור כל מילה מהמילון הנ״ל ניצור קבוצה המכילה את האינדקסים של התנועות שהמילה מופיעה בהן.

לדוגמא:

עבור העמודה Kind of Contract – רשימת הערכים האפשריים לשדה תהיה: {House, Car, Family}. וקבוצות האינדקסים יהיו: House = {1,3,4,7}, Car = {2,5}, Family = {6}. במידה ונרצה להריץ שאילתא על בסיס הנתונים הכוללת מספר תנאים לערכי שדות. יהיה עלינו לבצע חיתוך של קבוצות האינדקסים והמילונים הרלוונטיים.

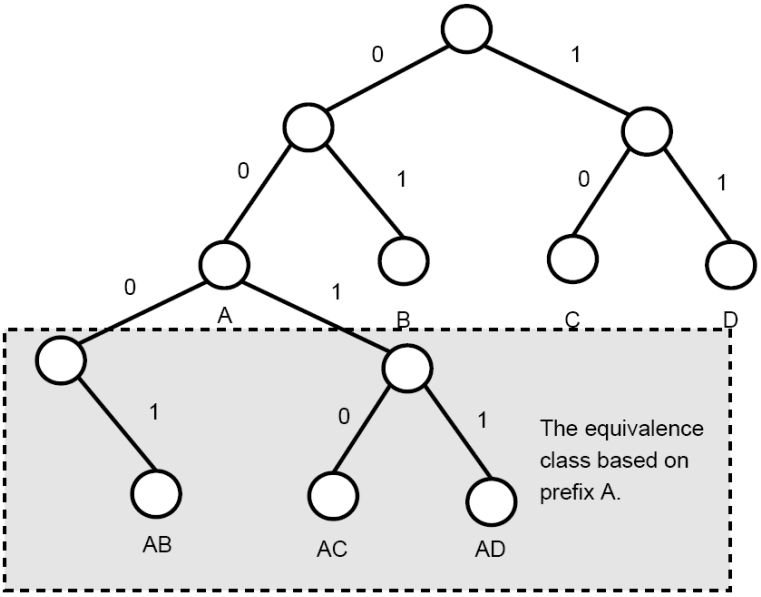
בכריית חוקי הקשר נשתמש בעצים אלו בצורה הבאה:



איור 3.4.9 ג' [7]

באיור 3.4.9 ג' יצרנו עץ חיפוש בינארי רגיל עבור ארבעת העצמים מבסיס הנתונים {A,B,C,D}. לכל עלה המייצג איבר בבסיס הנתונים נחבר Radix Tree. העץ המצורף יכיל את האינדקסים של התנועות בבסיס הנתונים המכילות את המידע הנ"ל. במצב כזה, כעת בכדי למצוא תמיכה יהיה עלינו לבצע חיתוכים בין ה Radix Tree של כ"א מהאיברים. החיתוכים יתנו לנו את מספר הפעמים שהאיבר מופיע. לדוג' אם מסלול אל העצם A הינו 00(כלומר שמאלה שמאלה בעץ הבינארי) והמסלול ל B הינו 01 (כלומר שמאלה וימינה בעץ) המסלול ל AB יהיה 0001[53] (3 פעמים שמאלה ואז ימינה). נציין כי, העלות החישובית של החיפוש נתונה ע"י פעולות חיתוך פשוטות בין עצים. יתירה על כך ניתן לאחסן גם חלק מנתוני התמיכה על הדיסק בכדי להשתמש בהם שוב בכדי לחולל מועמדים נצטרך לבצע פעולת צירוף (join) בין החברים של אותה מחלקת שקילות. אם נשתמש בשיטת הקידוד שהוסברה לעיל באיור 3.4.9 ג', ניתן יהיה לראות כי כל האיברים של אותה מחלקת שקילות מצויים בתוך תת עץ של קבוצת הנתונים המגדירה את מחלקת השקילות. קבוצת הנתונים המגדירה את מחלקת השקילות הינה התחילית של כלל האיברים במחלקה הנ"ל. בעץ שכזה, בו כלל האיברים של תת עץ הינם בעלי אותה תחילית, קבוצות שקילות יכולות להיות מיוצגות כתתי עצים. באיור 3.4.9 ד' ניתן לראות את הקשר בין מחלקות השקילות לתתי העצים:

[53] יודגם בהמשך

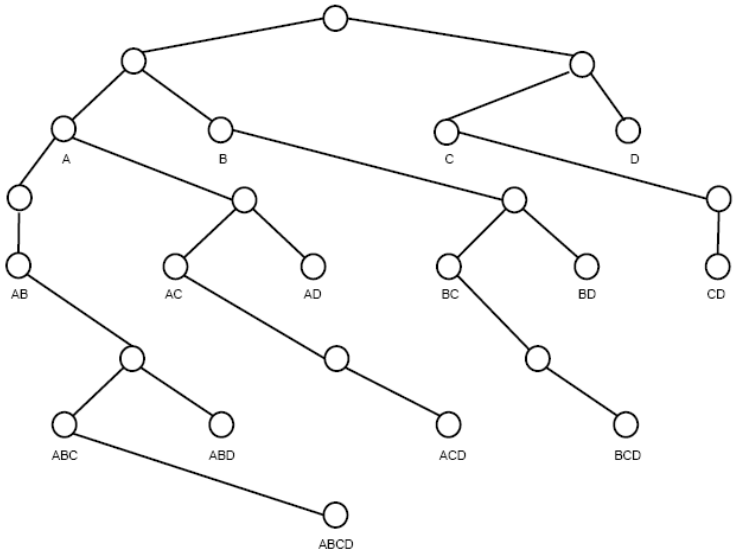


איור 3.4.9 ד' [7]

ניתן לראות בצורה ברורה באיור כי כלל תתי העצים של A הינם בעצם החברים במחלקת השקילות בעלת תחילית A.

כעת בכדי לייצר את רמת המועמדים הבאה בתור, יש לצרף את האיברים במחלקת השקילות. פעולת הצירוף תכלול הוספה של תת עץ מתחת לכל אחד מהעלים הקיימים, תת העץ יכיל את הצמתים שצורפו. ביצירה אנו ניצור רק בנים שמאלים בעץ.

לדוגמא: בכדי לקבל את ABC, נצרף את AB עם AC. ככל שנמשיך הלאה בתהליך יווצרו עוד ועוד צירופים התוצאה הסופית תהיה כמו באיור 3.4.9 ה'



איור 3.4.9 ה'[7]

בכדי למצוא את התמיכה עבור איבר מסוים, ראשית נוריד מן העץ איברים שאינם רלוונטיים לדוגמא: אם ידוע ש AB אינו תדיר, ניתן לומר כי גם ABC יהיה שכזה ולכן נוכל להוריד את כל תת העץ המושרש ב AB (בדומה לעיקרון אפריורי). מציאת התמיכה תתבצע ע"י חיתוכים של רשימות ה tid של איברים בבסיס הנתונים. ניתן יהיה ע"י הפצה מקבילית של חיתוכים מקומיים למצוא חיתוך של קבוצה המכילה אותם.

3.4.9.2 שיפורי ביצועים ב Eclat

ייצוג בזיכרון

בחלק זה נציג שיפורים שהוצגו ב [5] לאלגוריתם Eclat. השיפורים הם בהיבטים של סיבוכיות זמן ריצה וסיבוכיות מקום[54]. אופי המימוש של Eclat בחלק זה[5] יהיה שימוש ב Bit Matrices. ב Bit Matrix כל שורה במטריצה בטבלה מייצגת את העצמים והעמודות את התנועות בבסיס הנתונים. נדליק את הביט בתא הרלוונטי בכדי לציין כי העצם הנ"ל מופיע בתנועה המדוברת[55]. ניתן לייצג את טבלת הביטים בפועל בשתי דרכים:

- מטריצה דו מימדית – כלומר נניח והעמודות מייצגות את התנועות ובשורות מופיעים כלל העצמים בבסיס הנתונים. בכדי לציין שעצם מופיע בתנועה מסוימת נסמן 1 בכניסה המשותפת של שני בני הזוג.

[54] נתמקד בשיפורים אלגוריתמיים ומבניים עקרוניים ולא טכניים.

[55] יש לשים לב שמדובר ממש על שיטת Bit Table שנידונה בפירוט רב ב [39] שם דובר על מימוש השיטה בהקשר של אפריורי.

- ניתן לשייך לכל שורה (שמייצגת עצם) רשימה של העמודות שבהן הוא קיים. צורה זו דומה מאוד לשיטת הייצוג של tid המוכרת מ Eclat הרגיל.

<u>שיטת החיפוש</u>

כידוע Eclat מבצע חיפוש על עצי תחיליות. המעבר בין צומת לבנו הראשון תהיה כרוכה בבניה של מטריצת ביטים חדשה[56]. הבניה תתבצע ע״י חיתוך של השורה הראשונה עם כל השורות הבאות אחריה. אותו נוהל יבוצע גם עבור השורה השניה (חיתוך עם כל הבאות אחריה). בסופו של התהליך יגדלו כלל האיברים למצבם החדש, כלומר אם עוצמת כל קבוצת אברים בבסיס בעץ (ברמה המדוברת) היה 1, כעת הוא יהיה 2. כמובן שניתן יהיה לסנן בצורה דינאמית שורות המייצגות תתי קבוצות שאינן תדירות, מכיוון שהן לא יתרמו לנו יותר להמשך התהליך.
החיתוך של שתי השורות יתבצע ע״י שימוש ב and לוגי פשוט.

3.4.10 סיכום

לסיכום, נסקור את המאפיינים היחודיים של ארבעת האלגוריתמים שנסקרו מהאלגוריתמים הידועים והמוכרים מבוססי Apriori ודומיו.

- השימוש בחיתוך בין רשימות בכדי לחשב את תת הקבוצה התדירה המשותפת חוסך זמן חישוב רב.
- לא נעשה שימוש במבני נתונים מסובכים כגון : טבלאות גיבוב.
- האלגוריתמים מתאפיינים במספר סריקות מועט של בסיס הנתונים (בשונה מאפריורי).
- סכמת החיפוש ההיברידית (ראה סוף 3.4.8)

קיימות שיטות העושות שימוש דומה בעקרונות שלקוחים מ FP ו Eclat נציין בינהם את Transaction Mapping : בשיטה זו משתמשים בייצוג בסיס נתונים אנכי, ובעץ תחיליות לקסיקוגרפי בכדי לחולל את המועמדים לתתי קבוצות תדירות. [30]

[56] נצרכת מטריצה חדשה מכיוון שכעת האיברים משתנים, לדוג׳ במקום A כעת האיבר בשורה יהיה AB.

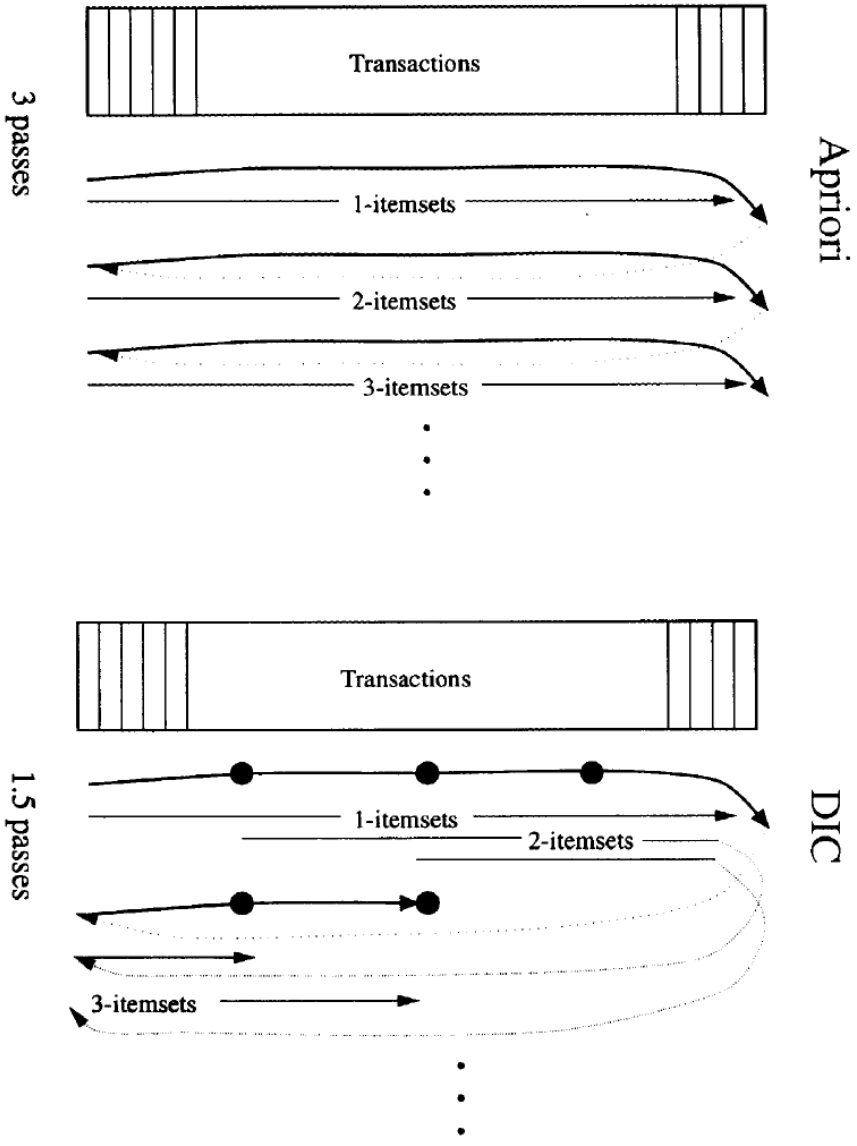
3.5 DIC מנייה דינמית של תתי קבוצות (*Dynamic Itemset Counting*)

3.5.1 תיאור האלגוריתם

DIC פורסם [6] בשנת 1997 בתור אלגוריתם חדש ומשופר לעומת אלגוריתם אפריורי. אלגוריתם זה מקטין במידה משמעותית את מספר הפעמים שעוברים על בסיס הנתונים[10]. העיקרון העומד בבסיס האלגוריתם דומה ל"רכבת" החולפת על פני בסיס הנתונים, עם עצירות במרווחים של כל M תנועות (M הינו קבוע הנקבע בצורה ניסיונית). כאשר "הרכבת" מגיעה לסוף בסיס הנתונים נוכל לומר כי בוצע מעבר אחד שלם על בסיס הנתונים. לאחר מכן, נתחיל שוב מהתחלה את המעבר על בסיס הנתונים. "הנוסעים" ברכבת יהיו ה itemsets , כאשר יש itemset "על הרכבת" נספור את המופע שלו בתנועות שנקראו.

אם נשווה את DIC לאפריורי נוכל לומר כי באפריורי עבור תתי קבוצות תדירות בגודל 1 בוצע מעבר 1 על בסיס הנתונים. עבור גודל 2, בוצע עוד מעבר וכן הלאה. ב DIC קיימת הגמישות של "העלאה לרכבת" של כל אחד בכל "עצירה" בתנאי שאותו "נוסע" ירד באותה תחנה שעלה בה בסיבוב הבא. בצורה כזו ה itemset "ראה" את כל בסיס הנתונים. בעצם ניתן להתחיל לספור מופעים של Itemset (בכדי לבדוק האם הוא תדיר), החל מהרגע שבו אנו חושדים ש Itemset הינו תדיר נתחיל לספור, ללא צורך לחכות לסוף בסיס סריקת בסיס הנתונים (מכיוון שהסריקה הינה מעגלית – החל מנקודה שבה itemset "עולה לרכבת"). כלומר בשונה מאפריורי שמתחיל לספור תתי קבוצות תדירות בגודל n רק החל מהמעבר ה n על בסיס הנתונים הרי שכאן כבר במעבר הראשון אנו מחפשים תתי קבוצות תדירות בגודל 1, 2 ואף יותר מכך (תלוי ב M) יוצא שאנו מנצלים את כל מעבר על בסיס הנתונים לחיפוש יותר סוגים של תתי קבוצות לעומת אפריורי.
לדוגמא:

אם אנו מבצעים כרייה של 40,000 עצמים בבסיס הנתונים ונתון ש M = 10,000.
אנו נספור את כלל תתי הקבוצות התדירות בגודל 1 במעבר הראשון על כלל בסיס הנתונים.
למרות זאת, נוכל להתחיל לספור תתי קבוצות תדירות בגודל 2 לאחר העצירה הראשונה (10,000 תנועות). ניתן יהיה להתחיל לספור תתי קבוצות תדירות בגודל 3 כבר לאחר העצירה השנייה (20,000 תנועות). לצורך הדוגמא נניח כי לא קיימות תתי קבוצות תדירות בגודל 4 שאותן יש לספור). ברגע שנגיע לסוף הקובץ, נפסיק לספור את תתי הקבוצות התדירות בגודל 1. נחזור להתחלה ונמשיך בספירה של קבוצות בגודל 2 ו 3. לאחר 10,000 התנועות הראשונות נפסיק לספור גם תתי קבוצות תדירות בגודל 2, ולאחר ה 20,000 הראשונות נפסיק לספור קבוצות בגודל 3. בסופו של דבר ביצענו 1.5 מעברים על בסיס הנתונים במקום 3 מעברים שאפריורי היה מבצע.
ניתן לראות את ההבדל באיור 3.5.1 א'. רואים בבירור באיור כי DIC מבצע פחות מעברים מאפריורי.



איור 3.5.1 א' [6] – תהליך הכריה

DIC , בדומה לאפריורי, משתמש בתכונת אפריורי – אם תת קבוצה מינימלית אינה תדירה גם הקבוצות המכילות אותה לא יהיו תדירות.

DIC יסווג את תתי הקבוצות שיימצאו במהלך החיפוש לארבעה סוגים שונים:

- תתי קבוצות תדירות ממש (התמיכה מעל התמיכה המינימלית) – יסומנו בריבוע
- תתי קבוצות שאינן תדירות – בוודאות מתחת לתמיכה המינימלית – יסומנו בעיגול
- תתי קבוצות החשודות כקבוצות תדירות – תתי קבוצות שספירתן לא הסתיימה עדיין אך התמיכה שלהן גדולה יותר מהתמיכה המינימלית = יסומנו בריבוע מקווקו.
- תתי קבוצות החשודות כאינן תדירות – עדיין בתהליך הספירה, יסומנו בעיגול מקווקו.

שלבי האלגוריתם [6] :

1. The empty itemset is marked with a solid box. All the l-itemsets are marked with dashed circles. All other itemsets are unmarked.

תת הקבוצה הריקה תסומן, כל שאר תתי הקבוצות בגודל 1 יסומנו כתת קבוצה – 4 (עיגול מקווקו – בתהליך הספירה).

2. Read M transactions. We experimented with values of M ranging from 100 to 10,000. For each transaction, Increment the respective counters for the itemsets Marked with dashes.

נקרא M תנועות מבסיס הנתונים, עבור כל תנועה (במהלך תהליך הקריאה) נעדכן מונים פנימיים לאלגוריתם (יוסבר בהמשך) העדכון יבוצע אך ורק עבור עצמים מקווקווים

3. If a dashed circle has a count that exceeds the support threshold, turn it into a dashed square. If any immediate superset of it has all of its subsets as solid or dashed squares, add a new counter for it and make it a dashed circle.

במידה ועברנו עבור תת קבוצה מסוימת (שמסומנת בעיגול מקווקו) את סף התמיכה המינימאלית (ואנו כאמור, עדיין בתהליך הספירה) נסמן את תת הקבוצה הנ״ל בריבוע מקווקו.

אם כעת כל בניה של הקבוצה המכילה את תת הקבוצה הנ״ל (super set), הינם ריבועים נסמן את הקבוצה הזו (קבוצת האב הישיר - super set)בתור עיגול מקווקו[57].

4. If a dashed itemset has been counted through all the transactions, make it solid and stop counting it.

אם תת קבוצה מקווקת נספרה ועברה את כלל התנועות הפוך אותה לקבועה (לא מקווקות – כך לא תדיר ישאר לא תדיר ותדיר ישאר תדיר).

5. If we are at the end of the transaction file, rewind to the beginning.

חזור להתחלה במקרה סיום

6. If any dashed itemsets remain, go to step 2.

באם נשארו עצמים לספור חזור לשלב 2 .

[57] סימון העיגול מציין שכעת ניתן להתחיל בספירת המופעים של האב מכיוון שספירת מופעי הבנים הסתיימה.

3.5.2 מבני נתונים

המימוש של DIC מצריך מבנה הנתונים מיוחד המסוגל לבצע מעקב אחר תתי קבוצות רבות במקביל ולבצע את הפעולות העיקריות הבאות:

- הוספת itemsets חדשים – יצירת תתי קבוצות מתוך התנועות הנתונות.
- תחזוק מונה עבור כל תת קבוצה (itemset). כאשר תנועה נקראת מתוך בסיס הנתונים המונים של תתי הקבוצות שקיימות בתנועה יקודמו. נציין כי פעולה זו הינה צוואר הבקבוק של האלגוריתם ונדון בשיפורה בהמשך.
- ניהול המצבים של תתי הקבוצות – ע"י מעבר ממקווקו לקו רגיל ומתדיר ללא תדיר (מעיגול לריבוע) במקרה הצורך.
- כאשר תתי קבוצות הופכות לתדירות יש לדעת אילו תתי קבוצות אחרות ניתן כעת לסמן בתור מקווקוות ולהתחיל לספור אותן, מכיוון שיש להן פוטנציאל להיות תדירות

מבנה הנתונים שישמש למטרה זו יהיה עץ גיבוב[58]. נשתמש במספר כללים:

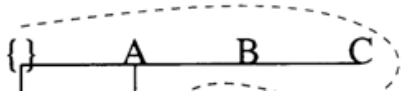
- כל קבוצת נתונים תסודר על פי העצמים שהיא מכילה(יוסבר בהמשך).
- לכל קבוצת נתונים שאנו סופרים / שנספרה קיים צומת המשויך אליה. כך גם כלל התחיליות המשתייכות אל קבוצת הנתונים הנ"ל
- השורש הינו תת הקבוצה הריקה
- כלל תתי הקבוצות בגודל 1 הינם בנים של השורש. ההסתעפויות מצמתים אלו יסומנו על ידי העצם שאותו הם מייצגים.
- שאר תתי הקבוצות יוצמדו לתחילית שלהם (התחילית תחיל הכל חוץ מאיבר האחרון של הקבוצה). תתי הקבוצות הנ"ל יסומנו על ידי האיבר האחרון בכל תת קבוצה.

ניתן לראות באיור 3.5.2 ב' דוגמא לעץ מהסוג הנ"ל.

נציין כי באיור מודגשת הפעולה שמתבצעת בזמן שהתנועה ABC נקראת מתוך בסיס הנתונים.

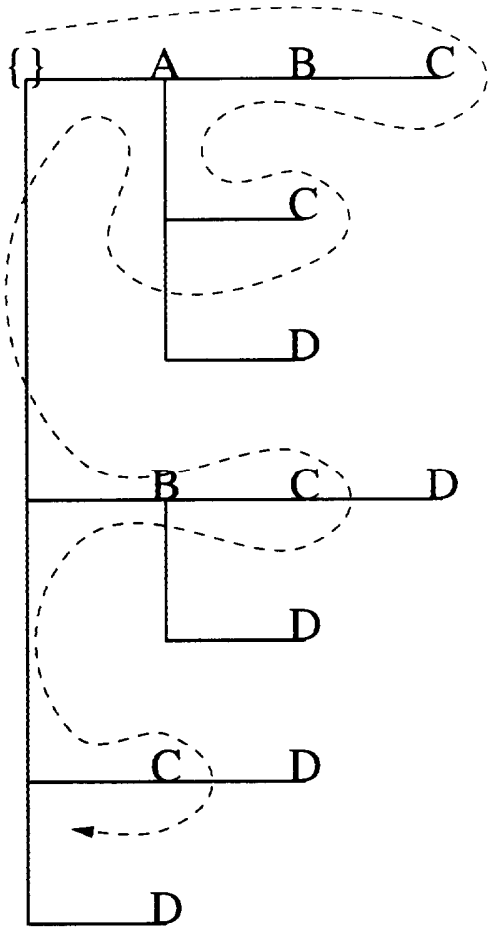
במצב כזה יש לקדם את המונים של: A,AB,ABC,AC,B,BC,C. כפי שהוסבר כל צומת מאחסנת את האיבר האחרון של תת קבוצת הנתונים (itemset) שהיא מייצגת.

לדוגמא: באיור 3.5.2 א' מוצג חלק מתוך עץ הגיבוב. C מייצג כאן את האיבר האחרון של תת הקבוצה ABC.



איור 3.5.2 א' [6]

[58] הרחבה על עץ גיבוב חורגת ממסגרת עבודה זו, בקצרה ניתן לומר כי עץ גיבוב הינו עץ של טבלאות גיבוב.



איור 3.5.2 ב' [6]

3.5.3 הסדר הפנימי של העצמים

בהינתן קבוצת תתי קבוצות, הצורה של מבנה הנתונים שמייצג אותם תלויה בהכרח בסדר הפנימי באיור 3.5.2 ב׳ סדר העצמים בעץ היה A B C D, עובדה זו משפיעה על מספר הפעמים ש A ו D מופיעים בעץ (1 לעומת 5). בכדי להגדיר כיצד ניתן יהיה לבצע אופטימזציות על הסדר הפנימי של העצמים יהיה עלינו לרדת לעומקו של תהליך קידום המונים הפנימיים בתוך מבנה הנתונים.

נניח ונתונה תנועה S עם עצמים S[n]….S[0], הנתונים בסדר מסוים. בכדי להגדיל את המונים נבצע את האלגוריתם הבא [6] :

```
lncrement(T,S)
 {
/“ Increment this node’s counter “/
T.counter++;
If T is not a leaf then for all i, 0 ≤ i ≤ n:
/“ Increment branches as necessary */
If T.branches[S[i]] exists:
        Then Increment(T.branches[S[i]], S[i+l..n])
Return.
}
```

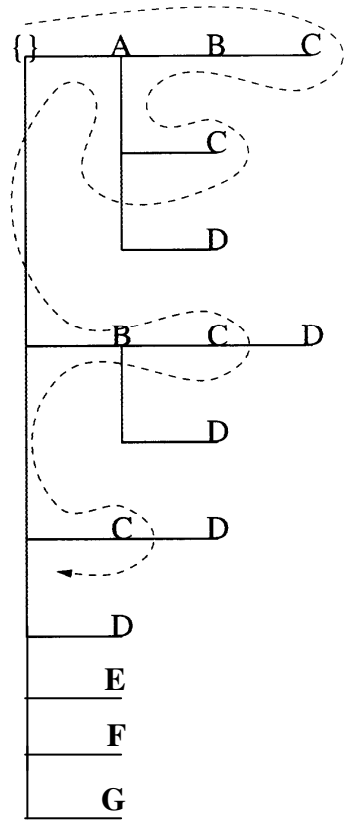
בעיקרו של האלגוריתם עומד הצורך לעדכן את המונים בעץ בצורה רקורסיבית.

כעת נראה את העלות החישובית של הפונקציה הנ״ל.

$$\sum_{I} n - Index(Last(I), S)$$

במקרה הנ״ל התחום של I יהיה כלל תתי הקבוצות שאינן עלים ב T ומצויים ב S.האיבר הפנימי בנוסחא : n – index(Last(I),S)) יסמן את מספר האיברים שנותרו ב S לאחר האיבר האחרון של I. איברים אלו ייבדקו בלולאה הפנימית של האלגוריתם. לכן יש יתרון למקם את העצמים שמופיעים בתתי קבוצות רבות להיות אחרונים בסדר המיון של העצמים, כך שמעט איברים ישארו אחריו, וכך נעבור בלולאה הפנימית שוב רק על מעט איברים, האיברים שמופיעים במעט תתי קבוצות יופיעו ראשונים.

לדוגמא : אם נביט באיור 3.5.2 ב׳, וננית כי קיימים איברים נוספים G,F,E. מבנה הנתונים כעת יראה כך :



איור 3.5.3 א' [6]

נניח וכעת אנו מכניסים לעץ את ABCDEFG, נראה מה הסיבוכיות של Increment במקרה הנ"ל. הקריאה לפונקציה תראה כך:

Increment({},ABCDEFG)

מכיוון שבמצב זה של הפונקציה, אורך המילה להוספה הינו 7. הלולאה הפנימית תתבצע 7 פעמים. כלומר יתווספו הקריאות:

Increment(A, BCDEFG), Increment(B,CDEFG), Increment(C,DEFG),

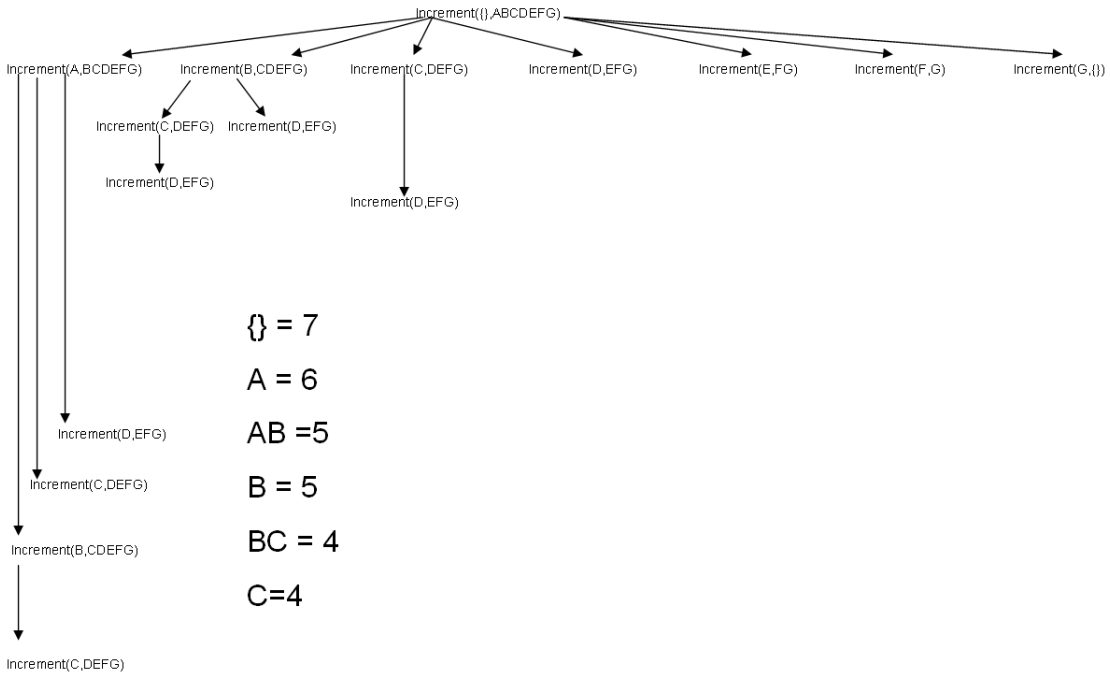
Increment(D,EFG), Increment(E,FG), Increment(F,G), Increment(G,{})

עבור Increment(A, BCDEFG) הלולאה תבוצע 6 פעמים ויתווספו הקריאות הבאות (עבור הבנים של A – B,C,D

Increment(B,CDEFG), Increment(C,DEFG), Increment(D,EFG)

וכך הלאה בצורה רקורסיבית.

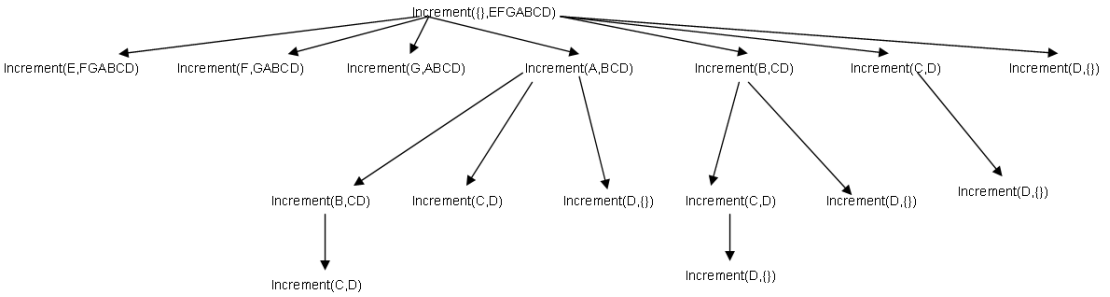
עץ מסכם מוצג להלן באיור 3.5.3 ב׳:



איור 3.5.3 ב׳ – מפת הקריאות לפונקציה

נציין כי למרות מפת הקריאות, מספר מעברי הלולאה באלגוריתם מתקבל ע״י אורך המילה בפרמטר השמאלי. במצב זה תתקבל תוצאה של 31(סוכמים רק את אורכי המילים של קריאה שאיננה עלה – 4+4+5+5+6+7=31).

במידה ונשנה את סדר האותיות במילה ל: EFGABCD העלות תקטן ל 16:



איור 3.5.3 ג׳ – מפת הקריאות לפונקציה

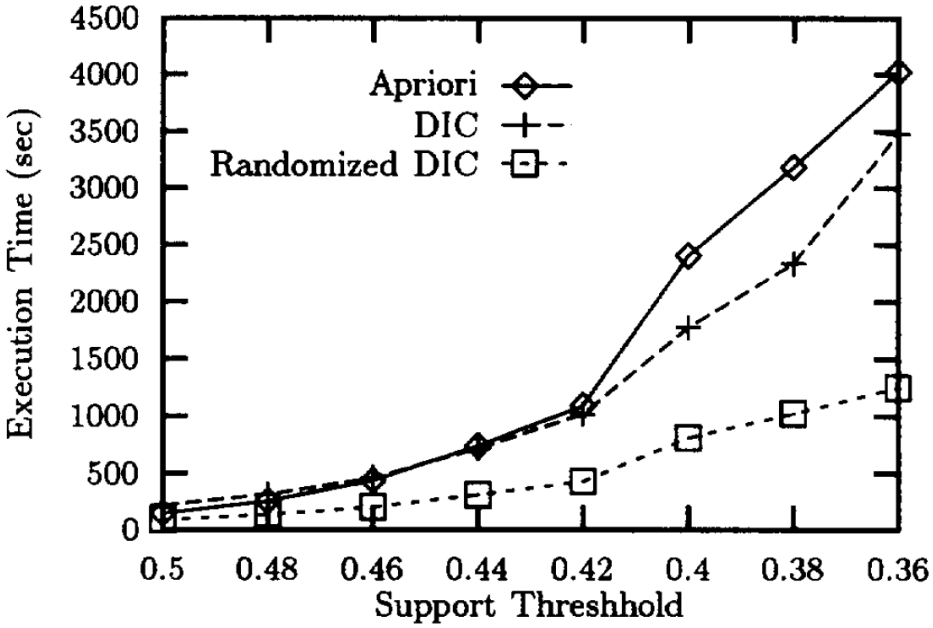
הפרשי העלויות בקריאות נובעים מכך שהעברנו לתחילת המילה את העצמים שאינם בעלי צאצאים, כך חסכנו את המעבר על מילה ארוכה לשווא. מאידך דווקא העצמים בעלי הצאצאים

המרובים – נדחו לסוף המילה וממילא למרות המעברים המרובים לא "נסחוב" איתנו במורד הרקורסיה מילה ארוכה מדי.

הסיבוכיות של DIC תלויה ב M , אך ניתן לומר כי במצבים רבים ניתן לסיים את ריצת האלגוריתם בשני מעברים בודדים על כל בסיס הנתונים – O(n).

3.5.4 תוצאות ניסיוניות

DIC הושווה ל אפריורי במתווי מידע שונים (סינטטי,אמיתי). באיור 3.5.4 א' מוצגות תוצאות ההרצה ההשוואתית.



איור 3.5.4 א' [6]

ניתן לראות כי ככל שהתמיכה יורדת (כלומר תתי הקבוצות התדירות גדלות) גדל הפער בין DIC לאפריורי (לטובת DIC). בצורה ניסיונית התקבל כי התוצאה הטובה ביותר עבור DIC הייתה כאשר M היה שווה ל 1000. מסתבר גם כי בניגוד למה שנטען מקודם הסדר הפנימי של האיברים בכל תנועה הינו זניח בהיבטים של סיבוכיות האלגוריתם.

בגרף זה מוצג גם DIC כאשר סדר התנועות שונה בצורה אקראית. השימוש בסדר תנועות אקראי הינו חשוב מכיוון ש DIC רגיש מאוד לרמת ההומוגניות של המידע. במקרים מסוימים האלגוריתם לא יבחין אם תת קבוצה אכן אמורה להיות תדירה עד שנעבור ממש על רוב בסיס הנתונים (לדוגמא אם רוב המופעים יופיעו רק בסוף הסריקה המעגלית של בסיס הנתונים שהאלגוריתם מבצע). ניסיונות שבוצעו באלגוריתם וכללו פיזור אקראי של המידע הובילו לשיפור משמעותי בביצועים. מאידך לא ניתן תמיד לבצע פיזור של בסיס הנתונים מסיבות שונות. בעיקר עקב בעיות של נפחי אחסון (ביצוע של רנדומיזציה יצריך יצירת עותק רנדומי של בסיס הנתונים – פעולה זו תכפיל את הנפח הנדרש לבסיס הנתונים ולעיתים נפח זה לא יהיה בנמצא.
כמו כן, תהליך הרנדומיזציה עצמו יקר מבחינה חישובית.

ניתן לפתור את הבעיה בשימוש ברנדומיזציה וירטואלית של המידע – במקום ליצור עותק אקראי של המידע נסרוק את המידע בצורה אקראית: נוכל לקבוע כי כל X תנועות נדלג למקום אחר בבסיס הנתונים בצורה אקראית. כך נסרוק את בסיס הנתונים בצורה אקראית ללא יצירת עותק אקראי של המידע. עבור אפריורי עובדה זו לא שינתה את התוצאות מאידך עבור DIC , התוצאות שהתקבלו טובות בהרבה.

3.5.5 סיכום

קיימים מספר יתרונות ל DIC. העיקרי שבהם הוא יתרון הביצועים. במצב שהמידע הומגני וM מספיק קטן, האלגוריתם יבצע סדר גודל של שני מעברים על כל בסיס הנתונים. בנקודה זו האלגוריתם מהיר בהרבה מאפריורי – בו נאלצים לבצע מספר מעברים כגודל של ה itemset המועמד המקסימלי. כמו כן, DIC מספק לנו את הגמישות להוסיף ולמחוק תתי קבוצות שהתגלו במהלך הסריקה הצורה דינאמית. כתוצאה מכך ניתן למקבל את האלגוריתם ולשפר בהרבה את ביצועיו. אחת החולשות של DIC היא פריסה לא הומוגנית של מידע. במצב כזה נוכל לעבור על בסיס הנתונים עד שנגלה אם עצם הינו תדיר / לא תדיר. במצב כזה כדאי ליצור מעבר אקראי על התנועות (ולא לפי הסדר) בכדי לשפר את הביצועים.[59]

ניתן לומר כי, השילוב בין DIC לסידור אקראי של התנועות בבסיס הנתונים משפר בצורה ניכרת את ביצועי האלגוריתם. ניתן להרחיב את DIC במספר היבטים שונים:

- מקביליות – ניתן ע"י DIC להשתחרר מהתלות בתוצאות של סריקה N של בסיס הנתונים בכדי להתחיל את סריקה N+1, כלומר ניתן כעת להמשיך בסריקה ללא ידיעה מוגמרת מי המועמדים להיות תדירים אלא על סמך חשש בלבד. ההתאמות הסופיות ייעשו בהמשך כאשר המידע המלא יגיע. באפריורי לעומת זאת, לא ניתן לעבור שלב באלגוריתם ללא סיום השלב הקודם. חוסר התלות בשלב הקודם ב DIC מאפשרת מימוש מקבילי של האלגוריתם.
- עידכונים בזמן אמת – ניתן בזמן אמת (במקרה של שינוי בבסיס הנתונים עצמו) לגלות כי תת קבוצה תדירה הפכה ללא תדירה / ההפך. אם קבוצה שאינה תדירה הופכת לתדירה יהיה עלינו לבצע סריקה מחודשת בכל בסיס הנתונים (ולא רק בחלק שהתווסף אליו) בכדי לגלות את המופעים החדשים. DIC מאוד מותאם למקרה שכזה שכן ממילא בכל איטרציה מתקיים מעבר מחדש ל כלל בסיס הנתונים. נוכל להרחיב את האלגוריתם כך שיתמוך בעידכונים בזמן אמת של בסיס הנתונים עצמו.

[59] במקרה שבו האקראיות אינה ישימה (עקב מגבלות זיכרון וכדו') מוצעות במאמר מספר דרכים להתמודד עם נושא האקראיות בצורה אחרת, לא נתייחס אליהן במסגרת זו.

Carma 3.6

באלגוריתם זה[17] מנסים לשפר את השלב של מציאת תתי הקבוצות התדירות בכריית חוקי ההקשר. האלגוריתם מציג אפשרות חדשה של מציאת תתי קבוצות תדירות – "בזמן אמת" כלומר מתן מידע רציף למשתמש במהלך האלגוריתם, אפשרות של קבלת קלט מהמשתמש – והשפעה של קלט זה על הפלט. למרות כל זאת ברצוננו לקבל תוצאה דטרמיניסטית ומדויקת. האלגוריתם מצריך שימוש ב 2 מעברים על בסיס הנתונים בכדי למצוא את כלל תתי הקבוצות התדירות.

ברמות תמיכה נמוכות נמצא כי Carma עדיף בזמן הריצה על אפריורי ו DIC. כמו כן האלגוריתם יעיל פי 60 בהיבטי סיבוכיות זיכרון מאפריורי ו DIC.[26]

במעבר הראשון: נעשית בניה רציפה של רשת של כלל תתי הקבוצות התדירות (כלומר, הקבוצות שהינן בפוטנציאל להיות תדירות על פי הנתונים שנאספו עד כה). עבור כל חלק ברשת,האלגוריתם יספק גבול עליון וגבול תחתון לתמיכה. במשך כל זמן עבוד התנועות בבסיס הנתונים האלגוריתם מציג למשתמש את חוקי ההקשר שנוצרו ביחד עם התמיכה של כל חוק. המשתמש רשאי בכל עת לשנות את רמת התמיכה ורמת הביטחון ולהגדיל / להקטין את כמות החוקים המתקבלת.

במעבר השני: מוגדרת התמיכה במדויק עבור כל חוק ותתי הקבוצות המיותרות נמחקות ממבני הנתונים.

3.6.1 תיאור כללי של האלגוריתם

האלגוריתם נחלק לשני חלקים:

Phase I

Phase II

Phase I

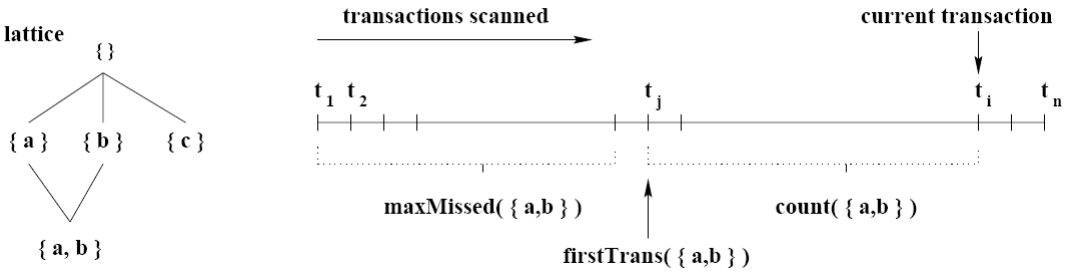
במהלך השלב הראשון של האלגוריתם, מתבצעת בניה רציפה של רשת של כלל תתי הקבוצות התדירות האפשריות. לאחר עיבוד של כל תנועה בבסיס הנתונים מתבצע עדכון של הרשת.

עבור כל תת קבוצה – v האלגוריתם מאחסן שלוש נתונים:

- Count(v) - מספר ההופעות של v מאז שהוכנס לרשת
- firstTrans(v) – האינדקס של התנועה שבה v הוכנס לרשת
- maxMissed(v) – הגבול העליון על מס' ההופעות של v לפני ש v הוכנס לרשת.

לדוגמא:

{a,b} הוכנס לרשת במהלך העיבוד של תנועה j, התנועה הנוכחית שאנו קוראים היא i.



איור 3.6.1 א' [17]

באיור 3.6.1 א' ניתן לראות את ההמחשה למושגים שהובאו לעיל.

נניח כעת כי אנו קוראים את תנועה – i. עבור כל תת קבוצה v ברשת (באיור 3.6.1. א') יש לנו גבול תחתון ועליון לתמיכה של v ב i התנועות הראשונות:

- גבול תחתון - count(v) / i כלומר: מספר הפעמים ש v הופיע מתוך כלל התנועות שנסקרו עד כה (i). יסומן ב minSupport(v)
- גבול עליון - (maxMissed(v) + count(v)) / I, כלומר: אחוז התנועות הכולל שבהן הופיעה תת קבוצה זו. יסומן ב maxSupport(v).

לאחר ש Phase I סיים את קריאת התנועה, מתבצע קידום של המונה count(v), עבור כלל תתי הקבוצות v הקיימות בתנועה הנ"ל. לאחר מכן, מוכנסות מס' תתי קבוצות אל תוך הרשת.נחשב את maxMissed ונשנה את firstTrans להצביע לתנועה הנוכחית.[60]

בסופו של דבר Phase I עלול להסיר כמה תתי קבוצות מהרשת אם ה max support שלהם היא מתחת רמת התמיכה הנוכחית. בסוף רצף התנועות האלגוריתם מבטיח כי הרשת תכיל רק תתי קבוצות של כל הקבוצות התדירות (כמובן יחסית לרף מסוים שהוגדר כתדיר – הרף הזה ניתן לשינוי דינאמי ע"י המשתמש במהלך ריצת האלגוריתם).

Phase II

בהתחלת הריצה של השלב השני באלגוריתם מתבצעת הסרה של כל תתי הקבוצות שבצורה וודאית אינן תדירות – כלומר תתי קבוצות עם Max support קטן מרמת התמיכה הנוכחית. הפעולה מתבצעת ע"י סריקה מחודשת של רצף התנועות מבסיס הנתונים. האלגוריתם מגדיר את מס' ההופעות המדויק עבור כל תת קבוצה שנותרה ומסיר את אלו שמתברר שהן קטנות מדי.

[60] נציין כי maxMissed הינו תמיד קטן יותר בערכו מהאינדקס של התנועה הנוכחית.

3.6.2 השלב הראשון של האלגוריתם – Phase I

בכדי להבין את השלב הראשון[61] באלגוריתם נקדים ונסביר שני מושגים בסיסים:

Support Lattice

בהינתן לנו רצף של תנועות ותת קבוצה v. נסמן ע"י $support_i(v)$ את התמיכה של v ב i התנועות הראשונות. נניח כי V היא הרשת של תתי הקבוצות כך שלכל תת קבוצה $v \in V$ יהיו לנו את שלושת המונים שהוזכרו לעיל ((count(v), firstTrans(v), maxMissed(v)) נכנה את V - Support Lattice, אם ורק אם V מכילה את כל תתי הקבוצות v עם $support_i(v) \geq s$. לכן ניתן לומר כי Support Lattice היא תת קבוצה של כלל תתי הקבוצות התדירות.

Support Sequence

עבור כל תנועה שנעבד בבסיס הנתונים, המשתמש חופשי לבחור סף תמיכה בצורה שרירותית. לכן נקבל רצף של רפים (ריבוי של רף) של תמיכה:

$$\sigma = (\sigma_1, \sigma_2, \sigma_3, \ldots)$$

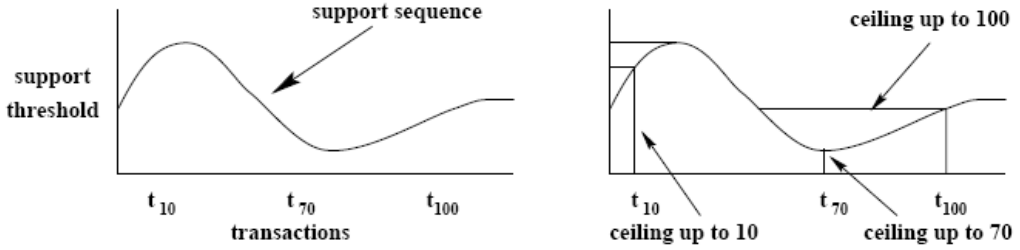
כאשר σ_i מסמן את רף התמיכה לתנועה ה i. כלומר התמיכה בתנועה זו תחושב ביחס לרף המקומי שהוגדר- σ_i.

אנו נקרא ל σ - Support Sequence .

$\lceil \sigma \rceil_i$ יסמן את הרצף המונוטוני היורד הקטן ביותר, אשר עד ל i גדול משמעותית מ σ או שווה לו ו 0 אחרת.

$\lceil \sigma \rceil_i$ יכונה התקרה של σ עד i. ראה איור 3.6.2 ב'

$avg_i(\sigma)$ יסמן את הממוצע הרץ של σ עד i. כלומר $\frac{1}{i}\sum_{j=1}^{i}\sigma_j$

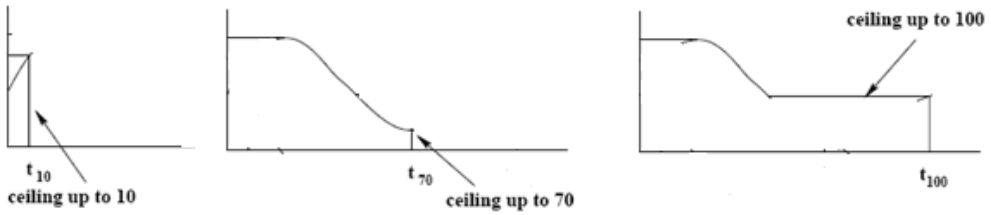


איור 3.6.2 ב' [17]

באיור 3.6.2 ב' ניתן לראות בצד שמאל דוגמא ל support sequence ובצד ימין 3 תקרות לדוגמא. באיור 3.6.2 ג' יובאו התקרות הנ"ל בצורה יותר ברורה ומפורטת.

---

[61] בהוכחת ובהסבר השלב הראשון והשני במאמר נשתמש במשפטים שהוכחו בצורה מרוכזת בסוף המאמר. לא נתמקד בהוכחות אלו במסגרת זו. אך נעשה בהם שימוש וניישם אותם בכדי לנתח את אופי פעולת האלגוריתם



איור 3.6.2 ג' [17]

כפי שציינו לעיל השלב הראשון בונה את "רשת התמיכה" שתשמש אותנו במהלך האלגוריתם.
הרשת תוגדר בצורה רקורסיבית:

בתחילה הרשת תאותחל $V = \{Ø\}$ כאשר $count(Ø) = 0$ ו $firstTrans(Ø) = 0$.

ו $maxMissed(Ø) = 0$. לכן V היא Support Lattice[62] עבור רצף התנועות הריק.

נניח כעת כי V היא support lattice עד תנועה i-1. נקרא מבסיס הנתונים את התנועה ה t_i – i.

אנו רוצים להפוך את V לרשת תמיכה עד i. נניח ו σ_i הינה רמת התמיכה הנוכחית. בכדי לנהל את הרשת עלינו להתקדם בשלושה שלבים:

<u>שלב 1 - קידום</u>

קידומו של count(v) עבור כל תתי הקבוצות $v \in V$ המוכלים ב t_i.

<u>שלב 2 - הכנסה</u>

נכניס את תת הקבוצה של v מתוך t_i אם ורק אם כלל תתי הקבוצות w של v כבר מוכלות ב V והן תדירות. כלומר $maxSupport(w) \geq \sigma_i$. בכדי להכניס את v ל Vנבצע את הפעולות הבאות:

- $firstTrans(v) = i$
- $count(v)=1$

וזאת מכיוון ש v מוכל בתנועה הנוכחית t_i.

מהעובדה כי $maxSupport(w) \geq \sigma_i$ ו גם w וגם v מוכלים ב t_i ומהעובדה כי כל תתי הקבוצות של w כבר מוכלות ב V, ניתן לומר כי:

$support_i(w) \geq support_i(v)$. לכן מתקיים כי:

$maxSupport(w) \geq maxSupport(v)$ כלומר, עפ"י ההגדרה של maxSupport נקבל:

$$MaxMissed(v) \leq maxMissed(w) + count(w) - 1$$

נציין כי ה 1 הוא בעצם – count(v).

בשימוש ב Theorem 1 נקבל

$$Support_n(v) > avg_n\left(\lceil \sigma \rceil_n\right) + \frac{|v|-1}{n} \qquad v \in V$$

[62] מדובר על רצף התנועות הריקות

עבור n = i-1 נקבל :

$$\text{Support}_{i-1}(v) > avg_{i-1}(\lceil \sigma \rceil_{i-1}) + \frac{|v|-1}{i-1} \quad v \in V$$

מכיוון ש v עדיין לא שייך ל V אי השוויון לא מתקיים ולכן נקבל :

$$\text{Support}_{i-1}(v) \leq avg_{i-1}(\lceil \sigma \rceil_{i-1}) + \frac{|v|-1}{i-1}$$

מכיוון ש

מכיוון ש $\text{maxMissed}(v) \leq \text{support}_{i-1}(v)$ (בגלל ש $\text{support}_{i-1}(v)$ מכיל את כלל המופעים של v ב i-1 התנועות הראשונות, ואילו maxMissed(v) מכיל רק את מספר המופעים של v לפני שהוכנס לרשת[63].

נקבל :

$$\text{maxMissed}(v) \leq \lfloor (i-1) avg_{i-1}(\lceil \sigma \rceil_{i-1}) \rfloor + |v| - 1$$

לכן נוכל להגדיר את maxMissed(v) כערך המינימאלי מביו הערכים הבאים :

$$\lfloor (i-1) avg_{i-1}(\lceil \sigma \rceil_{i-1}) \rfloor + |v| - 1$$

$$maxMissed(w) + count(w) \text{ -1} \mid w \subset v$$

מכיוון שאנו באיבר ה i, במקרה הספיציפי שלנו נציב באי השוויון ונקבל :

$$maxMissed(v) \leq i\text{-}1$$

שלב 3 – סינון

בשלב זה נסיר מהרשת את כלל תתי הקבוצות שגודלן גדול / שווה 2, אך ה maxSupport שלהם הינו מתחת ערך הסף הנוכחי לתמיכה : σ_i. בגלל התקורה של ביצוע שלב הסינון הוא יבוצע פעם ב $\lceil 1/\sigma_i \rceil$ או כל 500 תנועות (הגדול מבין שני הערכים הנ"ל)[64].

יש לציין כי בשיטה זו תתי קבוצות בגודל 1 לא מסוננות. לכן ניתן לומר בוודאות כי אם עצם לא מוכל ברשת הוא אינו מוכל באף תנועה בבסיס הנתונים שנסקרה עד כה. לכן כל פעם שנכניס תת קבוצה בגודל 1 אל הרשת נוכל בוודאות להציב maxMissed = 0, מכיוון שאנו יודעים בוודאות שלא הוכנסה אף תת קבוצה זהה אל הרשת.

---

[63] במקרה הזה מכיוון ש v עדיין לא הוכנס לרשת הערכים יהיו שווים, אך אנו דנים במקרה הכללי

[64] יש לציין כי כל היוריסטיקה אחרת של סינון תהיה קבילה בתנאי שתסנן רק את תתי הקבוצות שאינן תדירות. ובתנאי שלאחר שקבוצה הוסרה כל תתי הקבוצות שלה יוסרו גם כן. השיטה לעיל נבחרה בגלל היותה יעילה מבחינת סיבוכיות זיכרון.

נציג כעת את האלגוריתם במלואו ולאחריו דוגמא[17]:

Function PhaseI(transaction sequence (t_1,…..,t_n))

support sequence $\sigma = (\sigma_1 \ldots \sigma_n)$) : support lattice;
support lattice V ;
begin
V = { } maxMissed(v) = 0, firstTrans(v) = 0, count(v) = 0
for i from 1 to n do

// 1) Increment
for all $v \in V$ with $v \subseteq t_i$ do
count(v) + +;
end for

// 2) Insert
for $v \subseteq t_i$ with $v \notin V$ do
if $\forall w \subset v : w \subset V$ and maxSupport(w) > σ_i then
$V = V \cup \{v\}$
firstTrans(v) = i;
count(v) = 1;
maxMissed(v) = min { $\lfloor (i-1) avg_{i-1}(\lceil \sigma \rceil_{i-1}) \rfloor + |v| - 1$,
maxMissed(w) + count(w) -1 | $w \subset v$ }
if |v| == 1 then
maxMissed(v) = 0;
end if
end if
end for
// 3) Prune
if (i % max{$\lceil 1/\sigma_i \rceil$,500}) == 0 then
V = { $v \in V$ | maxSupport(v) > σ_i or |v| ==1}
End if

End for

return V ;

end

דוגמא:

הבהרה:

בדוגמא זו נשתמש בסימון: (maxMissed, firstTrans, count) ו ב [minSupport, maxSupport]. כלומר מספרים שיופיעו בצורה הנ"ל ישויכו לערכים הנ"ל.

לדוגמא: (0,1,1) יהיה שקול ל: maxMissed = 0, firstTrans = 1, count =1.

כמו כן, [0,5] יהיה שקול ל minSupport = 0 ו maxSupport = 5.

בהינתן לנו רצף תנועות T = {{a,b},{a,b,c},{b,c}} ורצף תמיכה של σ = (0.3, 0.9, 0.7)

נתחיל ע"י אתחול V לקבוצה הריקה. וכלל המשתנים הרלונטיים יאוחלו גם ל 0.

נקרא את $t_1 = \{a,b\}$, בתחילה נקדם את count(∅) מכיוון ש $\emptyset \subseteq t_1$, כמו כן

maxSupport(∅) = 1[65]. ומכיוון ש $\sigma_1 \leq$ maxSupport(∅) נוסיף את {a} ו {b} ל V.

כאמור maxMissed עדיין שווה 0. כלל המשתנים כעת יהיו (0,1,1).

מכיוון שאין אף תת קבוצה ב V שמקיימת maxSupport < 0.3 לא נוכל לסנן אף קבוצה מ V ונמשיך הלאה באלגוריתם.

עבור $t_2 = \{a,b,c\}$

קודם עלינו להעלות את המונה של ∅,{a},{b}. כעת נכניס את {c}, נקבע עבורו את maxMissed ל 0.

מכיוון ש $\{a,b\} \subseteq t_2$ ו a ו b הינם אלמנטים ב V עם $\sigma_2 = 0.9 \geq$ maxSupport, נוסיף את {a,b} ל V.

נחשב את maxMissed({a,b})

קודם נמצא את ערכו של $\lfloor avg_1(\lceil \sigma \rceil_1) \rfloor$.

$$\lfloor avg_1(\lceil \sigma \rceil_1) \rfloor = \lfloor avg(0.3,0,0,...) \rfloor = \lfloor 0.3 \rfloor = 0$$

maxMissed({a,b}) = min

{

$$\lfloor (2-1) avg_1(\lceil \sigma \rceil_1) \rfloor + |\{a,b\}| - 1 = \lfloor avg_1(\lceil \sigma \rceil_1) \rfloor + 2 - 1 = 0 + 2 - 1 = 1$$

maxMissed(w) + count(w) -1 | $w \subset v$ =1

} = 1.

נעדכן את המונים עבור {a,b} => (1,2,1). כמו כן רמת התמיכה של {a,b} היא:

maxSupport({a,b}) = 1.

עבור t_3={b,c}

---

[65] MaxSupport(v)=(MaxMissed(v) + count(v))/i = (0+1)/1=1

נעדכן את המונים עבור $b,c,\varnothing$. לאחר מכן נכניס את {b,c} אל תוך הרשת וזאת מכיוון ש {b} ו {c}, כלולים כבר ברשת ו $\text{maxSupport}(b)$ ו $\text{maxSupport}(c) > \sigma_3 = 0.5$

$\text{maxMissed}(\{b,c\}) = \min$

{

$$\lfloor (i-1)avg_{i-1}(\lceil \sigma \rceil_{i-1}) \rfloor + |v| - 1 = \lfloor 2avg_2(\lceil \sigma \rceil_2) \rfloor + 2 - 1 = \lfloor 2avg_2(0.9,0.9,0,0,...) \rfloor + 2 - 1 = 2$$

$$maxMissed(w) + count(w) - 1 \mid w \subset v = maxMissed(\{c\}) + count(\{c\}) - 1 = 1$$

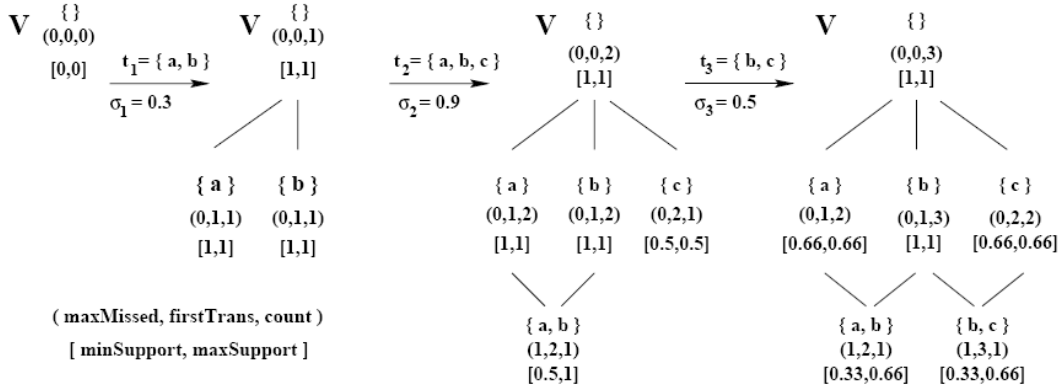
} $= \min(2,1) = 1$

מפני שלכל העצמים יש $\text{maxSupport} > 0.5$, לא ניתן להסיר אף אחד מהעצמים מתוך הרשת.

אם $\sigma_3 = 0.7$, אזי לא ניתן היה להכניס את {b,c} לרשת , כמו כן היינו מסירים את {a,b} וזאת מפני שהם לא עומדים בתנאי התמיכה הנדרשים.

מאידך לא ניתן היה להסיר את {c} וזאת מפני שלא מסירים תתי קבוצות בעלות עצם יחיד.

ניתן לראות באיור 3.6.2 ד׳ סכמה מסכמת של כלל התהליך של phase I



איור 3.6.2 ד׳ [17]

Phase I מבטיח לנו כי כלל תתי הקבוצות ברשת הינן בעלות תמיכה גדולה / שווה לסף התמיכה הנתון. וזאת מכיוון שמהלכו של התהליך אנו מבצעים סינון לתתי קבוצות שאינן עומדות בתנאי התמיכה הנדרשים.

3.6.3 השלב השני של האלגוריתם – Phase II

נניח ו V הינה רשת התמיכה המתקבלת מ Phase I. ונניח כי σ_n הינה רמת התמיכה של המשתמש לתנועה האחרונה שנקראה. ב Phase II נסיר את כלל תתי הקבוצות שאינו תדירות ונגדיר את התמיכה המדויקת לכלל תתי הקבוצות שנותרו.

בתחילה נסיר את כל תת הקבוצות הטריוויאליות – תתי קבוצות בעלות $\text{maxSupport} < \sigma_n$ מ V.

במהלך סריקה של רצף התנועות האלגוריתם יגדיל את המשתנה count ויקטין את maxMissed עבור כל תת קבוצה המוכלת בתנועה הנוכחית, עד לאותה תנועה שבה העצם הוכנס לרשת.

כאשר maxMissed = 0 נקבל כי minSupport = maxSupport. וזוהי בעצם רמת התמיכה האמיתית של תת הקבוצה הרלוונטית. כך ניתן יהיה להסיר תתי קבוצות שאינן עומדות ברמת הסף של התמיכה המינימלית הנדרשת מהן.[66]

התהליך יסתיים כאשר אינדקס התנועה הנוכחית יעקוף את firstTrans לכל תתי הקבוצות ברשת.

נציג כעת את האלגוריתם במלואו[17] :

```
Function PhaseII( support lattice V , transaction sequence (t1,..tn),
support sequence _ ) : support lattice;
integer ft, i = 0;
V = V \{v ∈ V \ maxSupport(v ) < σn}
while ∃v ∈ V  : i < firstTrans(v)
        i++;
        for all  v ∈ V
                ft = firstTrans(v);
                if v ⊆ ti  and ft < i then count(v)++, maxMissed(v)- -;
                if ft = = i then
                        maxMissed(v) = 0;
                        for all  w ∈ V : v ⊂ w and maxSupport(w) > maxSupport(v) do
                                maxMissed(w) =  count(v) - count(w);
                if maxSupport(v) < σn then V = V \{v}
end while
Return V;
```

[66] הצבה של maxMissed = 0 יכולה להוביל למצב של maxSupport(w) > maxSupport(v) כאשר w הינה תת קבוצה של v. לכן נציב מראש maxMissed(w) = count(v) – count(w) כאשר maxSupport(w) < maxSupport(v).

Carma 3.6.4

האלגוריתם השלם משלב בעצם, את שני השלבים שנידונו בסעיפים הקודמים.

כפי שצוין לעיל Phase 1 – מייצר תתי קבוצות תדירות בהתחשב בסף התמיכה הרלוונטי.

Phase 2 - מסיר את כל תתי הקבוצות שאינן התדירות, כך מובטח לנו שיישארו רק תתי קבוצות תדירות בבסיס הנתונים המעובד[17].

```
Function Carma( transaction sequence T, support sequence _ )
support lattice V ;

begin
        V = PhaseI( T, σ);
        V = PhaseII( V , T, σ)
return V ;


end;
```

3.6.5. ביצועים

בכדי להשוות בין האלגוריתם הנוכחי לאפריורי ו DIC, הורצו כל שלושת האלגוריתמים בתמיכה קבועה.

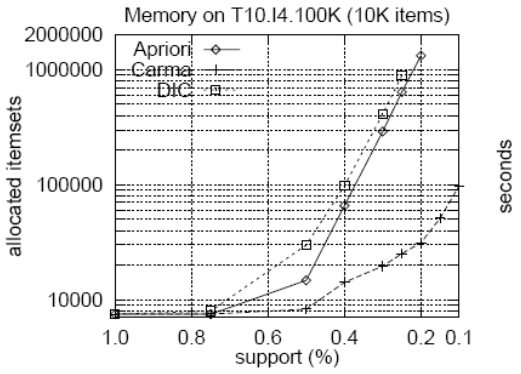


Figure 9

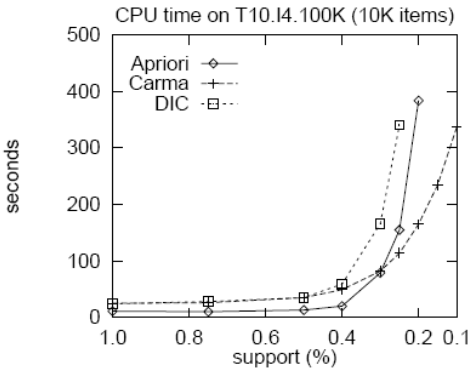


Figure 10

איור 3.6.5 א' [17]

ניתן לראות באיור הימני את הזמנים של האלגוריתמים, באיור השמאלי ניתן לראות את הזיכרון שכל אלגוריתם צורך במהלך הריצה.

עבור תמיכה של 0.5% ומעבר לכך, אפריורי עדיף על DIC ו Carma. אך ככל שהתמיכה יורדת (0.25%) המספר של תתי הקבוצות התדירות בגודל 1 גדל בצורה משמעותית. ניתן לומר כי מתחת ל 0.25% ביצועיו של Carma עדיפים על פני אלו של אפריורי ו DIC.

ניתן לייחס את ההפרשים לטובת Carma לעומת אפריורי בהתייחסות למספר הסריקות של בסיס הנתונים שבוצעו (4 של אפריורי לעומת 1.1 של Carma)

ביחס ל DIC ניתן לומר כי בעוד DIC סרק את בסיס הנתונים 2 פעמים ל Carma הספיקו 1.1 פעמים. כמו כן מבנה הנתונים של Carma קטן בצורה משמעותית (פי 35) לעומת DIC.

נציין כי תמיד Carma ישלים את האלגוריתם לאחר 2 מעברים על בסיס הנתונים לכל היותר.
למרות שבמקרה הכללי Carma לא מהיר משמעותית יותר מאפריורי ו DIC, עדיין קיימת עדיפות ל Carma בטווחים נמוכים של תמיכה. ניתן לייחס עדיפות זאת למבנה הנתונים היעיל בו עושה האלגוריתם שימוש.

נציין כי בחלק היישומי של העבודה המסכמת נעשה שימוש באלגוריתם Carma מתוך SPSS.

3.7 יצירת חוקי הקשר מ ***Frequent Itemsets***

לאחר שברשותנו קבוצה של frequent itemsets נוכל לחשב כעת את חוקי ההקשר[35].

נשתמש במשוואה הבאה [13]

$$confindence(A \Rightarrow B) = P(B \mid A) = \frac{\sup ort\_count(A \cup B)}{\sup port\_count(A)}$$

לכן, עבור כל L - frequent itemset . נחולל את כל תתי הקבוצות הלא ריקות שלו.

לאחר מכן עבור כל תת קבוצה s נדפיס את החוק : (L-s) → s.

החוק יודפס אך ורק אם : $\frac{\sup port\_count(L)}{\sup port\_count(s)} \geq \min\_conf$ כאשר min_conf הוא ה

confidence המינימלי שהוגדר .

יש לשים לב כי אנו לוקחים בחשבון רק את ה confidence וזאת בגלל שה support המינימלי כבר נלקח בחשבון בעצם יצירת ה itemset.

דוגמא :

בהמשך לדוגמא שהוצגה מקודם נוכל לומר כי :

תתי הקבוצות של { 2 3 5 }

{2} {3} {5} {2 3} {2 5} {3 5}

החוקים שיווצרו יהיו :

{2}→{3 5} confidence = 2 / 3 = 66%

{3} →{2 5} confidence = 2/ 3 = 66%

{5} →{2 3} confidence = 2/3 = 66%

{3 2} → {5} confidence = 2/2 = 100%

{2 5} →{3} confidence = 2 / 3 = 66%

{3 5} →{2} confidence = 2 / 2 =100%

במידה והחוקים הנ"ל יעברו את מדד ה confidence המינימאלי שהוגדר אזי ניתן יהיה לומר כי הם חוקים תקפים.

4. השוואה מסכמת בין השיטות

מאמרים רבים דנים בהשוואות[67] ובסקירות שונות של חלק מהשיטות / כולן.
בחלק זה נביא סקירה כוללת של האלגוריתמים שנדונו לעיל. כמו כן, יובאו מסקנות מניתוח השוואתי של האלגוריתמים והבדלים ביניהם בהיבטים של סיבוכיות זמן מקום והיבטים נוספים [15] [18].

<u>מרחב החיפוש</u>

למרחב החיפוש השפעה מרובה על ביצועי האלגוריתם. כפי שציינו בעבר[10] השיטה הנאיבית למציאת חוקי הקשר, בודקת את כלל תתי הקבוצות הקיימות (סה"כ 2^I – כאשר I הינו מספר האטומים בבסיס הנתונים). כתוצאה מכך פותחו אלגוריתמים שטבעו את המושג – תתי קבוצות המיועדות להיות תדירות:

(Candidate Itemsets). אלגוריתמים אלו ניסו בדרכים שונות לצמצם את מספר תתי הקבוצות שנבדקות. אחת מהן היא תכונת המונוטוניות של התמיכה הטוענת שעבור תת קבוצה שאינה תדירה גם תתי הקבוצות שמכילות אותה אינן תדירות. כתוצאה מכך מרחב החיפוש עבור תתי הקבוצות התדירות הצטמצם משמעותית, כעת ניתן יהיה לפסול מחיפוש תתי עץ שלמים ולצמצם את תתי הקבוצות שאותן עלינו לבדוק. כמו כן קיימת אפשרות ע"י מושג הגבול[68] לדעת את החסם של מספר תתי הקבוצות שעלינו לבדוק. נדון בהמשך בצורת צמצום מרחב החיפוש של כ"א מהאלגוריתמים.

<u>ייצוג בסיס הנתונים</u>

נתון חשוב בכל אלגוריתם היא הצורה שבה האלגוריתם מייצג את בסיס הנתונים. קיימות מספר אפשרויות לייצג בסיס נתונים בצורה בינארית:

- ייצוג אופקי – לכל תנועה יש מזהה ורשימה של עצמים שמופיעים בה.
- יצוג אנכי – לכל אטום מבסיס הנתונים נציין באילו תנועות הוא מצוי

	beer	wine	chips	pizza
100	1	1	1	0
200	1	0	1	0
300	0	1	0	1
400	0	0	1	1

	beer	wine	chips	pizza
100	1	1	1	0
200	1	0	1	0
300	0	1	0	1
400	0	0	1	1

איור 4.1 א'[10]

באיור 4.1 א' ניתן לראות דוגמאות לייצוגים השונים.
בכדי למצוא תמיכה של תת קבוצה בבסיס נתונים אופקי יהיה עלינו לסרוק את כל בסיס הנתונים ולבדוק בל תנועה האם העצם מופיע בה או לא. עבור בסיס נתונים אופקי ניתן לחשב את התמיכה בקלות ע"י חיתוך העמודות של כלל מרכיבי תת הקבוצה הנ"ל וספירת האחדות. החיסרון הוא

[67] חלק מן המאמרים עושה שימוש ב [9] וב [4] [5]לצורך מימוש האלגוריתמים.

[68] מושג הגבול – Border לא יורחב במסגרת זו, ניתן לקרוא עליו ב [10]

שעלינו להחזיק בזיכרון המרכזי את העמודות של תתי קבוצות רבות. בסופו של דבר במקרה ההבדלים בין השיטות אינם כה משמעותיים.[19]הממוצע

כעת נעבור על מאפייני והבדלי האלגוריתמים השונים במרוכז[69].

<u>אפריורי</u>

אלגוריתם אפריורי עובר על פני כל בסיס הנתונים וסופר עבור כל אטום את מספר המופעים שלו. לאחר מכן מסוננים האטומים שתמיכתן נמוכה מהתמיכה המינימלית. כעת מייצרים מהקבוצה שהתקבלה קבוצה בגודל גדול ב 1. וזאת ע״י פעולת join טבלאית. לאחר מכן שוב נבדוק תמיכה ונסנן. ונמשיך לשלב הבא. נציין[20] כי פעולת החיפוש באפריורי מבוססת אלגוריתם BFS לחיפוש בעצים.

<u>DIC</u>

ב DIC[19] מנסים להפחית את מספר המעברים על בסיס הנתונים(יחסית לאפריורי) ע״י חלוקת בסיס הנתונים לאינטרוולים. וספירה מעגלית עבור כל קבוצת מועמדים בפני עצמה. בכל מקרה יש לציין כי ביצועי DIC תלויים במידה רבה במידת הפיזור של המידע בבסיס הנתונים.

<u>Eclat</u>

Eclat שייך למשפחת האלגוריתמים המשתמשים בשיטת DFS, בשונה מ DIC ו Apriori, שבהם אנו עוברים על כל תתי הקבוצות מאותו גודל (לדוג׳ כלל תתי הקבוצות בגודל 3) ובודקים אותם – תהליך זה שקול למעבר בצורת BFS על עץ תתי הקבוצות. ב Eclat המעבר נעשה לעומק, כלומר אנו חוקרים במהלך האלגוריתם **לעומק** את כלל תתי הקבוצות המיוחסות לתת קבוצה מסוימת (נוצרות ממנה).באלגוריתם זה [5], תתי הקבוצות התדירות מחושבות ע״י חיתוכים אנכיים (בניגוד לאופקיים שבה משתמשים בשאר השיטות) של בסיס הנתונים. לא נעשה כאן שימוש בתכונת המונוטוניות של תתי הקבוצות. חילול קבוצה נעשה ע״י 2 תתי קבוצות שלה בלבד. עובדה זו גורמת לכך שייווצרו הרבה יותר תתי קבוצות לבדיקה מאשר באפריורי וב DIC.

נזכיר גם כי בשונה מ DIC ואפריורי ל Eclat אין שלב סינון של תתי קבוצות לא תדירות.

כאשר קיים בסיס נתונים בו קיימים הרבה עצמים תדירים Eclat יחולל כל תת קבוצה אפשרית בגודל 2 ללא קשר לקיומה בבסיס הנתונים או לא. מאידך, במצב שבו קיימים תתי קבוצות תדירות גדולות ורבות, אפריורי יהיה נחות מבחינת ביצועים לעומת Eclat שכן גם אפריורי יחולל בכל מקרה את כל תתי הקבוצות בגודל 2 ובמצב זה Eclat יעלה בביצועיו על אפריורי.

נציין כי למרות זאת, במצב שבו הזיכרון אינו מספיק בשביל לאחסן את כל תתי הקבוצות המועמדות להיות תדירות לא ניתן יהיה להשתמש ב Eclat. לעומת זאת, אפריורי יתמודד בקלות עם הבעיה (לדוגמא בשימוש בשיטת החלוקה).

---

[69] לא נדון כאן בצורה השוואתית ב Carma מכיוון שהוא אינו משתייך למשפחות האלגוריתמים הנבחנים אלא למשפחת online generation. ולכן לא ניתן לשייך את Carma ל DFS או ל BFS. סקירה השוואתית עבור Carma לעומת אפריורי ו DIC הובאה בפרק על Carma.

FP Growth

האלגוריתם הנ"ל משתמש בשיטת הפרד ומשול ע"י חלוקה של בסיס הנתונים לתתי קבוצות[16]. בכדי לספור את התמיכה עבור כל תתי הקבוצות שחוללו, האלגוריתם משתמש בשילוב של בסיס נתונים רחובי ואופקי. האלגוריתם מאחסן את התנועות של בסיס הנתונים במבנה של עץ. לכל עצם יש רשימה מקושרת המקשרת בין מופעיו בכל ענפי העץ. הסיבה לאיחסון התנועות בעץ בסדר יורד על פי התמיכה (כלומר, סידור העצמים בכל תנועה) הינה הרצון לחסוך מקום ע"י שיתופי ענפים בעץ[70].

בהינתן לנו עץ FP שכזה ניתן לומר בעצם כי העץ הינו בעצם תצוגה שונה של בסיס נתונים אופקי ואנכי בצורה הבאה:

כל רשימה מקושרת המתחילה מהטבלה הראשית המחוברת לעץ מהווה בעצם סוג של רשימת התנועות שבהן מצוי כל עצם – ייצוג אנכי של בסיס הנתונים.

אם נסתכל על העץ עצמו, החל מהשורש מייצג בעצם את התנועות של בסיס הנתונים – ייצוג אופקי של בסיס הנתונים. בכדי לבצע כריה על העץ נתחיל מכ"א מהמסלולים התדירים בעלי אורך 1. , נבנה "תת בסיס נתונים" / "תת עץ" המכיל את מסלולי התחיליות המכילים את התחילית הנוכחית בעץ. מקבוצת מסלולים זו נבנה עץ FP ונכרה אותו בצורה רקורסיבית.

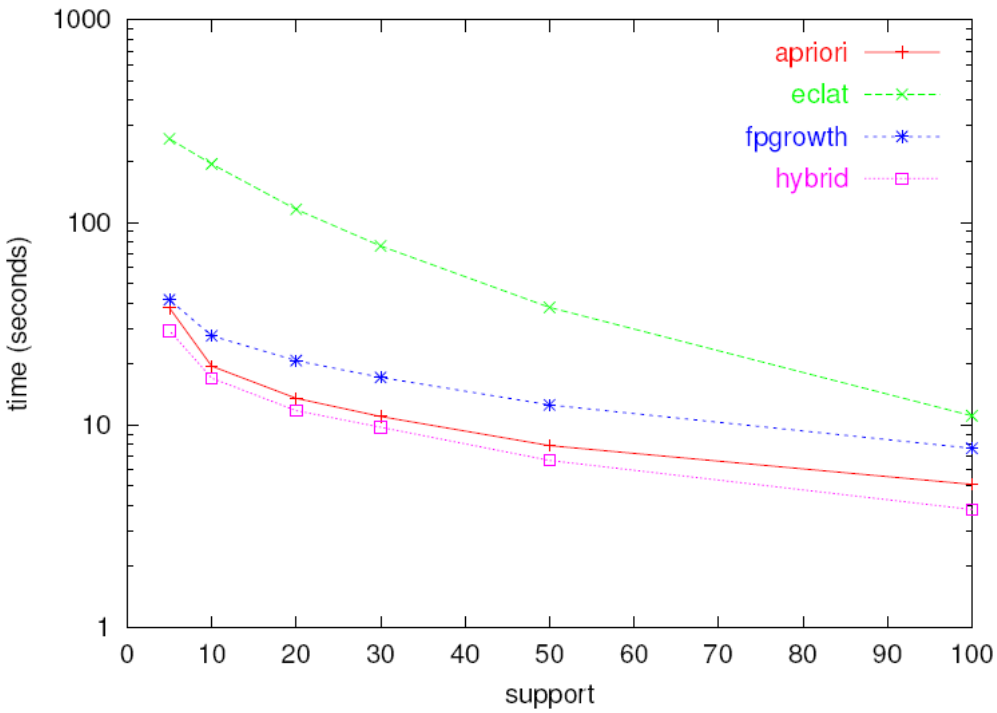
חוץ מהעץ FP עצמו האלגוריתם דומה מאוד ל Eclat. [22] ההבדל העקרוני בין שני האלגוריתמים הינו אופן חישוב התמיכה של כל תת קבוצה המועמדת להיות תדירה. כמו כן קיים הבדל בצורה שבה כל אחד מהאלגוריתמים מייצג ומנהל את החלק מבסיס הנתונים שבוצעה עליו הטלה.

היתרון העיקרי של FP – Growth לעומת Eclat הוא זה שכל רשימה מקושרת המייצגת את כלל התנועות שעצם מסוים מאוחסן בהם מאוחסנת בצורה דחוסה. החיסרון בעובדה זו הינה הניהול של מבנה הנתונים המסובך שנצרך לייצוג בסיס הנתונים בצורה קומפקטית. חישובים [11][10] מראים שכדי שהדחיסה תהיה יעילה גודלו של עץ ה FP צריך להיות 20% **לכל היותר** מהגודל של כלל ייצוג מבנה הנתונים ב Eclat.

תוצאות ההשוואה

בוצעה השוואה בין כל האלגוריתמים שהוצגו לעיל, נעשה שימוש באופטימיזציות שונות שלא נפרטן כאן. האלגוריתמים נבדקו מול בסיסי נתונים שונים ומול תמיכות מינימליות משתנות.

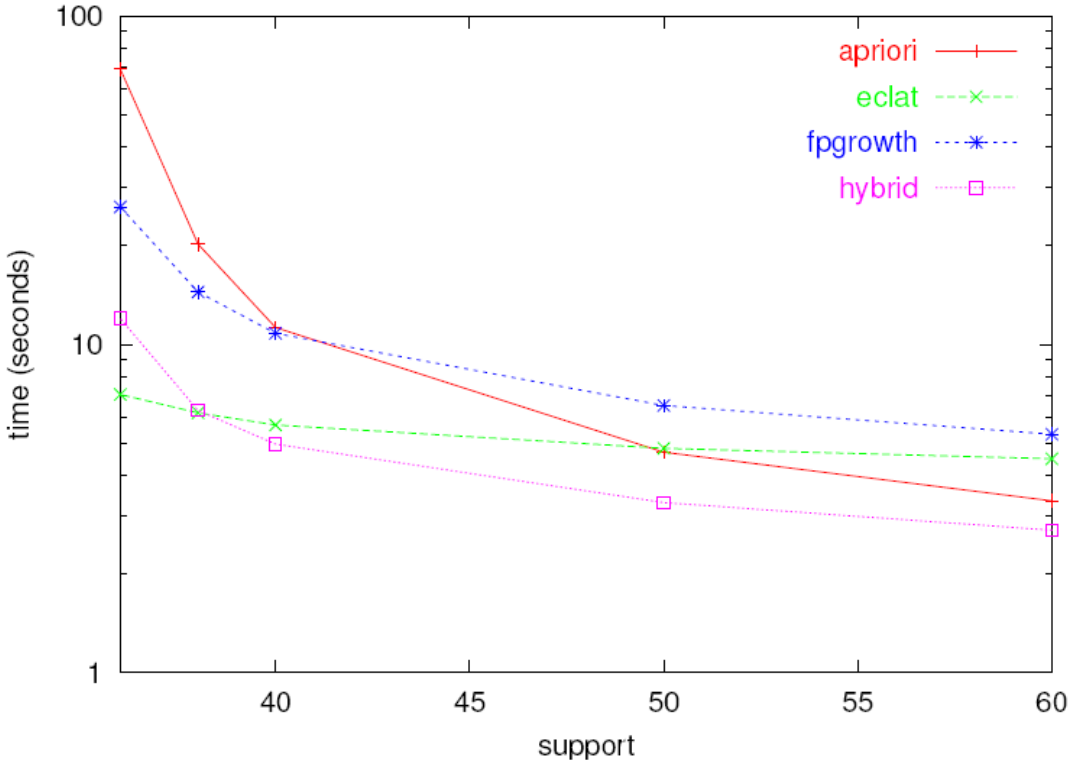
[70] נושא זה הוסבר בפירוט בפרק על FP - Growth



איור 4.1 ב׳- basket [10]

ניתן לראות שעבור בסיס הנתונים basket (איור 4.1 ב׳) ביצועיו של Eclat הינם הגרועים ביותר, כפי שתיארנו מראש התנהגות זו צפויה. מכיוון שעבור בסיס הנתונים הנ״ל מספר תתי הקבוצות התדירות הינו גדול מאוד. לכן תיווצר כמות גדולה מאוד של תתי קבוצות המיועדות להיות תדירות, תוצאה ישירה מכך היא הגדלת זמן הריצה של האלגוריתם.

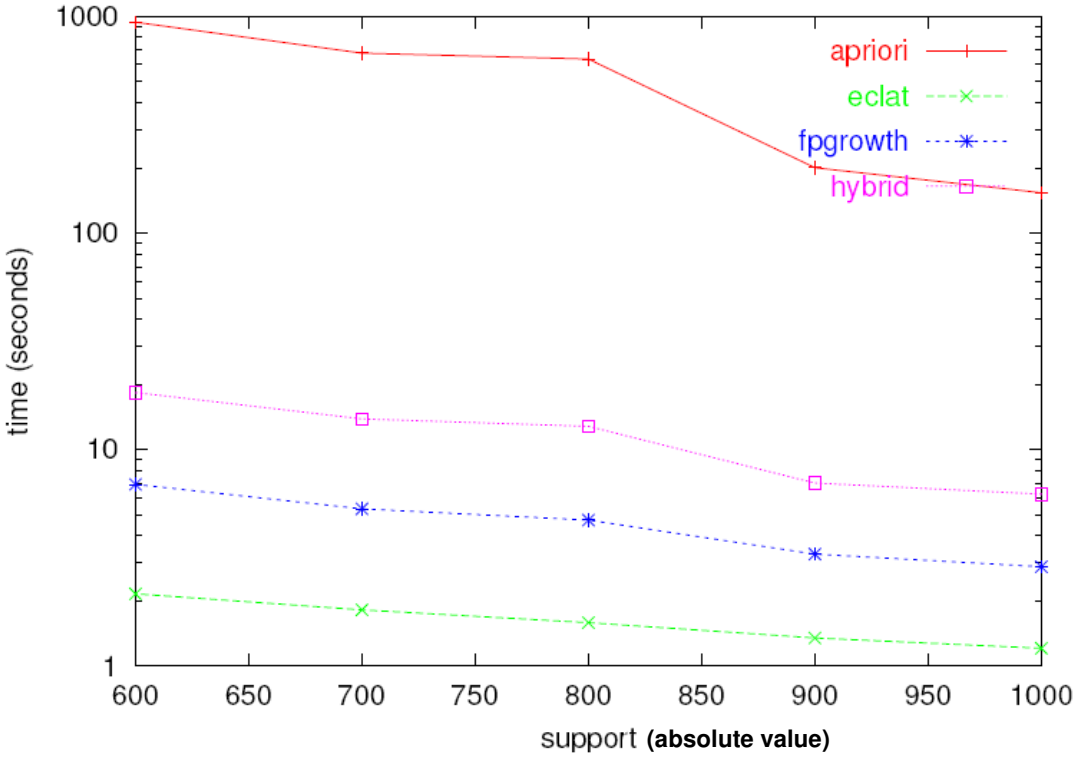
ניתן גם לשים לב שזמן הריצה עבור אפריורי טוב יותר מ FP – Growth. הסיבה לכך היא התקורה שבניהול מבנה הנתונים המסובך של FP – Growth. מכיוון שבסיס הנתונים הנ״ל הינו דליל מאוד עידכוני תמיכה לכלל התנועות בבסיס הנתונים מתבצעות בצורה מהירה. בשונה מ FP- Growth שייצר עבור בסיס הנתונים הנ״ל עץ נרחב, לכן העדכונים בעץ במהלך האלגוריתם לוקחים זמן רב ומאטים את פעילותו של האלגוריתם.



איור 4.1 ג' [10]

גם עבור בסיס הנתונים באיור 4.1 ג' hybrid ממשיך להוביל עם זמני הריצה הטובים ביותר. כאשר שיעורי התמיכה המינימלית גבוהים מ 40 ההבדלים בביצועים בין האלגוריתמים הינם זניחים. ונובעים בעיקר מהבדלים ותקורות באיתחול ומחיקת מבני הנתונים הפנימיים של כל אלגוריתם.

נציין כי עבור שיעורי תמיכה נמוכים במיוחד ביצועיו של Eclat עולים על ביצועי שאר האלגוריתמים. הסיבה העיקרית לירידה בביצועים של אפריורי בבסיס הנתונים הנ"ל היא קיומם של תנועות גדולות מאוד בבסיס הנתונים שעבורן מציאת תתי קבוצות וספירת תמיכה נמשכות זמן רב.



איור 4.1 ד' [10]

עבור בסיס הנתונים באיור 4.1 ד' ההבדלים בביצועים בין Eclat ל FP – Growth הינם זניחים וקיימים כתוצאה מהבדלים באיתחולים ומחיקות של מבני נתונים פנימיים באלגוריתם. גודלו של בסיס נתונים זה הינו קטן מאוד. לכן שניהם רצים מהר מאוד (יחסית). לעומת זאת זמן הריצה של אפריורי ארוך בצורה ניכרת. וזאת מכיוון שכל תנועה מכילה בדיוק 23 עצמים בעלי תמיכה גבוהה. מצב זה גורם להאטה משמעותית בזמן הריצה גם עבור אפריורי וגם עבור האלגוריתם ההיברידי[71].

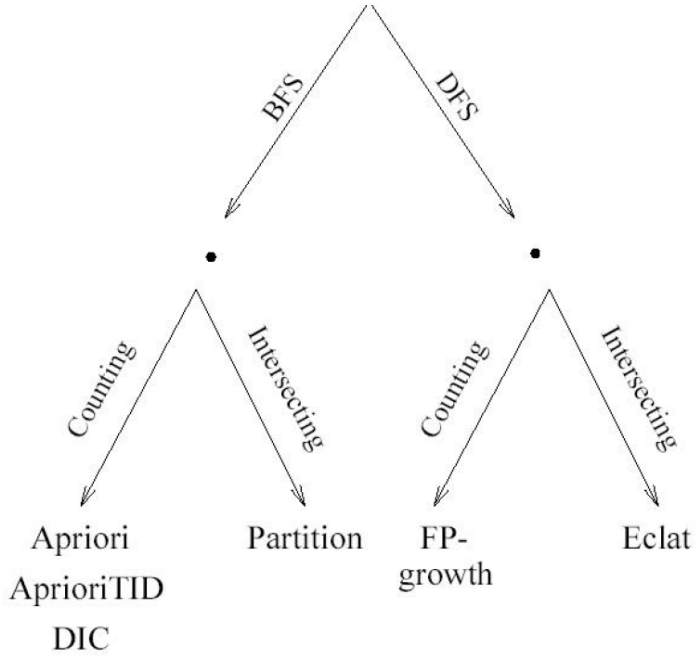
לסיכום נציג טבלה מסכמת של כל מאפייני האלגוריתמים שנבחנו. נציין כי מבחינת ביצועים במקרה ממוצע הדירוג הוא:

1. Eclat + Hybrid
2. FP Growth
3. Apriori

[71] נציין כי ב [12] בוצעה השוואה בין מבני נתונים שונים למימוש Apriori, Eclat, FP – Growth. לא נרחיב על תוצאותיה במסגרת זו.

שיטה	רציונל	חסרונות	יתרונות
אלגוריתם נאיבי	מציאת כל תתי הקבוצות התדירות בקבוצת תנועות נתונה(בסיס נתונים)	זמן ריצה מעריכי	פשוט וקל להבנה
אפריורי	הקטנה מראש של מספר התנועות ע״י תכונת אפריורי	1) לא יעיל עבור בסיסי נתונים גדולים מאוד 2) מניח כי תדירות ההופעה של כל ה itemsets בבסיס הנתונים הינה אחידה.	לא רץ בזמן מעריכי
FP Growth	הקטנה של סיבוכיות מקום והקטנת זמן ריצה ע״י הימנעות מחילול מיותר של תתי קבוצות תדירות	*ברמות תמיכה נמוכות העץ גדול מאוד וסבוך *מצריך ניהול של מבנה הנתונים	* דחיסת בסיס הנתונים *כל תת קבוצה שמחוללת הינה חלק מהתוצאה
Eclat	חיפוש ע״י שימוש במודל הרשת	חילול תתי הקבוצות נעשה רק על סמך 2 תתי קבוצות תדירות בכל פעם – מגדיל את כמות ה Itemsets שמחוללים.	*שימוש בחיתוך בין רשימות TID. *מבני נתונים מופשטים

תרשים מסכם המציין בצורה סכמטית את השתייכותו של כל אחד מהאלגוריתמים למשפחות השונות ניתן לראות באיור 4.1 ה'.



איור 4.1 ה'[18]

5. שימושים רפואיים בכריית חוקי הקשר

לכריית מידע בכלל ולכריית חוקי הקשר בפרט יש שימוש רב בתחומים שונים. בשנים האחרונות הולך וגובר השימוש בשיטות לכריית מידע לצורך מחקר רפואי וזאת בגלל כמות המידע הגדולה הקיימת בתחומים אלו. שימוש באמצעים ממוחשבים לניתוח המידע ולהיתוכן יכול לפתוח אופקים חדשים במחקר הרפואי ולקדם אותו צעדים רבים קדימה.
בחלק זה נסקור בקצרה שימושים רפואיים / ביולוגיים בכריית מידע בכלל ובחוקי הקשר בפרט. נשתדל לא להרחיב יתר על המידה אלא רק להסביר את הרעיון העיקרי של שימוש בחוקי הקשר בתחום הנידון.

5.1 כריית חוקי הקשר בתחום הרפואי

קיימות מספר שיטות כריית מידע בתחום הרפואי (אפידימיולוגי). בוצעו השוואות בין שימוש בחוקי הקשר לשיטות נוספות [45][44] [48]בחלק זה נתמקד דווקא בשימושים רפואיים בחוקי הקשר. נתמקד בעיקר בעקרונות עיבוד מידע רפואי בשימוש בחוקי הקשר. בסופו של הפרק יובאו מספר מחקרים לדוגמא.
קיימים מספר מאפיינים יחודיים עיקריים בכריית מידע רפואי:

- קשיים בטיוב הנתונים – בחלק גדול מן המקרים איכות הנתונים אינה גבוהה, חסרים נתונים וכדו'.
- בעיות בתהליך הכרייה ובניתוח התוצאות – לעיתים מתקבלות תוצאות שאינן אינפורמטיביות ואינן רלוונטיות למציאות הרפואית.

5.1.1 עקרונות כריית חוקי הקשר במידע רפואי

קיימים מספר היבטים טכניים המקשים על פעילות הכרייה של מידע רפואי[47][72]:

- סוג נתונים מגוון ועשיר (תמונות, נומרי, טקסטואלי).
- לרוב, המידע מכיל רעשים רבים.
- חלק גדול מבסיסי הנתונים קטנים וקשים להשגה.

בכדי לבצע כריית חוקי הקשר עלינו להעביר את המידע הגולמי עיבוד מקדים[47] שיאפשר לנו לאחסן אותו בבסיס הנתונים הטבלאי ולבצע עליו את פעולת כריית המידע. לצורך כך, נתייחס לכלל התכונות בבסיס הנתונים כ מספריות (נומריות) / או כניתנות לסיווג (קטגוריאליים).
בנוסף יש צורך להתייחס מראש למידע חסר עבור כל אחת מהתכונות בבסיס הנתונים. התייחסות זו הינה הכרחית מהסיבות הבאות:

- ערכים חסרים הינם נפוצים מאוד בבסיסי נתונים רפואיים.
- ישנה חשיבות למידע חסר מכיוון שניתן בעזרתו לגלות שגיאות באבחון. יש לשים לב שחוסר מידע אינו חד משמעי – לעיתים חוסר מידע מציין כי לנבדק אין שום בעיה באזור

[72] המאמר הנ"ל דן בעקרונות כריית מידע במחקר רפואי. לצורך דוגמא משתמש המאמר בבסיס נתונים המכיל מידע על חולי לב. לא נדון במסגרת זו בתהליך אותו ביצעו החוקרים על בסיס הנתונים, נתמקד בעקרונות כריית המידע הרפואי בלבד.

המדובר, ולעיתים מציין חוסר המידע – חוסר ממשי של מידע בבדיקה הנובע מחוסר יכולת להשיג את המידע הנדרש.

- עקב קוטנו של בסיס הנתונים לא נוכל להרשות לעצמינו להתעלם מרשומות חסרות ולכן חובה עלינו להתייחס אליהן ולמצות מהן את מידע ככל שניתן.

בכדי להתמודד עם התכונות המספריות של בסיס הנתונים נעשה שימוש בשיטת החלוקה. שיטה זו נוחה מאוד לשימוש בהקשר הרפואי, כיוון שמרבית הרופאים יודעים לסווג בעצמם את התוצאות המספריות לקבוצות. תהליך זה יבוצע ע"י אלגוריתם מיפוי שיהפוך (או בצורה אוטומטית או בהתערבות המשתמש) את הנתונים לתכונות קטגוריאליות בבסיס הנתונים.

Gender	Age	Smokes	LAD %	RCA %
F	53	Y	85	100
M	62	N	80	0
M	75	Y	70	80
M	73	Y	40	99
M	66	N	50	45

Table 1: Original medical data

M	F	$Age < 70$	$70 \leq Age$	S=Y	S=N	$LAD < 50$
1	2	3	4	5	6	7

$50 \leq LAD < 70$	$70 \leq LAD$	$RCA < 50$	$50 \leq$RCA
8	9	10	11

Table 2: Mapping table

איור 5.1.1 א' [47]

באיור 5.1.1 א' ניתן לראות רשומת נתונים לדוגמא ואת הטבלה שנוצרה ממנה. הטבלה מכילה מספר מזהה לכל תכונה בבסיס הנתונים. ניתן לשים לב כי LAD[73] פוצל לשלושה תחומים בעוד RCA[74] פוצל רק לשניים, עובדה זאת נובעת מהחשיבות של LAD באבחון והרצון לאבחנו בצורה מדויקת יותר מ RCA.

[73] **L**eft **A**nterior **D**escnding artery – מאפיין רפואי הקשור לעורק הכלילי בצידו השמאלי של הלב

[74] **R**ight **C**oronary **A**rtery - מאפיין רפואי הקשור לעורק הכלילי בצידו הימני של הלב

A'_1	A'_2	A'_3	A'_4	A'_5
2	3	5	9	11
1	3	6	9	10
1	4	5	9	11
1	4	5	7	11
1	3	6	8	10

Table 3: Mapped medical data to items

איור 5.1.1 ב' [47]

באיור 5.1.1 ב' ניתן את בסיס הנתונים ממש, כאשר בבסיס הנתונים מאוכסנים המספרים הסידוריים של כלל התכונות שהוגדרו ב 5.1.1 א'.

יש לציין כי לא כל החוקים מעניינים אותנו, למרות שהם נכונים. במקרה דנן אופי החוק שמעניין אותנו, יחפש קשר בין מאפייניו של הנבדק לבין המצב של הלב שלו (LAD, RCA). לדוג':

Age > 70 & Smoke = Y & Gender = M → RCA > 50

הינו חוק מעניין מכיוון שהוא מוצא קשר בין מאפיינים של נבדקים לבין רמת המחלה שלהם. כך נקבל פרופיל של חולה לב עם בעיה ב RCA.

לעומת זאת , חוקים מהצורה הבאה:

Age > 70 → Smoke = Y

LAD > 70 & RCA > 50 → Smoke = Y

אינם רלוונטיים כי מכיוון שהם לא תורמים שום מידע רפואי נחוץ.

מאידך,

LAD >70 → RCA >50

למרות שחוק זה הינו נכון, גם הוא לא רלוונטי מכיוון שהוא טריוויאלי, כאשר קיימת בעיה באזור אחד בלב ישנו סיכוי סביר שהבעיה קיימת גם בצד המקביל.

כמו כן, החוקרים טוענים ששימוש בחוק עם יותר מחמישה משתנים מאט ומסרבל את תהליך הכרייה, למרות שלעיתים המידע בחוק זה הינו נחוץ ונכון.

סוגיות שהועלו בתהליך הכרייה:

- **מיקום הנתונים** - קיימת חשיבות למיקומם של המאפיינים בחוקים. אומנם אין קשר בין רמת התמיכה למיקום המאפיין בחוק, שכן בהינתן חוק תדיר : x → y עלינו לומר כי גם X U Y הינו תדיר. מאידך המיקום של המאפיין בחוק משפיע על מידת העניין שיש לנו בחוק ועל רמת הביטחון של החוק הנ"ל. לכן יש חשיבות להגדרה מראש של חוקים המגבילים את מיקומם של המאפיינים בחוקי הכרייה לצד מסוים של החוק.
- **גודל החוק**- חוקים שמכילים מספר רב של מאפיינים מייצרים מספר חוקים רב וקשה לפיענוח, כמו כן מספר רב של מאפיינים מאט בצורה ישירה את תהליך הכרייה (עקב

הקשר המעריכי בין זמן הריצה לגודל בסיס הנתונים באפריורי). לכן קיים צורך ממשי בהגדרת סף למספר המאפיינים בחוק שברצוננו לכרות.

- **חוקים שאינם רלוונטיים** – כפי שצוין לעיל קיימים חוקים בעלי משמעות טריוויאלית, או חוקים שאינם תורמים כל מידע הנצרך למחקר. בהינתן קבוצה המכילה מאפיינים שאינם מעניינים, נוכל לומר כי גם הקבוצה המכילה אותה מכילה מאפיינים שאינם מעניינים. לכן נוכל להגדיר מראש רשימה של מאפיינים ששילובם בחוק יהפוך אותו ללא מעניין. וכך חוקים אלו יסוננו.
- **רמת תמיכה נמוכה** – יש להריץ את האלגוריתם ברמת התמיכה המינימלית הנמוכה ביותר. בכדי למנוע הרצות חוזרות ונשנות של האלגוריתם בכדי להוסיף עוד חוקים לתוצאות. ניתן גם לשקול הרצה של האלגוריתם ללא שימוש במאפיין התמיכה כלל אלא בשאר המאפיינים שהוזכרו לעיל, ובנוסף ניתן להקטין את מרחב החיפוש ע״י סינון חוקים שמוכלים רק בתנועה אחת בבסיס הנתונים.
- **מידע רועש** – האלגוריתם צריך להימנע מלכרות חוקים בעלי מאפיינים חסרים מכיוון שהם אינם נכונים. מאידך קיימת חשיבות לתנועות אלו למרות שהן פגומות בעיקר לצורכי מציאת שגיאות בבסיס הנתונים.
- **תמיכה מקסימלית -** מכיוון שבסיס הנתונים שבו אנו עושים שימוש הינו בסיס נתונים רב מימדי, ייתכנו חוקים רבים שיהיו בעלי תמיכה גבוהה אך בעלי מספר מאפיינים קטן. בעיה זו אופיינית בעיקר בכריית מאפיינים נומריים. והיא נובעת מהמספר הגדול של אפשריות שנוצרו מחלוקת התחומים של המאפיין הנומרי. לכן יוגדר גם חסם עליון לתמיכה.

בכדי להתייחס לכל הסוגיות שהועלו להלן, מוצג במאמר אלגוריתם כרייה המהווה שיפור לאפריורי. השיפור אינו ייחודי לכריית מידע רפואי ויכול לשרת אותנו בכל סוג דומה של בעיות בנתונים. למרות זאת, הצורך בשיפור עלה דווקא מתוך עולם התוכן הרפואי מכיוון שדווקא שם הבעיות הללו הינן משמעותיות ביותר.

האלגוריתם יעשה שימוש מובנה במגבלות על החוקים. סוגי המגבלות יהיו:

- מגבלה על מיקום מאפיין בחוק (שמאל, ימין ,שניהם)
- מגבלה על שייכות לקבוצה – לכל מאפיין תוגדר רשימת קבוצות שאליהן הוא יכול להשתייך / לא יכול להשתייך. שייכות לקבוצה מסוימת גוררת בהכרח הפיכת חוק ללא רלוונטי אם מצוי בחוק הזה מאפיין מסויים. כעת ישנה משמעות וחשיבות למאפיינים המצויים יחד עם המאפיין הנבדק[75].

בהינתן קבוצת עצמים $X = \{i_1,i_2,\ldots,i_k\}$ ניתן לומר כי החוק הוא ״חוק ימני מעניין״ אם מתקיים:

$$\forall i_j \in X \;\; ac(i_j) \neq 2$$

[75] ניתן לומר כי בעצם השתייכות של שני מאפיינים לאותה קבוצה קובעת כי הם לעולם לא יוכלו להיות ביחד באף חוק. ומבטיחה מחקר נפרד של כל אחד מהם ללא השפעות פנימיות של אחד על השני.

כאשר ac הינה פונקציה הממפה עצם למגבלה על מיקומו (ימין,שמאל שניהם, ובצורה מספרית (1,2,3).

ניתן לומר אותו דבר על "חוק שמאלי מעניין" אם מתקיים:

$$\forall i_j \in X \;\; ac(i_j) \neq 1$$

כמו כן, ניתן לומר כי X הינו מעניין ברמת הקבוצה אם מתקיים:

$$\forall i_j \forall i_{j'} \in X \quad i_j \neq i_{j'} \Rightarrow group(i_j) \neq group(i_{j'})$$

כלומר אין 2 עצמים המשתייכים לאותה קבוצה. תנאי זה הינו הכרחי מכיוון שזו הנחת היסוד של המגבלה הנ"ל. האלגוריתם לא מטפל (מטעמי פשטות) במקרה שמאפיין שייך ליותר מקבוצה אחת. הפונקציה group תחזיר מספר חיובי אם העצם מוגבל בשייכות לקבוצה ו 0 אם לא.

במהלך האלגוריתם התווספו (על אפריורי) הבדיקות והשלבים הבאים:

בהינתן תת קבוצה המועמדת להיות תדירה: $X = \{i_1, i_2, \ldots, i_k\}$ נבדוק האם:

$$group(i_j) \neq group(i_{j'})$$

$$group(i_j) * group(i_{j'}) > 0 \quad \left| j \neq j', \;\; 1 \leq j, j' \leq k \right.$$

בדיקות אלו מוודאות כי כל עצם שייך לקבוצה אחת בלבד וכי לכל עצם קיימת קבוצה שהוא שייך אליה. במידה וכן אזי נוכל להמשיך בבדיקה של האם X הינה תת קבוצה המועמדת להיות תדירה.

אם שני מאפיינים שייכים לאותה קבוצה אזי החוק שמכיל את שניהם אינו רלוונטי, לכן עלינו מראש לוודא שאין מאפיינים שמצויים באותה קבוצה.

כמו כן, במהלך בדיקה האם תמיכה של תת קבוצה גדולה מהתמיכה המינימלית נשלב גם את התמיכה המקסימלית שהוגדרה לעיל ונבדוק כי התמיכה שלנו לא גדולה מהתמיכה המקסימלית.

בשלב של יצירת החוקים נוודא את חוקיות החוק בהתייחס למיקום המאפיינים ע"י הפונקציה ac שהוגדרה לעיל.

נציין כי מהות השיפור שהוצג הינה בנכונות התוצאות ובטיוב החוקים שמתקבלים. לא מדובר על שיפור התהליך מבחינת סיבוכיות זמן ריצה, מקום וכדו'.

מבחינת **סיבוכיות זמן** הבדיקה הנ"ל אינה מוסיפה דבר ברמת סדרי גודל לסיבוכיותו של אפריורי. מכיוון שמדובר רק על בדיקה פנימית בתוך האלגוריתם.

כמו כן, מבחינת **סיבוכיות מקום** לא מבצעים שינויים במבני הנתונים הקיימים לכן גם כאן אין שינוי מהסיבוכיות של אפריורי.

5.1.2 כריית חוקי הקשר במידע רפואי

שימוש רפואי באלגוריתמי כריית חוקי הקשר פחות נפוץ עקב הקושי בהפיכת חוקי ההקשר למודל שימושי. הסיבות לכך הינן:

- לרוב, חלק גדול מהחוקים שמתקבלים אינם רלוונטיים. בכדי לקבל חוקים רלוונטיים עלינו להוריד בצורה משמעותית את התמיכה – מה שפוגע באמינותו של החוק.
- כמות חוקי ההקשר המופקים על ידי אלגוריתמי כריית מידע מסוג חוקי הקשר בכריית מידע בתחום הרפואי על ידי שימוש במדד תמיכה מינימלי נמוך הינו עצום. עקב כך ניתוח חוקי ההקשר ומציאת החוקים המתאימים דורש עבודה איטית וארוכה של מומחה מתחום המידע, ובחלק מהמקרים אינה אפשרית במסגרת מגבלות המשאבים המוקצים למחקר.
- לעיתים קרובות כמות החוקים המתקבלת הינה גדולה מאוד ומצריכה שימוש בשיטות צמצום שונות / שימוש במומחה לצורך סינון התוצאות. נמנה לדוגמא מספר שיטת צמצום קיימות:
  - יצירת קריטריונים לאישורו של חוק. הקריטריון יכול להיות מיקום של מאפיין בחוק (ימין / שמאל), כמות מאפיינים וכדי.
  - שימוש באלגוריתם שייתן ציון לכל חוק בזמן שהוא נותר על סמך קריטריונים שייקבעו מראש וכך ניתן יהיה לדעת מה חשיבותו של החוק הנ"ל.

בחלק זה יובאו לדוגמא מספר מחקרים העושים שימוש רפואי בכריית חוקי הקשר. לא נעמיק בכל אחד מהמחקרים, נציג את הרעיון העיקרי הכולל שימוש בחוקי הקשר לכריית מידע רפואי.

<u>שימוש בחוקי הקשר לצורך חיזוי מחלות לב</u>

מטרות המחקר[49] כללו אימות של חוקים קיימים שהתקבלו ע"י מערכת מומחה ממוחשבת וגילוי חוקים חדשים. החוקים הללו אמורים למצוא קשר בין מידע רפואי גולמי של החולה לבין מצבו הבריאותי של לב החולה – בהתמקד במצב העורקים הכליליים.

בשלב ההתחלתי השתמשו החוקרים בבסיס נתונים רפואי קיים המכיל רשומות מידע לגבי החולים. כל רשומה בבסיס הנתונים מכילה מידע רלוונטי לגבי מצבו של החולה : מידע אישי - גיל וכדו'. ומדדים שונים – משקל, גובה לחץ דם וכדו'. כמו כן גם מחלות רקע מצורפות לרשומה הנ"ל. בנוסף מצורפת לרשומה האבחנה המלאה של הרופא המטפל. נוסיף כי לרשומה מצורפים גם תאריכים שונים של בדיקות ואת התוצאות שלהן. כמו כן מצורפות תמונות (בצורה בינארית) של אזורים שונים בלב (עבור כל מטופל).

בשלב העיבוד המקדים נבחרו מתוך בסיס הנתונים 25 שדות מאפיינים רפואיים שייעשה בהם שימוש בתהליך הכרייה. המאפיינים נבחנו ולכל מאפיין הוגדר סוג המידע המוכל בו, טווח ערכים וכדו'. המאפיינים חולקו לשלושה קבוצות:

- P – מאפיינים רפואיים של איזור הלב
- R – גורמי סיכון למחלות לב (עישון, גנטיקה וכדו')
- D - רמת התחלואה של הלב.

ניתן בעצם לומר כי מטרת המחקר היא למצוא קשר בין מאפיינים המשתייכים ל P ו R אל D. כלומר יכולת לחזות את מצבו הרפואי של הלב על סמך מאפיינים ידועים מראש.
כחלק מהעיבוד המקדים היה צורך להמיר את רשומות המידע הרפואי לפורמט המאפשר כריית חוקי הקשר. סוגי המידע בבסיס הנתונים היו : קטיגוריאלים, מספריים, ערכי זמן, ותמונות. בכדי לפשט את תהליך הכרייה הוחלט להתייחס לכל השדות כמספריים או קטיגוריאלים. לצורך כך נעשה שימוש באלגוריתם מיפוי והמרה[76] שהמיר את כלל המאפיינים לסוגים הנ"ל.
בכדי לוודא את נכונותו הרפואית של החוק הוחלט כי אנו מעוניינים אך ורק בחוקים שמכילים לפחות שני חולים. כך נוכל בוודאות לסנן חוקים שאולי תקפים מבחינת תמיכה ורמת ביטחון אך לא תקפים מבחינה רפואית. בכדי להפוך את זמן הריצה ליעיל ואת כמות החוקים לכזו שניתן להשתלט עליה, הוחלט על שימוש במגבלות שונות במהלך תהליך הכרייה בכדי להבטיח שהמידע שיתקבל אכן יהיה בעל ערך. לדוגמא : על סמך ידע מוקדם יכלו החוקרים להגביל מיקומים של מאפיינים בחוקים המתקבלים. חוקים שלא עמדו בתנאי הנ"ל סוננו ולא נעשה בהם שימוש.
דוגמא נוספת למגבלה, הגבלת קבוצות למאפיינים : כלומר ישנם מאפיינים שחייבים להופיע ביחד באותו חוק. לאחר שהתקבלו התוצאות התבצעו מספר צעדי סינון שהקטינו את מספר החוקים שהתקבלו : ע"י הסרה של חוקים ידועים / לא רלוונטיים וכדו'. לאחר מכן חולקו החוקים לתתי קבוצות המייצגות קשרים שברצוננו לחקור. מכל תת קבוצה נבחרו רק מספר חוקים בעלי מאפייני התמיכה / רמת ביטחון הגבוהים ביותר. כל החוקים שנותרו עברו סינון ואישור ע"י מומחים בתחום. כותבי המאמר מציינים כי חלק מהחוקים שהתקבלו היו טריוויאליים בעוד שכמה אחרים היו מעניינים ולא ידועים[77]. החוקים שהתקבלו שימשו להעשרה של מערכת מומחה קיימת בכדי להקל על תהליך זיהוי המחלה ואבחונם של חולים נוספים.

[76] לא נרחיב על אלגוריתם המיפוי במסגרת זו

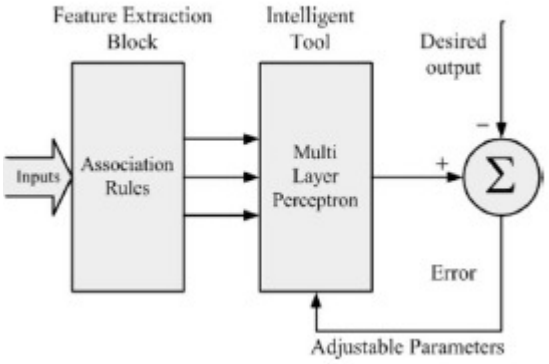
[77] לא נתמקד בחוקים שהתקבלו, חוקים אלו שייכים למרחב הבעיה ולא נדון בהם במסגרת זו.

<u>שימוש בחוקי הקשר לצורך חקר מחלת הסרטן</u>

קיימות שיטות שונות שאינן דווקא משתמשות בחוקי הקשר לכריית מידע הקשורות למחלת הסרטן [42][41]. (חלקן משתמש בכריית מידע רפואי וחלקן אף בכריית תמונות). אנו נתמקד בחלק זה בשימוש בחוקי ההקשר.

חקר הסרטן בשימוש בחוקי הקשר מתמקד הן ברמה התאית והן ברמת האיברים בגוף. קיימים שימושים שונים לחוקי הקשר בתחום הנ״ל. בחלק מהמקרים נעשה שימוש בחוקי הקשר בכדי לצמצם את מרחב המידע של הבעיה לפני הפעלת שיטה אחרת[43].

ב [43] מוצג שימוש של כריית חוקי הקשר ככלי לצמצום נתוני פלט למערכת מומחה מבוססת רשת נוירונים. (ראה איור של המערכת ב 5.1.2 א׳)



איור 5.1.2 א׳ [47]

במקרה שלנו, נעשה שימוש בכריית חוקי הקשר בכדי לדעת אילו מאפיינים הינם חשובים באמת בחוק ואלו מיותרים. לאחר שהתגלו המאפיינים הרלוונטים, הכניסו החוקרים רק אותם כפרמטרים לרשת הנוירונים. בכך, צומצם מספר הקלטים של הרשת וקטן כח החישוב הנדרש לעיבוד המידע – מה ששיפר בצורה משמעותית את ביצועי המערכת.

ב [50] נעשה שימוש בכריית חוקי הקשר בכדי לאפיין קשרים בין מאפיינים של צילומי CT של סרטן הריאה. הוגדרו כ 37 מאפיינים שניתן להבחין בהם בצילום וכך ניתן היה לייצג כל צילום CT ע״י רשומה טבלאית. החל משלב זה ניתן היה להפעיל את אלגוריתם הכריה בצורה רגילה. בסופו של תהליך התגלו כ 123 חוקים המשקפים קשרים בין מאפיינים שונים שהוגדרו מראש בצילום.

כותבי המאמר [46] טוענים כי במקרים רבים קיימים גורמים למחלת הסרטן. לעיתים ניתן אף למנוע את המחלה ע״י הימנעות מהגורמים הללו. מחקר מעמיק ומקדים על גורמי המחלה יכול לסייע רבות במניעת מחלות אצל אנשים ובצמצום החולים במחלת הסרטן. קיימים מספר סוגי סרטן שמחקרים מצאו עבורם גורמים שמגדילים בצורה משמעותית את הסיכוי לחלות בסרטן. החיסרון הבולט במחקרים אלו הוא אי היכולת לברור את המוץ מן התבן – ולהבחין בין הגורמים העיקריים למשנים. בדיוק לצורך זה ניתן להשתמש בחוקי ההקשר. בעזרת חוקי ההקשר ניתן לנתח בצורה מדויקת מחקרים בתחום חקר הסרטן ואת הסיבות המשמעותיות לכל אחת

מהמחלות. ניתן איפה לומר כי מטרת המחקר היא מציאתם של הגורמים העיקריים לכל אחד מסוגי הסרטן שנסקרו. וזאת, כפי שציינו, מכיוון שבמחקרים הקיימים היום ישנם גורמים רבים מדי לכל אחד מסוגי הסרטן.

לצורך המחקר השתמשו החוקרים במאגר של נתוני מניעה[78] מסרטן שנאספו במשך השנים ע"י חוקרים שונים. בסיס הנתונים הכיל מידע לגבי כל אחד מסוגי הסרטן שנבדקו במחקר.

נניח כי D הינו אוסף מידע של n חולי סרטן : $D = \{C_1,C_2,C_3\ldots,C_n\}$.

כמו כן, $\lambda=\{i_1,i_2,..,i_n\}$ מהווה אוסף של גורמי מניעה למחלה. נתון כי $\lambda \supseteq C_i$.

נניח ו X ו Y הינם שני גורמי מניעה כך ש : $\lambda \supset Y \quad \lambda \supset X$ ו $X \cap Y = \acute{O}$ חוק ההקשר יהיה

מהצורה $X \rightarrow Y$. כמו כן הוחלט שכלל המאפיינים הינם קטיגוריאלים.

תהליך הכרייה יחולק לשני שלבים :

- מציאת כל הצירופים של גורמי מניעה עבור סרטן מסוג מסוים, על צירופים אלו להיות בעלי תמיכה גדולה יותר מהתמיכה המינימילת שהוגדרה.
- שימוש בגורמי המניעה התדירים בכדי לחולל חוקים נכונים.

הפעלת אלגוריתם הכרייה על כל אחד מהגורמים יצרה חוקים המגדירים את הגורמים הרלוונטיים לכל סוג סרטן. שימוש בחוקים אלו יכול לסייע בהפחתה משמעותית של הסיכוי לחלות במחלת הסרטן. לא נפרט את החוקים שהתקבלו במסגרת זו.

[78] נתון מניעה הינו מידע לגבי גורם שיכול למנוע סרטן מסוג מסוים. לדוגמא : המנעות מעישון וכדו'.

6.יישום

בחלק זה יבוצע יישום בתחום כריית מידע בשימוש בחוקי הקשר. היישום מבוסס על בסיס נתונים שנלקח מתוך [38]. בסיס הנתונים מתאר סוגים שונים של מיקומי גידולים ואת מיקומי הגרורות בהתאם.

המחקר יענה על השאלות הבאות:

1. האם קיים קשר בין מיקומי הגידולים השונים (בינם לבין עצמם)? כלומר האם העובדה שמצוי גידול במקום מסוים. תעיד על הימצאות גידול במקום אחר?
2. האם קיים קשר בין מיקומי גרורות הסרטן (בינן לבין עצמן), כלומר, האם הימצאותה של גרורה במקום אחד תעיד על הימצאות גרורה במקום אחר ?
3. האם קיים קשר בין מיקומי הגידולים למיקומי הגרורות? כלומר, האם עובדת ההימצאות של גרורה / גידול באיבר X תגרור בהכרח הימצאות באיבר הסמוך.

בחלק זה נעשה שימוש בתוכנת SPSS - Clementine. תוכנה זו משמשת לסוגים שונים של כריית מידע, ביניהם, כריית חוקי הקשר.

התוכנה מאפשרת שימוש במגוון רחב של אפשרויות מידול בנושאים: כריית מידע, בינה מלאכותית, וסטטיסטיקה. העקרון הבסיסי המאפיין את הכלי הנבחר הינו שימוש מהיר במודלים מסובכים על מידע עסקי, ללא השקעת זמן במימוש אלגוריתמים סבוכים לכריית מידע.

בתחום כריית המידע החבילה תומכת במודולים הבאים:

- מודולים לסיווג (בניה של עצי החלטה על סמך בסיס נתונים קיים)
- מודול לחלוקה לקבוצות (סגמנטציה)
- מודול לחוקי הקשר

בכדי לבנות מודל בשימוש בתוכנה יש להתייחס לשלושת הפרמטרים הבאים:

1. מקור המידע – המקור שממנו נלקח המידע (בסיס הנתונים) שאותו ברצוננו לכרות.
2. סיווג השדות במידע – הגדרות המתייחסות לסוג של כל שדה בבסיס הנתונים. לדוג' – טווח ערכים אפשרי, סוג השדה וכדו'.
3. מודל – המודל שימומש בעזרת המידע שהושג.

תחום העיסוק של עבודה זו הינו חוקי ההקשר ולכן נתמקד בו. בתת תחום זה של מודולים ממומש בכלי SPSS האלגוריתם Carma (שנסקר כבר קודם) ואפריורי (שגם נסקר קודם לכן).

אלגוריתם נוסף שממומש בתוכנה הוא אלגוריתם GRI(Generalized Rule Induction).

לא נרחיב את ההסבר על האלגוריתם הנ"ל מכיוון שהוא לא נסקר בעבודה.

6.1 סקירת הבעיה הרפואית

האיברים והרקמות בגוף עשויים מתאים. למרות שתאים בחלקי גוף שונים נראים ומתפקדים באופן שונה זה מזה, הרי שכל התאים מתחדשים באותה צורה כלומר, על-ידי חלוקה. החלוקה בדרך-כלל סדירה ומבוקרת, אולם אם יוצא התהליך מהבקרה, ימשיכו התאים להתחלק ללא צורך. כתוצאה מכך, נוצר גוש תאים הקרוי גידול. גידולים עשויים להיות **שפירים** או **ממאירים.**

גידול שפיר הינו גידול בו התאים אינם מתפשטים לחלקי גוף אחרים, אולם, אם הם ממשיכים לגדול באתר המקורי הם עלולים לגרום ללחץ על האיברים סביבם.

גידול ממאיר בנוי מתאים בעלי יכולת להתפשט מחוץ לאתר המקורי של הגידול, וללא טיפול הם עלולים לחדור לרקמות בסביבה ולהרסן.

תאים הניתקים מן הגידול הראשוני מתפשטים לאיברים אחרים בגוף, באמצעות זרם הדם או הלימפה. באתרים החדשים הם עשויים להמשיך ולהתחלק וליצור גושים חדשים הקרויים "גידולים משניים", או "גרורות".

קביעת סוג הגידול – ממאיר או שפיר – נעשית על-ידי בדיקות מיקרוסקופיות של תאי הגידול.

לקיחת דגימת רקמה מאיזור הגידול לשם בדיקה נעשית על-ידי ביופסיה – הוצאת רקמה בניתוח.

חקר הסרטן כיום מתמקד בהבנת גורמי המחלה ובמניעתה. שכיחותה הגבוהה של המחלה והקשיים בטיפול בה הובילו לכמות גדולה מאוד של מחקרים בתחום. בשנים האחרונות החל להתפתח גם השימוש בכריית מידע לחקר המחלה. תחום כריית המידע יכול לאפשר אבחון ממחושב של המחלה, חיזוי גורמים למחלה, ועוד תחומים רבים. חלק מהשימושים כולל כריית מידע מתוך נתוני בדיקות,צילומי רנטגן, נתונים של חולים, וכדו'. אנו עדים בזמן האחרון לכמות גדולה מבעבר של מחקרים העוסקים בתחום זה מתוך רצון למצוא דרכים חדשות להתמודד עם המחלה על סמך הנתונים הרבים הקיימים בהווה. במחקר זה ברצוננו לחפש קשרים בין מיקומים של גידולים וגרורותיהם. כלומר ננסה למצוא קשר בין מיקומי גידולים / מיקומי גרורות / מיקומי גרורות וגידולים.

6.2 איסוף ועיבוד המידע

בסיס הנתונים מכיל 339 רשומות. כל רשומה מתארת את סוג הגידול ואת המיקומים של הגרורות. בבסיס הנתונים קיימים 18 מאפיינים כולל המאפיין של סוג הגידול.

לא היה צורך ממשי לעבד את המידע וזאת מכיוון שהמידע הגיע מעובד ישירות מבסיס הנתונים.

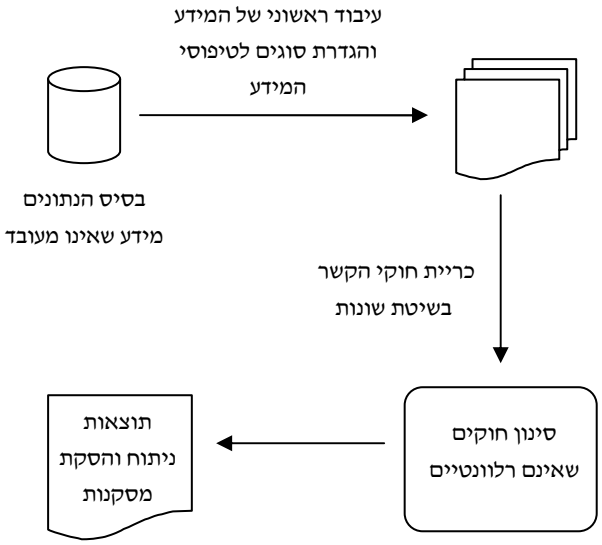
למרות זאת, בוצע עיבוד מקדים שכלל המרות של ייצוגים של שדות שונים (לדוגמא : 1 ו 2 ל True ו False). דוגמא נוספת להמרה שבוצעה הייתה המרתו של השדה class. שדה זה הינו שדה קטגוריאלי (כלומר ערכו היה אחד מתוך רשימה של 22 ערכים). שדה זה הומר ל 22 שדות בוליאניים שציינו עבור כל רשומה בבסיס הנתונים האם היא שייכת לקבוצה הנ"ל או לא. כמובן שלא תיתכן הימצאות של רשומה השייכת לשני קבוצות.

בסיס הנתונים הועתק לקובץ Excel ומשם נקרא ישירות דרך תוכנת ה SPSS.

מבנה בסיס הנתונים (הגדרת השדות מופיע בנספח 9.1 א')

6.3 תהליך הכרייה

נציג להלן תרשים זרימה גנרי ברמת העל של תהליך הכרייה שבוצע בפועל.



איור6.3 א' – תהליך כריית המידע

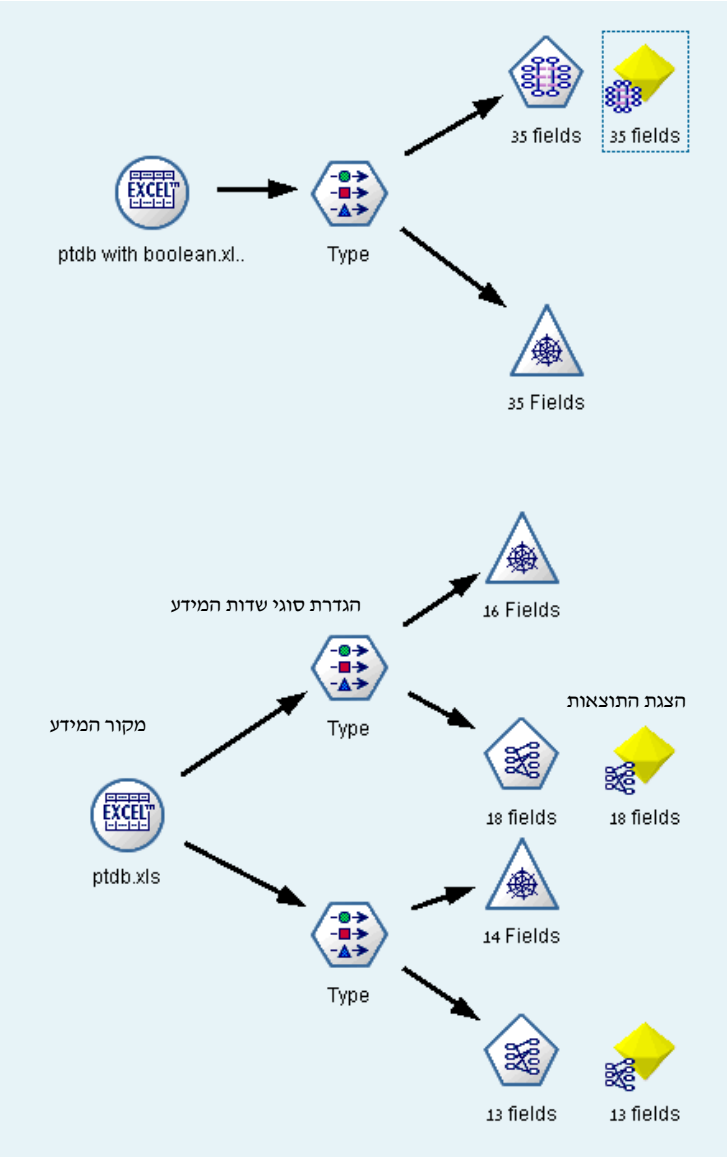
6.3.1 עיבוד מקדים

תהליך הכרייה כולל שימוש בשני אלגוריתמים שונים : Apriori ו Carma. מכיוון שלכל אחד מהאלגוריתמים דרישות שונות ביחס למידע המתקבל, בוצעו התאמות לנתונים בצורה הבאה:

- עבור אפריורי - לא נדרשו התאמות מיוחדות והנתונים עובדו כפי שהתקבלו מבסיס הנתונים.
- עבור Carma - האלגוריתם דרש כי כלל המאפיינים יהיו בוליאניים. לכן נדרש להמיר את המאפיין Class, המייצג את סוג הגידול (בעל 22 ערכים) ל22 משתנים שונים שכל אחד מהם יהיה בעל ערך בוליאני. לדוגמא: במקום משתנה 1 ששמו class, יתווספו לבסיס הנתונים עוד 22 עמודות (משתנים) הערך של כל עמודה ייצג בצורה בוליאנית האם הגידול שייך למחלקה הזו. לדוגמא: עמודה 15, הערך הבוליאני של העמודה יציין האם הגידול שייך / לא שייך לעמודה הנ"ל. באם הגידול שייך לעמודה הערך הבוליאני יהיה True, אחרת הערך יהיה False. בצורה כזו נייצג את המשתנה המספרי Class ע"י 22 משתנים בוליאניים (כמספר הערכים האפשרי).

בבסיס השימוש בתוכנת SPSS עומד העיקרון שהמידע עובר בעצם סוג של זרימה : ממקור המידע עד לפלט הניתן להבנה אנושית(חוקים). בדרך המידע עובר "בתחנות" שונות בהן נעשה עליו עיבוד כפי שנדרש. מכיוון שההתאמות והשינויים במידע שכל אחד מהאלגוריתמים דורש היו שונים, היה צורך להגדיר מסלולי זרימה שונים של המידע עבור כל אלגוריתם.

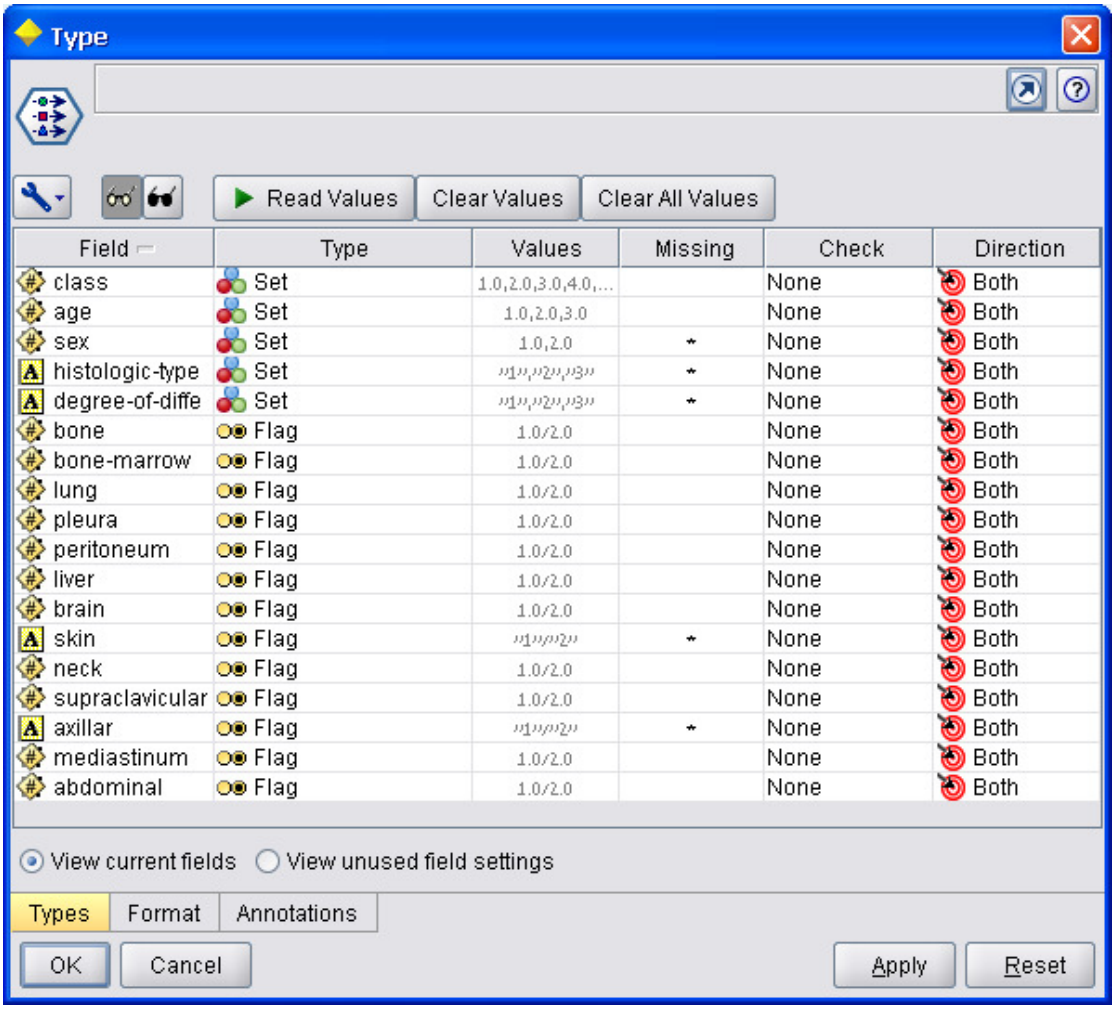
הוגדרו בתוכנה שני מסלולים שונים לשימוש השוואתי בשני האלגוריתמים(ראה איור 6.3.1 א׳).
ניתן לראות באיור 6.3.1 א׳ דיאגרמה של התהליך הכולל עבור שני האלגוריתמים.
במקטע העליון ניתן לראות קלט של קובץ נתונים המכיל את המידע המומר לצורה בוליאנית.
כלומר של האלגוריתם Carma. מתחתיו מוצג המידע שעליו הופעל אפריורי.
עבור שני האלגוריתמים המידע עובר עיבוד בתוך הצומת Type (על העיבוד הנ״ל נדבר בהמשך).
לאחר מכן המידע עובר לצומת תצוגתית המציגה את המידע בתצוגות שונות (גם עליהן נרחיב בהמשך).



איור6.3.1 א׳ – זרימת המידע בתוכנת SPSS

כאמור, לאחר הקריאה מבסיס הנתונים (קובץ ה Excel), הנתונים עוברים לצומת Type בצומת זו מוגדרים סוגי המידע עבור כל שדה בבסיס הנתונים (איור 6.3.1 ב׳). באיור 6.3.1 ב׳ אנו מגדירים עבור כל סוג שדה (מאפיין) של בסיס הנתונים את סוגו (מחרוזת, מספר בוליאני וכדו׳) ואת

הערכים האפשריים שלו (אם ישנם). פעולה זו מתבצעת עבור כלל השדות של בסיס הנתונים הנידון (מפורטים בנספח 9.1 א').



איור 6.3.1 ב' – הגדרת סוגי מידע עבור כל שדה

מלבד היכולת להגדיר עבור כל שדה בבסיס הנתונים את סוג המידע שהוא מכיל ואת הערכים האפשריים, ניתן גם להגדיר עבור כל שדה את המיקום שלו בחוק שאנו מצפים לקבל כתוצאה. כלומר עלינו לסמן האם המאפיין הנ"ל יכול להיות בצד ימין של החוק (גורם) או בצד שמאל (תוצאה) או בשניהם.

בכדי לקבל מגוון חוקים גדול יותר הוגדרו עבור כל אלגוריתם שני מצבי כריית חוקים:

- כל מאפיין יכול להיות בכל אחד מצדדי החוק
- לכל מאפיין הוגדר מיקום מדויק בחוק (גורם, תוצאה)

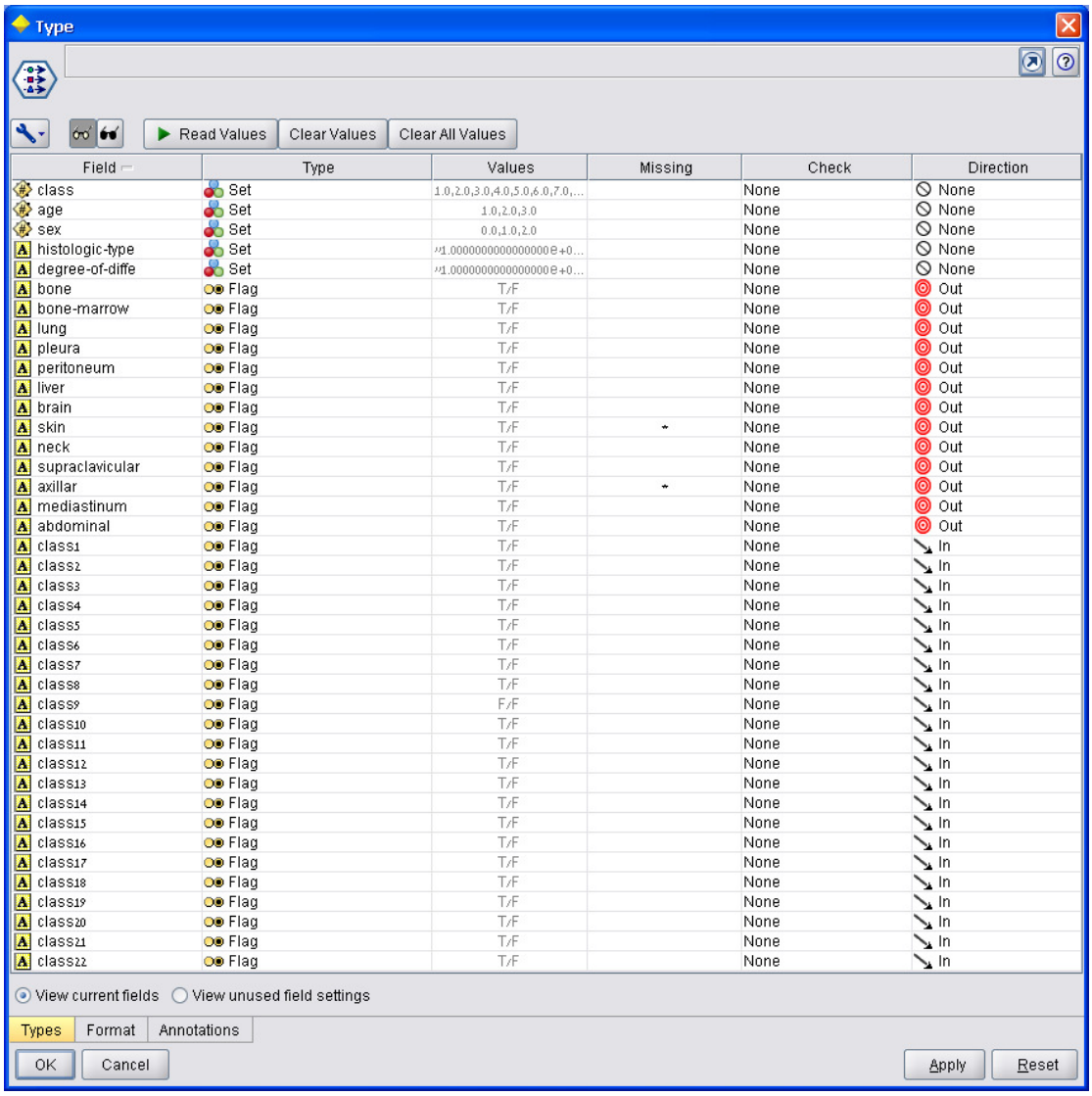
שני מצבים אלו נבדקו עבור שני האלגוריתמים בכדי למצות את תהליך הכרייה ולמצוא חוקים גם אם לא ציפינו למוצאם.

בפועל תהליך זה בוצע ע"י הפיצול של זרימת המידע (ראה איור 6.3.1 א') ממקור המידע לשני צמתי Type שונים. בצומת אחד הוגדרה חשיבות למיקומם של השדות בחוק ובשני לא.

נציין כי בבסיס הנתונים נמצא שדה שלא רלוונטי לכרייה histlogic_type, השדה מתאר את סוג הדגימה שנלקחה מהתא הסרטני. מכיוון שעבדוה זו אינה מתייחסת לסוגי הדגימות אלא למיקומים של הגרורות והגידולים הוחלט על התעלמות משדה זה במהלך כריית המידע.

פעולה זו נעשית בשימוש במסך המופיע באיור 6.3.1 ב' – והגדרה של ה Direction עבור השדה הנ"ל ל None.

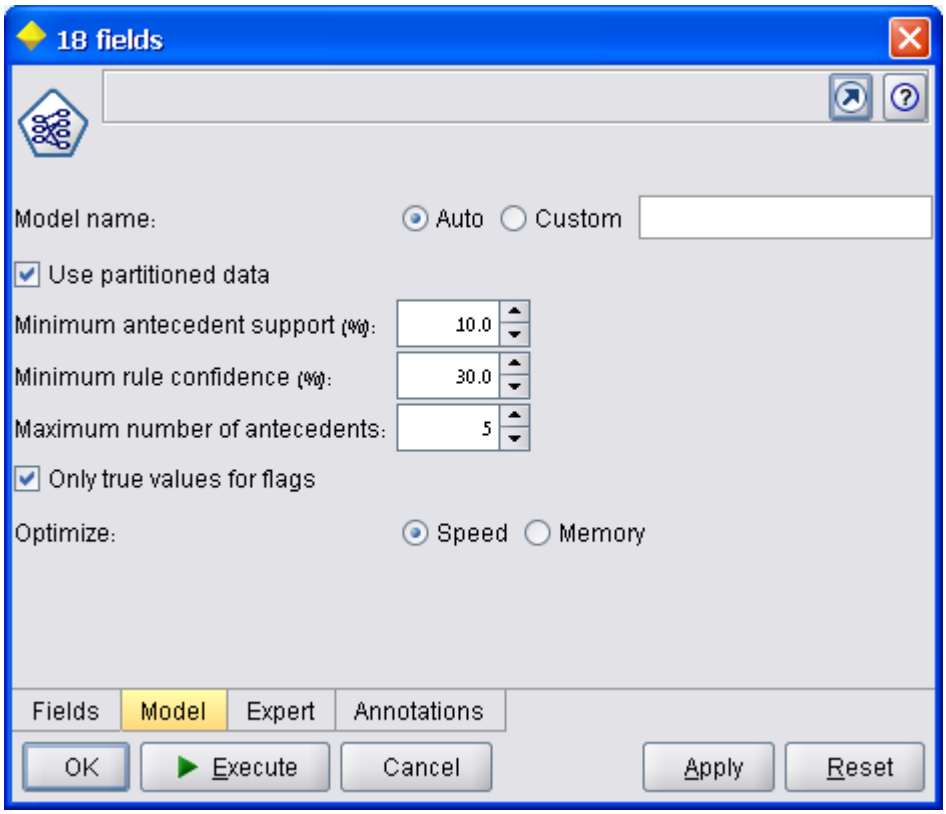
באיור 6.3.1 ג' ניתן לראות את מסך ההגדרה של המידע והשדות עבור Carma – שם הוגדר מיקום לכל אחד מהשדות בבסיס הנתונים. ניתן לשים לב לפיצול של השדה הקטיגוריאלי class ל22 שדות בוליאניים



איור 6.3.1 ג' – הגדרת כיווניות לכל שדה

ההגדרות הללו הינן ברמת צומת ה Type ולכן עבור כל בסיס נתונים הוגדרו שני צמתים.
שני האלגוריתמים יודעים להתחשב בערכים חסרים (תופעה נפוצה מאוד בעיקר במידע רפואי).
למרות שקיימות שיטות נוספות להתמודדות עם בעיה זו, בשני המימושים של האלגוריתמים בחרו להשתמש בפיתרון פשטני לבעיה:
בשני האלגוריתמים מתעלמים מערכים חסרים, מצד אחד האלגוריתם יטפל (כלומר יכלול בחישובי התמיכה ובשאר חישובי האלגוריתם) רשומות המכילות שדות בעלי ערכים חסרים אך מאידך לא יתייחס לשום חוק המכיל שדה אחד או יותר של ערכים חסרים. בצורה זו לא יווצרו חוקים המכילים ערכים חסרים.
במקרה כזה יש לציין לתוכנה עבור איזה שדה קיימים ערכים חסרים בבסיס הנתונים. הסימון ייעשה ע"י סימון * בשדה שקיימים עבורו ערכים חסרים. באופן עקרוני הנ"ל נעשה באופן אוטומטי. כלומר התוכנה מזהה שדות כאלו ומסמנת אותם. אך ניתן גם להגדיר שדה ככזה גם אם לא הוזן כך מראש.

לאחר מכן יש להגדיר את הצומת של אלגוריתם הכרייה עצמו. בצומת זו מוגדרים הפרמטרים הרלוונטיים לתהליך הכרייה. ביניהם תמיכה ורמת ביטחון מינימלים.
באיור 6.3.1 ד' ניתן לראות דוגמא למסך הגדרת מאפייני כרייה עבור אלגוריתם הכרייה. מסך ההגדרה דומה עבור שני האלגוריתמים השונים.



איור 6.3.1 ד' – הגדרת משתני הרצה לאלגוריתם

6.4 תוצאות

מפני שמדובר במידע רפואי, ניתן היה לצפות מראש שהתמיכה עבור החוקים תהיה נמוכה. לכן רמת התמיכה שהוגדרה הייתה 10%. במקרים בהם התקבלו מספר נמוך של חוקים בתוצאה. הורדנו את רמת התמיכה לנמוכה יותר. למרות הורדת רמת התמיכה עדיין אנו יכולים לעשות שימוש בפרמטר הביטחון בכדי לוודא את מהימנותו של חוק נתון. ההשלכה המעשית של צעד זה על התוצאות היא: חוקים שיתקבלו כאשר רמת התמיכה נמוכה הינם חוקים שקשה יהיה לטעון ברמת סבירות גבוהה לנכונותם מכיוון שאין לנו מדגם מייצג בגודל מספיק גדול בכדי לאשש בצורה וודאית את נכונותו של החוק. מאידך, נוכל להשתמש בחוקים אלו כסמנים המאותתים לנו על קשר בין מאפיינים של בסיס הנתונים שכדאי לחקור להעמיק ולברר את מהותו.

לפני שנדון בתוצאות בפירוט נרחיב לגבי מספר מדדים שלא נדונו בעבודה אך הם רלוונטיים להבנת התוצאות:

Lift

מדד למדידת חשיבותו / מידת העניין של חוק. מדד זה ביחס לחוק מסוים מוגדר בצורה הבאה:

$$Lift(s) = conf(s)/exp\_conf(s)$$

כלומר מה היחס בין רמת הביטחון של החוק לרמת הביטחון הצפויה שלו.
רמת הביטחון הצפויה של חוק הינה בעצם אחוז המופעים של צד התוצאה של החוק בבסיס הנתונים. כלומר רמת התמיכה שלו.

לדוגמא: עבור החוק $a \rightarrow b$

$$Lift(a \rightarrow b) = conf(a \rightarrow b) / support(b)$$

ערכי המשתנה יאפשרו לנו לאפיין את יחסינו לחוק:

- אם מדד ה Lift גדול מ 1 - ניתן לומר כי חלק הסיבה בחוק (צד שמאל) וחלק התוצאה בחוק (צד ימין) מופיעים בד"כ ביחד יותר מהממוצע הצפוי. ניתן להסיק מכך שלהופעה של גורם הסיבה בחוק יש השפעה חיובית על גורם התוצאה. ולכן נוכל לומר כי הקשר בין גורם הסיבה בחוק לגורם התוצאה אינו סטטיסטי אלא בעל משמעות ואמיתי.
- אם מדד ה Lift קטן מאחד - ניתן להבין כי החוק פחות נפוץ מהממוצע בבסיס הנתונים. כלומר שני חלקי החוק אינם נפוצים כ"כ ביחד בבסיס הנתונים. ניתן להסיק מכך שלהופעה של גורם הסיבה בחוק יש השפעה שלילית על גורם התוצאה.
- אם מדד ה Lift שווה ל1 – נוכל לומר כי אין השפעה כלל של גורם הסיבה על גורם התוצאה. כלומר, אין משמעות להופעה של גורם הסיבה (צד שמאל) בחוק על גורם התוצאה (צד ימין).

השימוש במדד זה יבוא לידי ביטוי בעיקר בהחלטה האם חוק הינו רלוונטי או לא. בעזרת המדד הנ"ל ניתן יהיה להחליט עד כמה החוק משקף נתונים מציאותיים וכך נוכל להחליט האם ברצוננו להתייחס לחוק או לא.

Rule Support

Rule Support מייצג את אחוז הרשומות שבהם יש את שתי צדדי החוק. כלומר עבור החוק A→B : אנו נחפש את אחוז הרשומות בבסיס הנתונים שבצד הסיבה (שמאל) מופיע A ובצד התוצאה (ימין) מופיע B.

מדד זה יהיה משמעותי יותר באלגוריתם Carma, בכל אופן לא ייעשה בו שימוש באפריורי.

הערה כללית לניתוח התוצאות :

בכדי להחליט האם חוק הינו 'חזק' או לא נשתמש בכללים הבאים[79] :

- אם מדד ה lift גדול מ 1.1 אזי החוק יחשב חזק
- אם מדד ה lift גדול מ 1 אך קטן מ1.1 ורמת ה confidence היא מעל 80% החוק גם ייחשב חזק.
- אם מדד ה lift גדול מ 1 אך קטן מ1.1 ורמת ה confidence קטנה מ 80% אך גם התמיכה וגם רמות הביטחון קרובות לרמות של חוקים חזקים אחרים באותו סט תוצאות אזי נחשיב את החוק כחזק.

נשים לב שמדד התמיכה הינו קצת פחות משמעותי במקרה זה מכיוון שבסיס הנתונים קטן יחסית, לכן נייחס משקל קצת יותר רב למדד הביטחון בתהליך החלטה על 'חוזק' של חוק.

6.4.1 אפריורי – חשיבות למיקום השדה בחוק

במצב זה בוצעה הכרייה רק על המאפיינים המייצגים את מקום הגידול ומקום הגרורות. לכל מאפיין הוגדר מיקום רלוונטי בחוק. למקום הגידול – גורם (צד שמאל) למקום הגרורה – תוצאה (צד ימין). נציין כי רק נתונים אלו השתתפו בכרייה. במצב זה עבור נתונים של תמיכה : 10% ורמת ביטחון של 30% התקבלו הנתונים הבאים (כולם חזקים) :

Antecedent (גורם)	Consequent (תוצאה)	ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 1.0	mediastinum = 1.0	15	84	24.779	61.905	15.339	2.281
class = 1.0	bone = 1.0	14	84	24.779	39.286	9.735	1.417
class = 1.0	liver = 1.0	16	84	24.779	34.524	8.555	1.074
class = 5.0	abdominal = 1.0	13	39	11.504	46.154	5.31	1.361
class = 5.0	peritoneum = 1.0	11	39	11.504	41.026	4.72	1.464
class = 5.0	liver = 1.0	12	39	11.504	33.333	3.835	1.037

טבלה 6.4.1 א' – תוצאות אפריורי עבור רמת תמיכה של 10%

בטבלה 6.4.1 א' מוצגות התוצאות שהתקבלו עבור אלגוריתם אפריורי בזמן שהוגדר מיקום (גורם / תוצאה) לכל אחד מהמאפיינים בבסיס הנתונים.

[79] מדובר בקביעה שרירותית בכדי לקבל אפשרות כל שהיא של מדד החלטה לגבי טיב החוק. כמובן שקיימים גם מדדים שמוטמעים באלגוריתם עצמו כמו רמת ביטחון ותמיכה מינימלית.

נדגים איך החוק הראשון ייראה בפועל:

Class = 1 → mediastinum

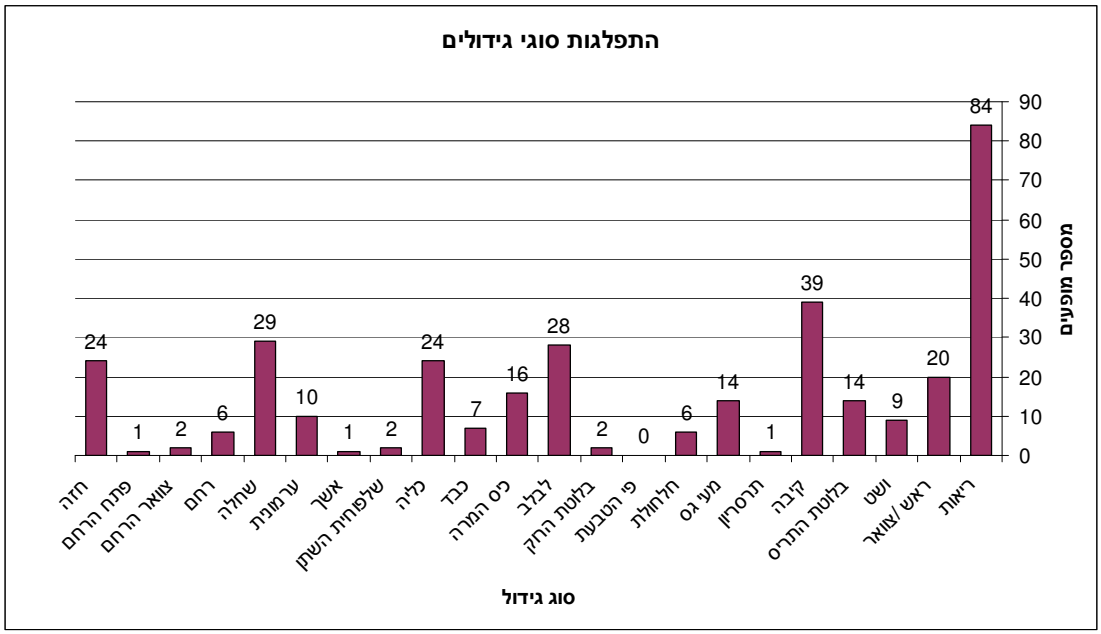
כלומר גידול בריאה (class =1) גורם לגרורות באיזור חלל החזה בסבירות של 60%.

ניתן לשים לב לשני תופעות מעניינות בתוצאות:

1. רוב החוקים שעמדו בתנאי הסף הינם בעלי Lift גבוה מ 1 בצורה משמעותית.
2. בסיס הנתונים אינן מתפלג בצורה אחידה:

מעיון בתוצאות הכרייה ניתן לראות שאין ייצוג כמעט לאף אחד מסוגי הגידולים.

בדיקה קצרה שערכנו בבסיס הנתונים מעלה שהתפלגות סוגי הגידולים בבסיס הנתונים היא כזו (איור 6.4.1 א'):



איור 6.4.1 א' – התפלגות סוגי גידולים בבסיס הנתונים

ציר ה X באיור 6.4.1 א' מייצג את מספרו הסידורי של הגידול וציר Y מייצג את כמות המופעים של הגידול בבסיס הנתונים.

ניתן לראות בצורה ברורה שסוגי גידולים בריאות ובקיבה הינם דומיננטיים מאוד ונפוצים בבסיס הנתונים (84 מופעים ו 39 מופעים בהתאמה). מכיוון שבמקרה זה לא ניתן היה להשיג בסיסי נתונים נוספים בכדי לעבות את כמות המופעים בבסיס הנתונים עלינו למצוא דרך לאפשר גם לגידולים מסוגים שונים (שאינם בריאות ובקיבה) להופיע ברשימת החוקים הסופית.

בכדי לתת הזדמנות ביטוי לשאר סוגי הגידולים הורדנו את רמת התמיכה (ל 5%) אך ללא הורדת רמת הביטחון.

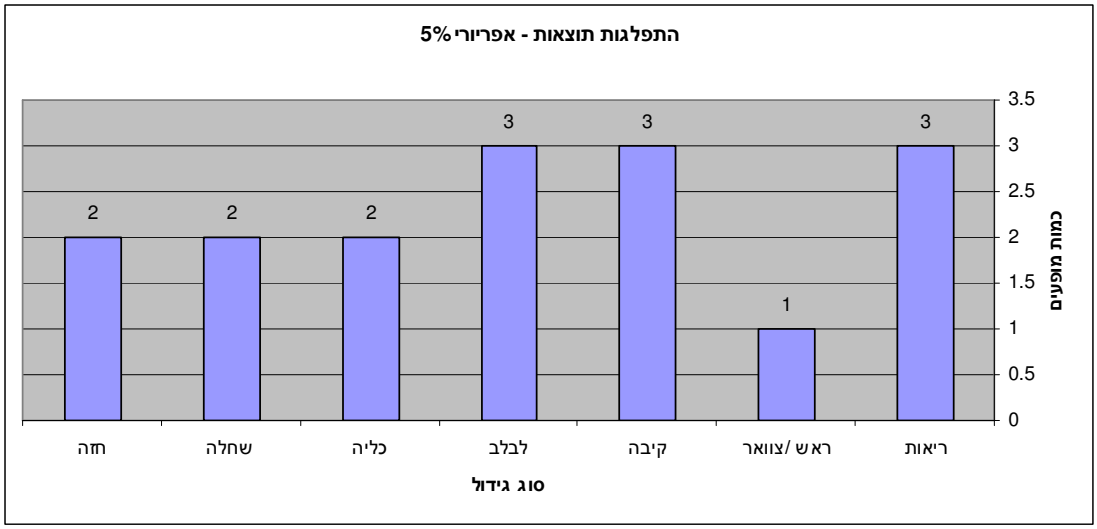
Antecedent (גורם)	Consequent (תוצאה)	ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 1.0	mediastinum = 1.0	15	84	24.779	61.905	15.339	2.281
class = 1.0	bone = 1.0	14	84	24.779	39.286	9.735	1.417
class = 1.0	liver = 1.0	16	84	24.779	34.524	8.555	1.074
class = 5.0	abdominal = 1.0	13	39	11.504	46.154	5.31	1.361
class = 5.0	peritoneum = 1.0	11	39	11.504	41.026	4.72	1.464
class = 5.0	liver = 1.0	12	39	11.504	33.333	3.835	1.037
class = 18.0	peritoneum = 1.0	7	29	8.555	82.759	7.08	2.953
class = 18.0	pleura = 1.0	6	29	8.555	37.931	3.245	1.714
class = 11.0	liver = 1.0	9	28	8.26	75	6.195	2.333
class = 11.0	abdominal = 1.0	10	28	8.26	64.286	5.31	1.895
class = 11.0	peritoneum = 1.0	8	28	8.26	50	4.13	1.784
class = 14.0	bone = 1.0	5	24	7.08	54.167	3.835	1.953
class = 14.0	lung = 1.0	4	24	7.08	50	3.54	2.26
class = 22.0	supraclavicular = 1.0	2	24	7.08	41.667	2.95	2.316
class = 22.0	bone = 1.0	3	24	7.08	41.667	2.95	1.503
class = 2.0	neck = 1.0	1	20	5.9	100	5.9	7.705

טבלה 6.4.1 ב' – תוצאות אפריורי עבור רמת תמיכה 5%

כצפוי, ניתן לראות בטבלה 6.4.1 ב' כיצד מספר החוקים עלה. קיבלנו כעת עוד כמה חוקים ברמת ביטחון גבוהה יותר ממקודם. לכן למרות רמת התמיכה הנמוכה נוכל להתייחס גם לחוקים אלו. מסקנות מהחוקים יוצגו בצורה אחודה בהמשך.

כעת ניתן להבחין בחוקים שלמרות שלא עמדו בתנאי הסף שהוגדרו קודם, עדיין יכולים לעניין אותנו – ולראיה ניתן לראות שגם עבורם מדד ה Lift מגדיר כי מדובר בנתונים משמעותיים. אם היינו ממיינים את החוקים אך ורק על סמך מדד התמיכה ורמת הביטחון ייתכן שהיינו מפספסים את החוקים הללו. (בולט בצורה משמעותית חוק מס' 1).

בכל מקרה כעת רמת ההתפלגות בין החוקים המופיעים בתוצאות הינה משמעותית יותר מקודם ראה איור 6.4.1 ב'.



איור 6.4.1 ב' – התפלגות סוגי גידולים בתוצאות אפריורי 5%

ניצור כעת נגזרת של בסיס הנתונים ללא שתי הקבוצות הנפוצות ונבצע כרייה בנפרד על בסיס הנתונים ללא שתי הקבוצות. שימוש בשיטה זו יאפשר להציג חוקים שלא הוצגו בשיטה הקודמת עקב חוסר הפילוג האחיד של בסיס הנתונים.הרצה חוזרת של האלגוריתם עם פרמטרי תמיכה מינימלית של 10% וביטחון של 30% נתנה את התוצאות הבאות:

Antecedent (גורם)	Consequent (תוצאה)	Rule ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 18.0	pleura = 1.0	3	29	13.426	37.931	5.093	1.821
class = 18.0	peritoneum = 1.0	4	29	13.426	82.759	11.111	2.75
class = 11.0	peritoneum = 1.0	8	28	12.963	50	6.481	1.662
class = 11.0	liver = 1.0	9	28	12.963	75	9.722	2.418
class = 11.0	abdominal = 1.0	10	28	12.963	64.286	8.333	1.827
class = 14.0	bone = 1.0	1	24	11.111	54.167	6.019	2.167
class = 14.0	lung = 1.0	2	24	11.111	50	5.556	2.077
class = 22.0	axillar = 1.0	5	24	11.111	87.5	9.722	7.875
class = 22.0	supraclavicular = 1.0	6	24	11.111	41.667	4.63	3.333
class = 22.0	bone = 1.0	7	24	11.111	41.667	4.63	1.667

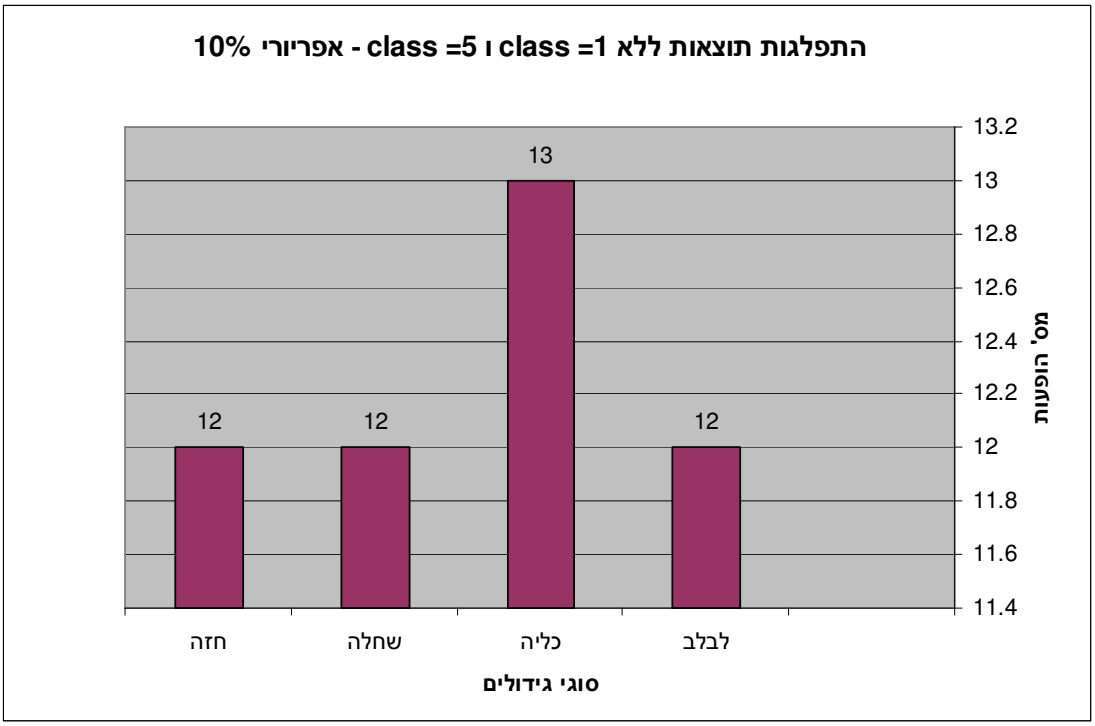
טבלה 6.4.1 ג' – תוצאות אפריורי עבור רמת תמיכה 10% (ללא class = 1,5)

ניתן לראות בצורה ברורה ומובהקת כי החוקים שהתקבלו כעת הינם רבים יותר מאשר קודם (10 לעומת 6) כאשר class 1 ו 5 היו בבסיס הנתונים. מאידך ניתן לראות כי רמת הפילוג בחוקים לא השתנתה (איור 6.4.1 ג'). הסיבה לגידול כמות החוקים הינה אכן ההסרה של שני סוגי הגידולים 1 ו 5 מבסיס הנתונים. ההורדה של רכיב כה משמעותי בבסיס הנתונים שינתה את גודלו של בסיס הנתונים וכעת בכדי שתת קבוצה תהיה תדירה עליה להופיע פחות פעמים ממקודם. לדוגמא:

אם גודלו של בסיס הנתונים היה X ומס' המופעים של תת קבוצה מסוימת היה Y אך רמת התמיכה המינימלית הינה Z. אם בבדיקה הראשונה יצא שקבוצה זו אינה תדירה בהכרח עלינו לומר כי:

$\frac{y}{X} < z$ אך כעת לאחר שגודלו של בסיס הנתונים קטן (נניח שקטן ב a מופעים) נקבל:

$\frac{y}{X-a}$ בכל מקרה נוכל לומר בוודאות כי $\frac{y}{X-a} > \frac{y}{X}$ לכן, קיים סיכוי כי ביטויים שמקודם לא עמדו ברמת התמיכה המינימלית יעמדו כעת ברמה זו.



איור 6.4.1 ג' – התפלגות סוגי גידולים בתוצאות אפריורי 10% ללא class = 1 ו class = 5

כעת נבצע הרצה על אותו בסיס נתונים כאשר רמת התמיכה הינה 5%

Antecedent (גורם)	Consequent (תוצאה)	Rule ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 18.0	peritoneum = 1.0	13	29	13.426	82.759	11.111	2.75
class = 18.0	pleura = 1.0	12	29	13.426	37.931	5.093	1.821
class = 11.0	liver = 1.0	18	28	12.963	75	9.722	2.418
class = 11.0	abdominal = 1.0	19	28	12.963	64.286	8.333	1.827
class = 11.0	peritoneum = 1.0	17	28	12.963	50	6.481	1.662
class = 22.0	axillar = 1.0	14	24	11.111	87.5	9.722	7.875
class = 14.0	bone = 1.0	10	24	11.111	54.167	6.019	2.167
class = 14.0	lung = 1.0	11	24	11.111	50	5.556	2.077
class = 22.0	supraclavicular = 1.0	15	24	11.111	41.667	4.63	3.333
class = 22.0	bone = 1.0	16	24	11.111	41.667	4.63	1.667
class = 2.0	neck = 1.0	1	20	9.259	100	9.259	6
class = 12.0	liver = 1.0	8	16	7.407	81.25	6.019	2.619
class = 12.0	abdominal = 1.0	9	16	7.407	75	5.556	2.132
class = 4.0	bone = 1.0	3	14	6.481	78.571	5.093	3.143
class = 7.0	liver = 1.0	6	14	6.481	64.286	4.167	2.072
class = 7.0	abdominal = 1.0	7	14	6.481	64.286	4.167	1.827
class = 4.0	lung = 1.0	4	14	6.481	50	3.241	2.077
class = 7.0	peritoneum = 1.0	5	14	6.481	42.857	2.778	1.424
class = 4.0	mediastinum = 1.0	2	14	6.481	35.714	2.315	2.204

טבלה 6.4.1 ד' – תוצאות אפריורי עבור רמת תמיכה 5% (ללא class = 1,5)

כעת נבצע הרצה דומה על בסיס נתונים המכיל אך ורק את class =1 ו class = 5 (מדד התמיכה יהיה 10% ורמת הביטחון 30%)

Antecedent (גורם)	Consequent (תוצאה)	Rule ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 1.0	***mediastinum***	***6***	***84***	***68.293***	***61.905***	***42.276***	***1.336***
class = 1.0	***bone***	***4***	***84***	***68.293***	***39.286***	***26.829***	***1.208***
class = 1.0	***liver***	***5***	***84***	***68.293***	***34.524***	***23.577***	***1.011***
class = 5.0	***abdominal***	***3***	***39***	***31.707***	***46.154***	***14.634***	***1.456***
class = 5.0	***peritoneum***	***1***	***39***	***31.707***	***41.026***	***13.008***	***1.682***
class = 5.0	liver	2	39	31.707	33.333	10.569	0.976

טבלה 6.4.1 ה' – תוצאות אפריורי עבור רמת תמיכה 10% (רק עם class = 1,5)

ניתן להבחין בצורה ברורה כי רמת התמיכה עלתה אך רמת הביטחון נשארה ללא שינוי (עבור אותם חוקים).

לדוגמא: חוק מס' 15 בטבלה 6.4.1 א':

Antecedent (גורם)	Consequent (תוצאה)	ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 1.0	mediastinum	15	84	24.779	61.905	15.339	2.281

לעומתו חוק מס' 6 בטבלה 6.4.1 ה' (אותו חוק בדיוק)

Antecedent (גורם)	Consequent (תוצאה)	ID	Instances	Support %	Confidence %	Rule Support %	Lift
mediastinum	class = 1.0	6	84	68.293	61.905	42.276	1.336

דוגמא נוספת: חוק מס' 12 בטבלה 6.4.1 א':

Antecedent (גורם)	Consequent (תוצאה)	ID	Instances	Support %	Confidence %	Rule Support %	Lift
class = 5.0	liver = 1.0	12	39	11.504	33.333	3.835	1.037

לעומתו חוק מס' 2 בטבלה 6.4.1 ה' (אותו חוק בדיוק)

Antecedent (גורם)	Consequent (תוצאה)	ID	Instances	Support %	Confidence %	Rule Support %	Lift
liver	class = 5.0	2	39	31.707	33.333	10.569	0.976

הסיבה להגדלת התמיכה הינה הקטנתו של בסיס הנתונים. מכיוון שבסיס הנתונים הינו כעת קטן יותר (אך מס' המופעים של מחלקה 1 ו5 לא השתנה) לכן אחוז המופע שלהם מתוך כלל בסיס הנתונים גדל בצורה משמעותית.

רמת הביטחון נשארה זהה וזאת מכיוון שרמת הביטחון יכולה להיות מובעת באמצעות מדד התמיכה:

$$Confidence(a \to b) = \frac{Support(A \to B)}{Support(A)}$$

ולכן, מפני ש Support(A) לא השתנה, וגם support(a→b) לא השתנה רמת הביטחון נותרה זהה.

מאידך ניתן לראות כי מדד ה lift קטן בצורה משמעותית, ה lift נתון ע"י הנוסחא

$Lift(a \rightarrow b) = conf(a \rightarrow b) / support(b)$

ומכיוון שרמת הביטחון לא השתנתה אך רמת התמיכה עלתה קיבלנו שגם ה lift קטן.

כפי שציינו קודם לכן, רמת הביטחון לא משתנה בכדי לשמור על רמת תוצאות מינימלית. כלומר אנו מעוניינים לוודא כי התוצאות הינן **נכונות ומהימנות** יותר מאשר נפוצות. לכן לא התפשרנו על רמת הביטחון.

כעת נציג טבלה מסכמת לכלל הפעולות שבוצעו בחלק זה

שיטה	רמת תמיכה מינימלית	רמת ביטחון	חוקים חזקים	סה"כ חוקים	זמן ריצה
אפריורי כלל הנתונים	10%	30%	6	6	פחות משניה
אפריורי כלל הנתונים	5%	30%	16	16	
אפריורי ללא 1,5	10%	30%	10	10	
אפריורי ללא 1,5	5%	30%	19	19	
אפריורי רק 1,5	10%	30%	5	6	

טבלה 6.4.1 ו' – טבלה מסכמת עבור אפריורי

מהטבלה המסכמת עולות מספר נקודות:

1. הורדה של רמת התמיכה המינימלית אכן מגדילה את כמות החוקים החזקים שמתקבלת.
2. סינון קבוצות ששכיחות בצורה בולטת מבסיס הנתונים מאפשר לחוקים נוספים "לצוץ ולעלות" בתהליך הכרייה. במקרה שלנו אומנם התווספו חוקים חדשים אך לא התווספו מיקומים חדשים של גידולים (עוד ערכים של המשתנה class - מיקום הגדול).
3. עקב גודלו היחסית קטן של בסיס הנתונים התקבלו התוצאות במהירות יחסית גבוהה (פחות משניה) בכלל האפשרויות שהוצגו לעיל.
4. כמובן ,שחוקים שהתגלו כאשר רמת התמיכה המינימלית היא 10% יתגלו גם כאשר רמת התמיכה היא 5%. לכן כמויות החוקים הן חופפות בין שתי רמות התמיכה.

Carma 6.4.2

בשונה מאפריורי ב Carama לא ניתן להגדיר אילו שדות יהיו בכל צד של החוק. הצורה היחידה שבה ניתן להגביל את השדות היא למנוע מהם להופיע כלל בחוק. ניתן לומר ש Carma לא מגביל את המשתמש בבחירת מיקום השדה בחוק עבור כל אחד מהשדות. בניגוד לאפריורי רמת התמיכה הנמדדת היא של כל החוק ולא רק של חלקו הראשון. לכן יתקבלו הרבה יותר חוקים.

נציין בנוסף שהגדרת התמיכה כאן הינה עבור Rule Support ולא ה Support המוכר מאפריורי.

התוצאות שהתקבלו עבור 10% תמיכה (ברמת החוק – rule support) ו 30% רמת ביטחון מינימלית:

Antecedent (גורם)	Consequent (תוצאה)	Rule ID	Instances	Support %	Confidence %	Rule Support %	Lift
abdominal	liver	3	115	33.923	57.391	19.469	1.785
abdominal	mediastinum	12	115	33.923	39.13	13.274	1.442
abdominal	peritoneum	13	115	33.923	39.13	13.274	1.396
abdominal	lung	18	115	33.923	33.043	11.209	1.494
liver	abdominal	2	109	32.153	60.55	19.469	1.785
liver	mediastinum	16	109	32.153	35.78	11.504	1.318
liver	peritoneum	17	109	32.153	35.78	11.504	1.277
peritoneum	abdominal	8	95	28.024	47.368	13.274	1.396
peritoneum	liver	11	95	28.024	41.053	11.504	1.277
peritoneum	pleura	14	95	28.024	37.895	10.619	1.713
mediastinum	class1	4	92	27.139	56.522	15.339	2.281
mediastinum	abdominal	6	92	27.139	48.913	13.274	1.442
mediastinum	liver	10	92	27.139	42.391	11.504	1.318
mediastinum	lung	15	92	27.139	36.957	10.029	1.67
class1	mediastinum	1	84	24.779	61.905	15.339	2.281
lung	abdominal	5	75	22.124	50.667	11.209	1.494
pleura	peritoneum	7	75	22.124	48	10.619	1.713
lung	mediastinum	9	75	22.124	45.333	10.029	1.67

טבלה 6.4.2 א' – תוצאות Carma עבור רמת Rule Support 10%

ניתן לראות כי חוק מס' 1 בטבלה 6.4.2 א' הינו מקביל לחוק מס' 15 בטבלה 6.4.1 א'.

הסיבה שאיננו רואים את שאר החוקים שהתקבלו מאפריורי נעוצה ברמת ה Rule Support שלהם. מעיון בתוצאות אפריורי ניתן לראות כי רמת ה Rule Support של שאר החוקים נמוכה מ 10% לכן הם לא מוצגים כאן.

אם נריץ שוב את Carma עם רמת Rule Support נמוכה יותר -5% נקבל: את הטבלה בנספח ב'

ניתן לראות כי חוץ מחוק 79 (מודגש באדום) שמדד ה Lift שלו קטן מאחד, כלל החוקים הינם חזקים.

נציין כי חלק גדול מהחוקים אינו דומה לחוקים שהוצגו עד עתה. עובדה זו נגרמה כתוצאה מהעובדה (שכבר הוסברה לעיל) שבמימוש אלגוריתם זה אין משמעות לכיווניות של החוק.

פירוט אחוד של כלל החוקים יוצג בפרק 6.4.4.

בכל אופן ניתן לומר בצורה ברורה כי כל חוק שנמצא ע"י carma נמצא גם ע"י אפריורי באחת מצורות ההרצה שלו.

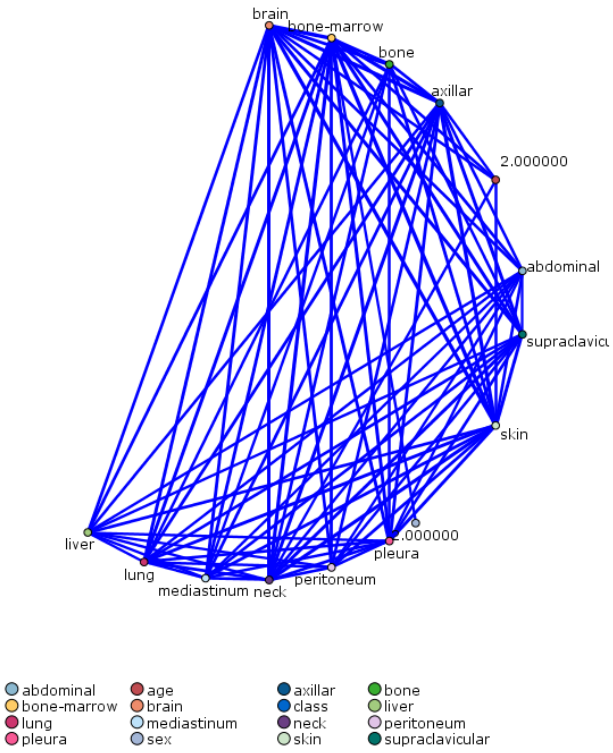
6.4.3 אפריורי – ללא חשיבות לסדר

בחלק זה הורץ אלגוריתם אפריורי כאשר כל אחד ממאפייני בסיס הנתונים יכול להיות בכל אחד מצדדיו של החוק. הרצה נוספת זו נועדה למצוא קשרים נוספים בין המאפיינים של בסיס הנתונים. ביניהם מאפיינים שלא בוצעה עליהם כרייה ב סעיף 6.4.1 (כמו גיל, סוג גידול, רמת חומרה וכו'...).

מפאת גודלה של טבלת התוצאות (242 שורות) היא לא תובא כאן אלא בנספח ג' . תוצאות מעניינות מהרצה זו יובאו בהמשך(חלק 6.4.4).

<u>מציאת קשרים בין שדות בבסיס הנתונים</u>

בכדי לנסות למצות מידע נוסף מבסיס הנתונים נשתמש בדיאגרמת הקשרים בין השדות (Web Diagram). דיאגראמה זו מבצעת מעבר על בסיס הנתונים ומגדירה את רמת הקשר בין השדות. רמת הקשר עושה שימוש בסיסי במודל Market – Basket שתואר בתחילת העבודה. בעצם, ניתן לומר כי האלגוריתם מחשב את מס' הפעמים שזוג שדות כל שהוא מופיע ביחד ועל סמך מס' המופעים מגדיר את רמת הקשר בין השדות הנ"ל. דיאגרמה זו הינה שימושית מכיוון שהיא מגדירה בצורה ויזואלית וקלה להבנה את הקשרים בין השדות. כמו כן, היא מאפשרת למצוא קשרים בין ערכים של שדות.



איור 6.4.3 א' – קשרים בין שדות בבסיס הנתונים

באיור 6.4.3 א' ניתן לראות קשרים חזקים בלבד בין שדות בבסיס הנתונים.
במקרה זה אין כיווניות כלומר כלל השדות בבסיס הנתונים עוברים בדיקה למציאת רמת הקשר ביניהם ללא קשר למיקומם. כמו כן מוצגים רק קשרים בין שני שדות בלבד (בשונה מאפריורי שמשלב חוקים עם יותר משני שדות). למרות זאת המידע המוצג בגרף הינו תצוגתי בלבד מפני שההרצה של אפריורי בסעיף 6.4.3 מכסה את כלל המקרים שהגרף מכסה.
במקרה שלנו הגרף לא תרם רבות לחוקים שהתקבלו. מכיוון שכמעט כל החוקים שהוגדרו "חזקים" (חוזק מבוטא כאן ברמת תמיכה)בגרף הזה הינם חוקים שאינם רלוונטיים (בעיקר חוקי שלילה) כדוגמת : אם אין גרורות במוח עצם אזי לא קימות גרורות במח.
באופן עקרוני קיימת חפיפה מלאה בין קשרים שמתקבלים בדיאגרמה הנ"ל לבין שימוש בחוקי כריית הקשר לכן לא נשתמש בדיאגרמה הנ"ל בהמשך העבודה.

באופן כללי ייתכן וניתן יהיה לראות הבדלים בחוקים שהתקבלו בשתי השיטות, וזאת מכיוון שאנו משתמשים ב אלגוריתמים בעלי מדדים שונים של תמיכה.
CARMA משתמש ב Rule Support, בעוד שאפריורי משתמש בהגדרה המוכרת לתמיכה. סיבה זו גורמת לבדלים בתוצאות בין האלגוריתמים.

6.4.4 החוקים שהתקבלו

<u>קשר בין גידולים לגרורות</u>

בחלק זה נביא בצורה מרוכזת את כלל החוקים הרלוונטיים שהתקבלו מכריית הנתונים(בכלל השיטות שנבדקו).[80]

יובאו רק חוקים שרמת הביטחון שלהם הינה מעל 50%[81] ומדד ה lift גדול מ 1.

מקום הגידול	מיקום הגרורות	רמת ביטחון
ראש / צוואר	צוואר	100%
שחלה	צפק	82%
לבלב	כבד	75%
לבלב	חלל הבטן	64%
ריאה	mediastinum[82]	61.905%
כליה	עצמות	54.167%
כליה	ראות	50%
לבלב	צפק	50%

טבלה 6.4.4 א' – ריכוז חוקים שהתקבלו עם class 1 ו class 5

בטבלה זו יובאו חוקים נוספים שהתגלו לאחר סינון class 1 ו class 5 מבסיס הנתונים

מקום הגידול	מיקום הגרורות	רמת ביטחון
חזה	בית השחי	87.5%
כיס המרה	כבד	81.25%
בלוטת התריס	עצמות	78.5%
כיס המרה	חלל הבטן	75%
מעי גס	חלל הבטן	64.28%
מעי גס	כבד	64.28%
בלוטת התריס	ריאות	50%

טבלה 6.4.4 א' – ריכוז חוקים שהתקבלו ללא class 1 ו class 5

<u>חוקים שונים</u>

בחלק זה נביא מספר חוקים שהתגלו במהלך הכרייה הקשורים לשאר המאפיינים.

עקב אורכה של רשימת החוקים שהתגלו היא תובא בנספח ד'.

---

[80] יש לשים לב לשוני בין הגדרת התמיכה של Carma לבין ההגדרה של אפריורי. להסבר נוסף ראה תחילת פרק היישום.

[81] המספר 50% הינו שרירותי. ונועד בכדי לבחור רק את החוקים בעלי סיכוי גבוה להיות נכונים.

[82] חלל החזה בין שני שקי הצדר, הכולל בתוכו את האיברים הפנימיים שבתוך החזה מלבד הריאות: כלומר את הלב,אבי העורקים, הוושט, קנה הנשימה התחתון, ועוד איברים נוספים מכונה בעברית הָבַּיְנָה.

בכדי להציג את כלל החוקים(טבלאות 6.4.4 א', 6.4.4 ב') בצורה קריאה ונוחה יותר נשתמש בטבלה הבאה:

גידול/גרורה	צוואר	צפק	כבד	חלל הבטן	mediastinum	עצמות	ראות	חלל החזה	בית השחי
ראש /צוואר	100%								
שחלה		82%							
לבלב		50%	75%	64%					
ריאות				גברים רמת חומרה 3 70.27%	61.905% גילאי 30-59 61.404% דרגת חומרה 3 72.727% גברים 61.818%			גברים מעל 60 60.465% רמת חומרה 3 גילאי 30-59 74.286%	
כליה						54.167%	50%		
חזה									87.5%
כיס המרה			81.25%	75%					
בלוטת התריס						78.5%	50%		
מעי גס			64.28%	64.28%					
רמת חומרה 3 (חמור)			מעל גיל 60 57.143%	52% מעל גיל 60 57.143%	גברים 55.35% גברים בגילאי 30-59 56.364%				

טבלה 6.4.4 ג' – ריכוז חוקים הקשורים לשאר המאפיינים שהתקבלו.

בטבלה הבאה נציג מאפיינים שאינם קשורים לקשרים בין גידולים אלא לקשרים בין מאפיינים שונים.

מאפיין/גרורה	כבד	חלל הבטן	חלל החזה
נשים חולות בסרטן	<u>מעל גיל 60</u> 50.769% <u>דרגת חומרה 3</u> 50%	<u>מעל גיל 60</u> 52%	<u>דרגת חומרה 3 וגיל 30-59</u> 61.538%

טבלה 6.4.4 ד' – ריכוז חוקים – קשרים בין מאפיינים שונים

כעת נביא טבלה מסכמת לקשרים הקשורים לשאר המאפיינים בחוק.
השורות בטבלה ייצגו את המאפיין הגורם בחוק והעמודות את מאפיין התוצאה. באופן כללי מדובר על גרורות (כלומר קשר בין מיקום גרורה במקום אחד למיקום גרורה במקום אחר) אא"כ צוין אחרת.

גורם/תוצאה	צוואר	צפק	כבד	חלל הבטן	חלל החזה	ריאות
חלל הבטן		<u>גילאי 30-59</u> 50%	57.391% <u>מעל גיל 60</u> 62.26% <u>גברים:</u> 57.447% <u>גברים 30-59</u> 53.448% <u>נשים:</u> 57.353% <u>נשים מעל 60</u> 61.765%		<u>גברים</u> 51.06% <u>דרגת חומרה 3</u> 61.538%	

גורם/תוצאה	צוואר	צפק	כבד	חלל הבטן	חלל החזה	ריאות
			<u>דרגת חומרה 3</u> 64.103%			
כבד				60.55% <u>גילאי 30-59</u> 57.407% <u>נשים</u> 65% <u>גברים</u> 55.1% <u>רמת חומרה 3</u> 55.556%	<u>רמת חומרה 3</u> 57.778%	
ריאות				50.667% <u>גילאי 30-59</u> 50% <u>נשים</u> 53.846%	<u>גברים</u> 50% <u>גילאי 30-59</u> 50%	
חלל החזה			<u>דרגת חומרה 3</u> 50% <u>נשים</u> 51.282%	<u>נשים</u> 53.846%		52.174%
צפק וחלל הבטן			57.778%			
חלל החזה			57.778%			

גורם/תוצאה	צוואר	צפק	כבד	חלל הבטן	חלל החזה	ריאות
וחלל הבטן						
צדר		<u>גילאי</u> <u>30-59</u> 54.054%	<u>נשים</u> 55.556%			
מעל עצם הבריח					<u>גילאי</u> <u>30-59</u> 50%	
צפק וכבד				66.667%		
חלל החזה וכבד				66.667%		
ריאות וחלל הבטן			71.053%		57.895%	
ריאות וחלל החזה			52.941%	64.706%		

טבלה 6.4.4 ה' – ריכוז חוקים – קשרים בין גרורות

6.5 סיכום וניתוח התוצאות

בחלק זה הצגנו יישום של אלגוריתמים לכריית מידע בתחום חקר הסרטן. חשוב לציין שלמרות השוני בין האלגוריתמים הקיימים לכריית חוקי הקשר, לרוב, התוצאות שיוצגו בהרצות בתנאים זהים שלהם יהיו דומות. וזאת מכיוון שהפרמטרים שעל פיהם אנו בוחרים תתי קבוצות תדירות ויוצרים חוקי הקשר הינם זהים.

עקב גודלו הקטן של בסיס הנתונים מספר החוקים שהתקבלו היה קטן יחסית, לכן הורדנו את רמת התמיכה המינימלית בכדי לאפשר לחוקים בעלי רמת תמיכה נמוכה לבוא לידי ביטוי. פעולה נוספת שבוצעה הייתה הורדה של שני מחלקות שכיחות יתר על המידה מבסיס הנתונים. כך התאפשר לקבוצות נוספות לבוא לידי ביטוי בחוקים שהתקבלו.

עם זאת, לא הורדנו את רמת הביטחון המינימלית בכדי לא להתפשר על איכותם של החוקים שמתקבלים. למרות זאת מספר החוקים שהתקבלו היה קטן יחסית. כמו כן, ניתן לראות כי טבלת החוקים אינה שלמה כלומר לא לכל מקרה קיים חוק.

ניתן לייחס זאת לגודלו הקטן של בסיס הנתונים. מחקר שכזה בבסיס נתונים גדול ומקיף יותר יניב תוצאות משמעותיות יותר באופן מובהק.

התוצאות לא נבדקו מבחינה רפואית. אך גם ללא ידע רפואי ניתן להבחין בקשר בין גרורות במקום מסוים בגוף למקום אחר הקרוב אליו. או קשר בין גידול במקום מסוים לגרורות במקום הסמוך אליו. כמו כן ניתן להבחין בקשר בין העלייה בגיל החולה להסתברות למציאת הגרורות במקום מסוים. בולט במיוחד הקשר בין גרורות בחלל הבטן לגרורות בכבד הקשר הינו דו כיווני ומופיע בתוצאות בוריאציות שונות.

7. סיכום והצעה להמשך מחקר

בעבודה זו סקרנו בצורה השוואתית את תחום כריית המידע בכללותו והתמקדנו בכריית חוקי הקשר.בתחום חוקי ההקשר סקרנו ששה אלגוריתמים והשוונו ביניהם בהיבטים של סיבוכיות זמן ומקום. האלגוריתמים שנסקרו היו:

- האלגוריתם הנאיבי
- אפריורי
- FP Growth
- Eclat
- DIC
- Carma

לאחר מכן סקרנו את תחום כריית חוקי ההקשר במחקר הביולוגי ובמחקר הרפואי – בהתמקדות על מחלת הסרטן.

בפן היישומי של העבודה נעשה שימוש בבסיס נתונים המכיל מידע לגבי חולי סרטן מיקום הגידולים ומיקום הגרורות. בחלק היישומי בוצע שימוש באלגוריתם אפריורי וב Carma. לצורך כריית המידע. באמצעות בסיס נתונים זה נבדקו קשרים בין מיקומי הגידולים הסרטניים למיקומי הגרורות בגוף. כמו כן נבדקו גם קשרים עבור מיקומי הגרורות בינם לבין עצמן. נמצאו מספר חוקים המצביעים על קשר בסבירויות שונות בין הגידולים והגרורות. בעיקר בין גרורות בחלל הבטן לגרורות בכבד, נתון שחזר על עצמו לאורך כלל הבדיקות שבוצעו.

בכדי לתקף ולאמת את תוצאות המחקר עלינו תחילה לעבות משמעותית את בסיס הנתונים שנעשה בו שימוש לבסיס נתונים שיכיל מידע רב,מגוון, ומפולג בצורה אחידה יותר מבסיס הנתונים הקיים כיום. כמחקר המשך בתחום מדעי המחשב ניתן לבדוק ולהשוות את התאמתם של האלגוריתמים השונים למחקר רפואי בתחום כריית המידע. כידוע, בסיסי נתונים רפואיים הינם קטנים מאוד ולכן החוקים בבסיסי נתונים אלו הינם בעלי רמת תמיכה מינימלית נמוכה. לכן ניתן לצפות שאלגוריתמים שונים יתמודדו בצורה שונה עם בסיסי נתונים שכאלו.

כמחקר המשך בתחום הרפואי ניתן להרחיב את המחקר לבסיסי נתונים (הקשורים לחקר הסרטן) גדולים יותר ובעלי נתונים רבים יותר ובכך להרחיב את התוצאות ולתקף את נכונותן במידת וודאות גבוהה יותר. כמו כן, ניתן לשקול לשלב דרכים אוטומטיות לצמצום החוקים שהתקבלו בכדי לקבל מידע איכותי ואמין ללא רעשים מיותרים.

חשיבות העבודה מתבטאת הן בתחום מדעי המחשב והן בתחום הרפואי.

בתחום מדעי המחשב העובדה מציגה ומנתחת בצורה השוואתית ומלאה את האלגוריתמים העיקריים לכריית חוקי הקשר. *בתחום הרפואי* נמצאו קשרים בין מיקומים של גידולים סרטניים שונים לגרורות סרטניות בגוף. מידע שכזה- אם יהיה ידוע מראש יכול למנוע התפתחות של גרורות בשאר הגוף במידה ומגלים מספיק מוקדם את הגידול. כמו כן נמצאו חוקים המציינים קשרים בין מיקומים של גרורות. חוקים אלו יכולים לסייע במניעה של התפתחות נוספת של גרורות מגרורות קיימות.ניתן לומר כי סוג כזה של עבודה הינה צעד נוסף בדרך למיצוי ידע חדש מידע הקיים כיום. בעזרת ידע זה ניתן יהיה לגלות אפיקים חדשים במחקר המדעי בכלל והרפואי בפרט.

8. מקורות

כריית מידע

[1] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad, "A tree projection algorithm for generation of frequent item sets," Journal of Parallel and Distributed Computing, vol. 61, no. 3, pp. 350-371, 2001.

[2] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *in SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, vol. 22, no. 2. New York, NY, USA: ACM, 1993, pp. 207-216.

[3] M. Akbar - FP Growth in Mining Text Documents, Montana State University. Course Notes

[4] C. Borgelt. Software for Frequent Pattern Mining.(Source Code) http://www.borgelt.net/fpm.html

[5] C. Borgelt.Efficient Implementations of Apriori and Eclat.*Workshop of Frequent Item Set Mining Implementations* (FIMI 2003, Melbourne, FL, USA).

[6] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," *in SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 1997, pp. 255-264.

[7] C.Cérin , M.Koskas, J.S.Gay,G. Le Mahec. Efficient Data-Structures and Parallel Algorithms for Association Rules Discovery enc, *Fifth Mexican International Conference in Computer Science (ENC'04)*,pp.399-406. 2004

[8] R. Dilly - Data Mining Chapter 1 (course notes)

[9] B. Goethals Frequent Pattern Mining Implementations. http://adrem.ua.ac.be/~goethals/software/

[10] B. Goethals, Survey on Frequent Pattern Mining (2003). http://adrem.ua.ac.be/~goethals/software/survey.pdf

[11] B.Goethals, Memory issues in frequent itemset mining (2004), *Proceedings of the 2004 ACM symposium on Applied computing,* pp 530-534, Nicosia, Cyprus.

[12] R. P. Gopalan . Y. Giri Sucahyo. Improving the Efficiency of Frequent Pattern Mining by Compact Data Structure Design. *Intelligent Data Engineering and Automated Learning* .Vol 2690/2003.Springer Berlin / Heidelberg (2003) pp.576-583

[13] J.Han & M. Kamber - *Data mining :Concepts and techniques* – Academic Press 2006

[14] J.HAN , J.PEI ,Y.YIN ,R.MAO - Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8, 53–87, 2004.

[15] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," Data Mining and Knowledge Discovery, vol. 15, no. 1, pp. 55-86, August 2007.

[16] Jiawei Han ,Jian Pei. Mining Frequent Patterns by Pattern-Growth: Methodology and Implications. *ACM SIGKDD*, December 2000. Volume 2, Issue 2 - page 14.

[17] C. Hidber, Online Association Rule Mining. Proceedings of the 1999 ACM SIGMOD international conference on Management of data, p.145-156, May 31-June 03, 1999, Philadelphia, Pennsylvania, United States.

[18] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining - a general survey and comparison," *SIGKDD Explor. Newsl.*, vol. 2, no. 1, pp. 58-64, June 2000

[19] J.Hipp, U.Guentzer and G.Nakhaeizadeh. Algorithms for Association Rule Mining – A General Survey and Comparison. Course Presentation(2003).

[20] M. Hegland, Algorithms for association rules, *ser. Lecture Notes In Artificial Intelligence*. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 226-234

[21] H.Jiang, Y.Zhao, X.Dong. Mining Positive and Negative Weighted Association Rules from Frequent Itemsets Based on Interest *iscid,* vol. 2, pp.242-245, *International Symposium on Computational Intelligence and Design*, 2008.

[22] R. Kessl. Frequent substructure mining - an introduction..Course Notes(2009).

[23] P.Krishna Reddy - Web Data Mining, Association Rules. Course Notes (2003)

[24] L. Liu , E. Li , Y. Zhang , Z. Tang, Optimization of frequent itemset mining on multiple-core processor, Proceedings of the 33rd international conference on Very large data bases, Vienna, Austria (2007) pp.1275-1285

[25] K.Malik, N.Raheja, P. Garg. "Enhanced FP-Growth Algorithm". *IJCEM International Journal of Computational Engineering & Management*, Vol. 12, April 2011.pp 54-56.

[26] Elmasri, Navathe – *Fundamentals of Database System.* Chapter 28 : Data Mining Concepts. Addison Wesley; 5 edition (2006).

[27] J. S. Park, M. S. Chen, and P. S. Yu, "An effective hash based algorithm for mining association rules," *in Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data* .San Jose, California (1995) pp. 175-186

[28] Gregory Piatetsky-Shapiro - Machine Learning and Data Mining – Course Notes.

[29] L. Schmidt-Thieme. Algorithmic Features of Eclat. *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04)*, volume 126 of CEUR Workshop Brighton, UK, 2004.

[30] M. Song, S. Rajasekaran, A Transaction Mapping Algorithm for Frequent Itemsets Mining, *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 472-481, Apr. 2006

[31] R. Srikant and R. Agrawal, "Mining generalized association rules," *Future Generation Computer Systems*, vol. 13, no. 2-3, pp. 161-180, 1997

[32] Tan, Steinbach, Kumar - Introduction to data mining - – chapter 6 + 7 (Course Notes)

[33] R.Winter, " Why Are Data Warehouses Growing So Fast?".April 2008 .http://www.b-eye-network.com/view/7188.

[34] O.R. Zaïane - Principles of Knowledge Discovery in Databases - Chapter 1 Introduction To Data Mining - 1999 (Course Notes)

[35] M. J. Zaki , Mitsunori Ogihara. Theoretical Foundations of Associations Rules. *3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (1998)

[36] M.J. Zaki . Scalable Algorithms for Association Mining. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 12, NO. 3, MAY/JUNE 2000.

[37] W.Zhang, H.Liao, N. Zhao, "Research on the FP Growth Algorithm about Association Rule Mining," *International Seminar on Business and Information Management*,vol. 1, pp.315-318, 2008.

[38] M. Zwitter and M. Soklic. Primary Tumor DataSet. University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.:

[39] צוריאל כהן (2009) "כריית חוקי הקשר" עבודה סמינריונית – האוניברסיטה הפתוחה

[40] http://www.kmining.com/info_definitions.html

[41] A. Bellaachia and E. Guven, “Predicting Breast Cancer Survivability using Data Mining Techniques”, Ninth Workshop on Mining Scientific and Engineering Datasets in conjunction with the Sixth SIAM International Conference on Data Mining (SDM 2006),

[42] B.Fang ,W.Hsu , M. Li Lee, Tumor cell identification using features rules, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Edmonton, Alberta, Canada, pp 495-500.(2002).

[43] M. Karabatak, M. Cevdet Ince. An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications: An International Journal Expert Systems with Applications* pp: 3465-3469. (March 2009)

[44] H. Kwasnicka and K. Switalski. Discovery of association rules from medical data - classical and evolutionary approaches. *Proceedings of Autumn Meeting of Polish Information Processing Society* (2005).pp 163-167.

[45] J. Li, A. W.-C. W. Fu, and P. Fahey, "Efficient discovery of risk patterns in medical data." *Artificial intelligence in medicine*, September 2008.

[46] J. Nahar, K. Tickle, A. Ali, and Y.-P. Chen, "Significant cancer prevention factor extraction: An association rule discovery approach," *Journal of Medical Systems* (2009)

[47] C. Ordonez, C. A. Santana, and L. de Braal, "Discovering interesting association rules in medical data," *in ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2000, pp. 78-85

[48] C. Ordonez, Comparing association rules and decision trees for disease prediction, *Proceedings of the international workshop on Healthcare information and knowledge management*, 2006, pp 17 - 24

[49] C. Ordonez , et al. "Mining constrained association rules to predict heart disease". *Proceedings of thr IEEE ICDM Conference*(2001), pp. 433–440.

[50] Y. Qiang, Y. Guo, X. Li, Q. Wang, H. Chen, D. Cuic. The Diagnostic Rules of Peripheral Lung Cancer Preliminary Study Based on Data Mining Technique. *Journal of Nanjing Medical University* Volume 21, Issue 3, April 2007, Pages 190-195 .

9. נספחים

9.1 נספח א'

1. **class**: lung, head & neck, esophasus, thyroid, stomach, duoden & sm.int, colon, rectum, anus, salivary glands, pancreas, gallblader, liver, kidney, bladder, testis, prostate, ovary, corpus uteri, cervix uteri, vagina, breast
מיקום הגידול הסרטני

2. **age**: <30, 30-59, >=60

3. **sex**: male, female

4. **histologic-type**: epidermoid, adeno, anaplastic
סוג הרקמה

5. **degree-of-diffe**: well, fairly, poorly
רמת שונות של התא הסרטני

6. **bone**: yes, no - עצם

7. **bone-marrow**: yes, no – מח עצם

8. **lung**: yes, no - ריאות

9. **pleura**: yes, no – חלל החזה

10. **peritoneum**: yes, no -צפק

11. **liver**: yes, no - כבד

12. **brain**: yes, no - מח

13. **skin**: yes, no - עור

14. **neck**: yes, no - צוואר

15. **supraclavicular**: yes, no - צוואר

16. **axillar**: yes, no – בית השחי

17. **mediastinum**: yes, no - חלל החזה

18. **abdominal**: yes, no -חלל הבטן
מיקומי הגרורות 6-18

בכדי להקל על הבנת תהליך הכרייה נציין כי כל מיקום של גידול מסומן במספר סידורי עוקב מ 1 עד 22. כדלהלן:

1	ריאות		12	כיס המרה
2	ראש / צוואר		13	כבד
3	ושט		14	כליה
4	בלוטת התריס		15	שלפוחית השתן
5	קיבה		16	אשך
6	תריסריון		17	ערמונית
7	מעי גס		18	שחלה
8	חלחולת (קצה המעי הגס)		19	רחם
9	פי הטבעת		20	צוואר הרחם
10	בלוטת הרוק		21	פתח הרחם
11	לבלב		22	חזה

בכלל המקרים נציין כי שדות קטגוריאלים (המכילים יותר משני ערכים אפשריים) הומרו לשדות בוליאנים. מספר השדות הסופי הוא כמספר הערכים. לדוגמא: המאפיין **histologic-type** מכיל שלושה ערכים אפשריים – בייצוג הבוליאני יתקבלו שלושה שדות שכל אחד הינו בוליאני.
כלומר:

histologic-type={ epidermoid, adeno, anaplastic }

המאפיין עבר המרה לשלשה שדות:

Epidermoid={True, False}, **adeno =** {True, False}, **anaplastic =**{True, False} ,

9.2 נספח ב'

Antecedent (גורם)	Consequent (תוצאה)	Rule ID	Instances	Support %	Confidence %	Rule Support %	Lift
abdominal	liver	26	115	33.923	57.391	19.469	1.785
abdominal	mediastinum	59	115	33.923	39.13	13.274	1.442
abdominal	peritoneum	60	115	33.923	39.13	13.274	1.396
abdominal	lung	73	115	33.923	33.043	11.209	1.494
liver	abdominal	19	109	32.153	60.55	19.469	1.785
liver	mediastinum	65	109	32.153	35.78	11.504	1.318
liver	peritoneum	66	109	32.153	35.78	11.504	1.277
peritoneum	abdominal	38	95	28.024	47.368	13.274	1.396
peritoneum	liver	50	95	28.024	41.053	11.504	1.277
peritoneum	pleura	61	95	28.024	37.895	10.619	1.713
bone	class1	67	94	27.729	35.106	9.735	1.417
bone	mediastinum	74	94	27.729	32.979	9.145	1.215
mediastinum	class1	27	92	27.139	56.522	15.339	2.281
mediastinum	abdominal	35	92	27.139	48.913	13.274	1.442
mediastinum	liver	47	92	27.139	42.391	11.504	1.318
mediastinum	lung	63	92	27.139	36.957	10.029	1.67
mediastinum	bone	71	92	27.139	33.696	9.145	1.215
mediastinum	pleura	77	92	27.139	32.609	8.85	1.474
class1	mediastinum	18	84	24.779	61.905	15.339	2.281
class1	bone	58	84	24.779	39.286	9.735	1.417
class1	liver	70	84	24.779	34.524	8.555	1.074
lung	abdominal	34	75	22.124	50.667	11.209	1.494
pleura	peritoneum	37	75	22.124	48	10.619	1.713
lung	mediastinum	44	75	22.124	45.333	10.029	1.67
pleura	abdominal	45	75	22.124	42.667	9.44	1.258
lung	liver	52	75	22.124	40	8.85	1.244
pleura	mediastinum	53	75	22.124	40	8.85	1.474
lung	abdominal and liver	64	75	22.124	36	7.965	1.849
lung	bone	72	75	22.124	33.333	7.375	1.202
pleura	class1	78	75	22.124	32	7.08	1.291
pleura	*liver*	*79*	*75*	*22.124*	*30.667*	*6.785*	*0.954*
abdominal and liver	lung	51	66	19.469	40.909	7.965	1.849
abdominal and liver	mediastinum	55	66	19.469	39.394	7.67	1.452
abdominal and liver	peritoneum	56	66	19.469	39.394	7.67	1.406
supraclavicular	mediastinum	46	61	17.994	42.623	7.67	1.571
supraclavicular	class1	57	61	17.994	39.344	7.08	1.588
supraclavicular	neck	75	61	17.994	32.787	5.9	2.526
class1 and mediastinum	liver	48	52	15.339	42.308	6.49	1.316
class1 and mediastinum	abdominal	68	52	15.339	34.615	5.31	1.02
class1 and mediastinum	bone	69	52	15.339	34.615	5.31	1.248
class1 and mediastinum	supraclavicular	76	52	15.339	32.692	5.015	1.817

abdominal and mediastinum	liver	24	45	13.274	57.778	7.67	1.797
abdominal and peritoneum	liver	25	45	13.274	57.778	7.67	1.797
abdominal and mediastinum	lung	36	45	13.274	48.889	6.49	2.21
abdominal and mediastinum	pleura	49	45	13.274	42.222	5.605	1.908
abdominal and mediastinum	class1	54	45	13.274	40	5.31	1.614
abdominal and peritoneum	pleura	62	45	13.274	37.778	5.015	1.708
neck	class2	42	44	12.979	45.455	5.9	7.705
neck	supraclavicular	43	44	12.979	45.455	5.9	2.526
liver and mediastinum	abdominal	11	39	11.504	66.667	7.67	1.965
liver and peritoneum	abdominal	12	39	11.504	66.667	7.67	1.965
liver and mediastinum	class1	28	39	11.504	56.41	6.49	2.277
class5	abdominal	40	39	11.504	46.154	5.31	1.361
liver and mediastinum	lung	41	39	11.504	46.154	5.31	2.086
abdominal and lung	liver	9	38	11.209	71.053	7.965	2.21
abdominal and lung	mediastinum	23	38	11.209	57.895	6.49	2.133
peritoneum and pleura	abdominal	39	36	10.619	47.222	5.015	1.392
lung and mediastinum	abdominal	14	34	10.029	64.706	6.49	1.907
lung and mediastinum	liver	32	34	10.029	52.941	5.31	1.647
axillar	class22	16	33	9.735	63.636	6.195	8.989
bone and class1	mediastinum	30	33	9.735	54.545	5.31	2.01
axillar	supraclavicular	33	33	9.735	51.515	5.015	2.863
abdominal and pleura	mediastinum	21	32	9.44	59.375	5.605	2.188
abdominal and pleura	liver	29	32	9.44	56.25	5.31	1.749
abdominal and pleura	peritoneum	31	32	9.44	53.125	5.015	1.896
bone and mediastinum	class1	22	31	9.145	58.065	5.31	2.343
liver and lung	abdominal	2	30	8.85	90	7.965	2.653
mediastinum and pleura	abdominal	17	30	8.85	63.333	5.605	1.867
liver and lung	mediastinum	20	30	8.85	60	5.31	2.211
class18	peritoneum	5	29	8.555	82.759	7.08	2.953

class1 and liver	mediastinum	7	29	8.555	75.862	6.49	2.795
class11	liver	8	28	8.26	75	6.195	2.333
class11	abdominal	15	28	8.26	64.286	5.31	1.895
mediastinum and supraclavicular	class1	13	26	7.67	65.385	5.015	2.639
class22	axillar	3	24	7.08	87.5	6.195	8.989
class1 and supraclavicular	mediastinum	10	24	7.08	70.833	5.015	2.61
liver and pleura	abdominal	6	23	6.785	78.261	5.31	2.307
abdominal and class1	mediastinum	4	21	6.195	85.714	5.31	3.158
class2	neck	1	20	5.9	100	5.9	7.705

תוצאות Carma עבור רמת Rule Support 5%

9.3 נספח ג'

Antecedent (גורם)	Consequent (תוצאה)	Rule ID	Instances	Support %	Confidence %	Rule Support %	Lift
age = 2.0	bone = 1.0	27	209	61.652	31.579	19.469	1.139
age = 2.0	peritoneum = 1.0	33	209	61.652	30.144	18.584	1.076
sex = 2.0	abdominal = 1.0	48	177	52.212	38.418	20.059	1.132
sex = 2.0	peritoneum = 1.0	32	177	52.212	35.593	18.584	1.27
sex = 2.0	liver = 1.0	47	177	52.212	33.898	17.699	1.054
sex = 1.0	mediastinum = 1.0	41	161	47.493	32.298	15.339	1.19
sex = 1.0	bone = 1.0	26	161	47.493	31.677	15.044	1.142
sex = 1.0	liver = 1.0	46	161	47.493	30.435	14.454	0.947
abdominal = 1.0	liver = 1.0	45	115	33.923	57.391	19.469	1.785
abdominal = 1.0	peritoneum = 1.0	31	115	33.923	39.13	13.274	1.396
abdominal = 1.0	mediastinum = 1.0	40	115	33.923	39.13	13.274	1.442
abdominal = 1.0	lung = 1.0	19	115	33.923	33.043	11.209	1.494
liver = 1.0	abdominal = 1.0	44	109	32.153	60.55	19.469	1.785
liver = 1.0	peritoneum = 1.0	29	109	32.153	35.78	11.504	1.277
liver = 1.0	mediastinum = 1.0	38	109	32.153	35.78	11.504	1.318
age = 3.0	abdominal = 1.0	35	107	31.563	49.533	15.634	1.46
age = 3.0	liver = 1.0	34	107	31.563	47.664	15.044	1.482
sex = 1.0 and age = 2.0	bone = 1.0	135	106	31.268	38.679	12.094	1.395
sex = 1.0 and age = 2.0	mediastinum = 1.0	183	106	31.268	36.792	11.504	1.356
sex = 2.0 and age = 2.0	peritoneum = 1.0	150	102	30.088	45.098	13.569	1.609
sex = 2.0 and age = 2.0	abdominal = 1.0	198	102	30.088	31.373	9.44	0.925
sex = 2.0 and age = 2.0	pleura = 1.0	79	102	30.088	30.392	9.145	1.374
degree-of-diffe = 3	mediastinum = 1.0	36	100	29.499	52	15.339	1.916
degree-of-diffe = 3	liver = 1.0	42	100	29.499	45	13.274	1.4
degree-of-diffe = 3	abdominal = 1.0	43	100	29.499	39	11.504	1.15
degree-of-diffe = 3	bone = 1.0	25	100	29.499	30	8.85	1.082
peritoneum = 1.0	abdominal = 1.0	30	95	28.024	47.368	13.274	1.396
peritoneum = 1.0	liver = 1.0	28	95	28.024	41.053	11.504	1.277
peritoneum = 1.0	pleura = 1.0	9	95	28.024	37.895	10.619	1.713
bone = 1.0	mediastinum = 1.0	23	94	27.729	32.979	9.145	1.215
mediastinum = 1.0	abdominal = 1.0	39	92	27.139	48.913	13.274	1.442
mediastinum = 1.0	liver = 1.0	37	92	27.139	42.391	11.504	1.318
mediastinum =	lung = 1.0	16	92	27.139	36.957	10.029	1.67

1.0							
mediastinum = 1.0	bone = 1.0	24	92	27.139	33.696	9.145	1.215
mediastinum = 1.0	pleura = 1.0	11	92	27.139	32.609	8.85	1.474
class = 1.0	mediastinum = 1.0	21	84	24.779	61.905	15.339	2.281
class = 1.0	bone = 1.0	20	84	24.779	39.286	9.735	1.417
class = 1.0	liver = 1.0	22	84	24.779	34.524	8.555	1.074
lung = 1.0	abdominal = 1.0	18	75	22.124	50.667	11.209	1.494
pleura = 1.0	peritoneum = 1.0	8	75	22.124	48	10.619	1.713
lung = 1.0	mediastinum = 1.0	15	75	22.124	45.333	10.029	1.67
pleura = 1.0	abdominal = 1.0	13	75	22.124	42.667	9.44	1.258
pleura = 1.0	mediastinum = 1.0	10	75	22.124	40	8.85	1.474
lung = 1.0	liver = 1.0	17	75	22.124	40	8.85	1.244
lung = 1.0	bone = 1.0	14	75	22.124	33.333	7.375	1.202
pleura = 1.0	liver = 1.0	12	75	22.124	30.667	6.785	0.954
abdominal = 1.0 and sex = 2.0	liver = 1.0	195	68	20.059	57.353	11.504	1.784
abdominal = 1.0 and sex = 2.0	peritoneum = 1.0	147	68	20.059	38.235	7.67	1.364
abdominal = 1.0 and sex = 2.0	lung = 1.0	112	68	20.059	30.882	6.195	1.396
abdominal = 1.0 and sex = 2.0	mediastinum = 1.0	180	68	20.059	30.882	6.195	1.138
liver = 1.0 and abdominal = 1.0	lung = 1.0	104	66	19.469	40.909	7.965	1.849
liver = 1.0 and abdominal = 1.0	peritoneum = 1.0	139	66	19.469	39.394	7.67	1.406
liver = 1.0 and abdominal = 1.0	mediastinum = 1.0	170	66	19.469	39.394	7.67	1.452
bone = 1.0 and age = 2.0	mediastinum = 1.0	131	66	19.469	31.818	6.195	1.172
age = 3.0 and sex = 2.0	abdominal = 1.0	160	65	19.174	52.308	10.029	1.542
age = 3.0 and sex = 2.0	liver = 1.0	158	65	19.174	50.769	9.735	1.579
peritoneum = 1.0 and age = 2.0	abdominal = 1.0	148	63	18.584	46.032	8.555	1.357
peritoneum = 1.0 and sex = 2.0	abdominal = 1.0	146	63	18.584	41.27	7.67	1.217
peritoneum = 1.0 and sex = 2.0	pleura = 1.0	63	63	18.584	39.683	7.375	1.794
peritoneum = 1.0 and age = 2.0	liver = 1.0	143	63	18.584	39.683	7.375	1.234
peritoneum = 1.0 and age = 2.0	pleura = 1.0	65	63	18.584	38.095	7.08	1.722
peritoneum =	liver = 1.0	141	63	18.584	34.921	6.49	1.086

1.0 and sex = 2.0							
supraclavicular = 1.0	mediastinum = 1.0	7	61	17.994	42.623	7.67	1.571
supraclavicular = 1.0	neck = 1.0	5	61	17.994	32.787	5.9	2.526
liver = 1.0 and sex = 2.0	abdominal = 1.0	194	60	17.699	65	11.504	1.916
mediastinum = 1.0 and age = 2.0	abdominal = 1.0	181	60	17.699	41.667	7.375	1.228
liver = 1.0 and sex = 2.0	peritoneum = 1.0	142	60	17.699	36.667	6.49	1.308
mediastinum = 1.0 and age = 2.0	liver = 1.0	175	60	17.699	36.667	6.49	1.14
mediastinum = 1.0 and age = 2.0	supraclavicular = 1.0	53	60	17.699	35	6.195	1.945
mediastinum = 1.0 and age = 2.0	pleura = 1.0	72	60	17.699	35	6.195	1.582
mediastinum = 1.0 and age = 2.0	lung = 1.0	100	60	17.699	35	6.195	1.582
mediastinum = 1.0 and age = 2.0	bone = 1.0	132	60	17.699	35	6.195	1.262
liver = 1.0 and sex = 2.0	mediastinum = 1.0	174	60	17.699	33.333	5.9	1.228
liver = 1.0 and sex = 2.0	lung = 1.0	107	60	17.699	30	5.31	1.356
abdominal = 1.0 and age = 2.0	liver = 1.0	197	58	17.109	53.448	9.145	1.662
abdominal = 1.0 and age = 2.0	peritoneum = 1.0	149	58	17.109	50	8.555	1.784
abdominal = 1.0 and age = 2.0	mediastinum = 1.0	182	58	17.109	43.103	7.375	1.588
abdominal = 1.0 and age = 2.0	pleura = 1.0	78	58	17.109	36.207	6.195	1.637
abdominal = 1.0 and age = 2.0	lung = 1.0	114	58	17.109	36.207	6.195	1.637
class = 1.0 and age = 2.0	mediastinum = 1.0	123	57	16.814	61.404	10.324	2.263
degree-of-diffe = 1	peritoneum = 1.0	6	57	16.814	38.596	6.49	1.377
class = 1.0 and age = 2.0	bone = 1.0	118	57	16.814	38.596	6.49	1.392
degree-of-diffe = 3 and sex = 1.0	mediastinum = 1.0	165	56	16.519	55.357	9.145	2.04
degree-of-diffe = 3 and sex = 1.0	liver = 1.0	186	56	16.519	41.071	6.785	1.277
degree-of-diffe	abdominal = 1.0	189	56	16.519	33.929	5.605	1

= 3 and sex = 1.0							
degree-of-diffe = 3 and sex = 1.0	bone = 1.0	133	56	16.519	32.143	5.31	1.159
class = 1.0 and degree-of-diffe = 3	mediastinum = 1.0	119	55	16.224	72.727	11.799	2.68
class = 1.0 and sex = 1.0	mediastinum = 1.0	122	55	16.224	61.818	10.029	2.278
degree-of-diffe = 3 and age = 2.0	mediastinum = 1.0	167	55	16.224	56.364	9.145	2.077
degree-of-diffe = 3 and age = 2.0	liver = 1.0	188	55	16.224	41.818	6.785	1.301
class = 1.0 and degree-of-diffe = 3	bone = 1.0	116	55	16.224	38.182	6.195	1.377
class = 1.0 and sex = 1.0	bone = 1.0	117	55	16.224	38.182	6.195	1.377
class = 1.0 and degree-of-diffe = 3	liver = 1.0	124	55	16.224	38.182	6.195	1.187
degree-of-diffe = 3 and age = 2.0	abdominal = 1.0	191	55	16.224	32.727	5.31	0.965
class = 1.0 and degree-of-diffe = 3	pleura = 1.0	59	55	16.224	30.909	5.015	1.397
class = 1.0 and degree-of-diffe = 3	abdominal = 1.0	125	55	16.224	30.909	5.015	0.911
class = 1.0 and sex = 1.0	liver = 1.0	126	55	16.224	30.909	5.015	0.961
class = 1.0 and sex = 1.0	abdominal = 1.0	127	55	16.224	30.909	5.015	0.911
degree-of-diffe = 3 and age = 2.0	bone = 1.0	134	55	16.224	30.909	5.015	1.115
liver = 1.0 and age = 2.0	abdominal = 1.0	196	54	15.929	57.407	9.145	1.692
liver = 1.0 and age = 2.0	peritoneum = 1.0	144	54	15.929	46.296	7.375	1.652
liver = 1.0 and age = 2.0	mediastinum = 1.0	176	54	15.929	40.741	6.49	1.501
age = 3.0 and abdominal = 1.0	liver = 1.0	156	53	15.634	62.264	9.735	1.936
age = 3.0 and abdominal = 1.0	mediastinum = 1.0	152	53	15.634	35.849	5.605	1.321
age = 3.0 and abdominal = 1.0	lung = 1.0	89	53	15.634	30.189	4.72	1.365
mediastinum = 1.0 and degree-of-diffe = 3	liver = 1.0	161	52	15.339	50	7.67	1.555
pleura = 1.0	peritoneum = 1.0	64	52	15.339	46.154	7.08	1.647

and age = 2.0							
mediastinum = 1.0 and degree-of-diffe = 3	abdominal = 1.0	163	52	15.339	46.154	7.08	1.361
mediastinum = 1.0 and sex = 1.0	abdominal = 1.0	177	52	15.339	46.154	7.08	1.361
class = 1.0 and mediastinum = 1.0	liver = 1.0	120	52	15.339	42.308	6.49	1.316
pleura = 1.0 and age = 2.0	mediastinum = 1.0	71	52	15.339	40.385	6.195	1.488
pleura = 1.0 and age = 2.0	abdominal = 1.0	77	52	15.339	40.385	6.195	1.19
mediastinum = 1.0 and sex = 1.0	bone = 1.0	129	52	15.339	36.538	5.605	1.318
mediastinum = 1.0 and sex = 1.0	liver = 1.0	171	52	15.339	36.538	5.605	1.136
mediastinum = 1.0 and degree-of-diffe = 3	pleura = 1.0	66	52	15.339	34.615	5.31	1.565
mediastinum = 1.0 and sex = 1.0	lung = 1.0	96	52	15.339	34.615	5.31	1.565
class = 1.0 and mediastinum = 1.0	bone = 1.0	115	52	15.339	34.615	5.31	1.248
class = 1.0 and mediastinum = 1.0	abdominal = 1.0	121	52	15.339	34.615	5.31	1.02
class = 1.0 and mediastinum = 1.0	supraclavicular = 1.0	51	52	15.339	32.692	5.015	1.817
class = 1.0 and mediastinum = 1.0	pleura = 1.0	58	52	15.339	30.769	4.72	1.391
age = 3.0 and liver = 1.0	abdominal = 1.0	155	51	15.044	64.706	9.735	1.907
bone = 1.0 and sex = 1.0	mediastinum = 1.0	128	51	15.044	37.255	5.605	1.373
liver = 1.0 and sex = 1.0	abdominal = 1.0	192	49	14.454	55.102	7.965	1.624
liver = 1.0 and sex = 1.0	mediastinum = 1.0	172	49	14.454	38.776	5.605	1.429
liver = 1.0 and sex = 1.0	peritoneum = 1.0	140	49	14.454	34.694	5.015	1.238
abdominal = 1.0 and sex = 1.0	liver = 1.0	193	47	13.864	57.447	7.965	1.787
abdominal = 1.0 and sex =	mediastinum = 1.0	178	47	13.864	51.064	7.08	1.882

1.0 abdominal = 1.0 and sex = 1.0	peritoneum = 1.0	145	47	13.864	40.426	5.605	1.443
abdominal = 1.0 and sex = 1.0	pleura = 1.0	75	47	13.864	36.17	5.015	1.635
abdominal = 1.0 and sex = 1.0	lung = 1.0	110	47	13.864	36.17	5.015	1.635
peritoneum = 1.0 and sex = 2.0 and age = 2.0	pleura = 1.0	207	46	13.569	41.304	5.605	1.867
peritoneum = 1.0 and sex = 2.0 and age = 2.0	abdominal = 1.0	235	46	13.569	39.13	5.31	1.153
peritoneum = 1.0 and sex = 2.0 and age = 2.0	liver = 1.0	234	46	13.569	34.783	4.72	1.082
peritoneum = 1.0 and abdominal = 1.0	liver = 1.0	138	45	13.274	57.778	7.67	1.797
degree-of-diffe = 3 and liver = 1.0	mediastinum = 1.0	162	45	13.274	57.778	7.67	2.129
mediastinum = 1.0 and abdominal = 1.0	liver = 1.0	169	45	13.274	57.778	7.67	1.797
pleura = 1.0 and sex = 2.0	peritoneum = 1.0	62	45	13.274	55.556	7.375	1.982
degree-of-diffe = 3 and liver = 1.0	abdominal = 1.0	184	45	13.274	55.556	7.375	1.638
mediastinum = 1.0 and abdominal = 1.0	lung = 1.0	94	45	13.274	48.889	6.49	2.21
mediastinum = 1.0 and abdominal = 1.0	pleura = 1.0	68	45	13.274	42.222	5.605	1.908
peritoneum = 1.0 and abdominal = 1.0	pleura = 1.0	61	45	13.274	37.778	5.015	1.708
pleura = 1.0 and sex = 2.0	mediastinum = 1.0	69	45	13.274	33.333	4.425	1.228
pleura = 1.0 and sex = 2.0	abdominal = 1.0	76	45	13.274	33.333	4.425	0.983
peritoneum = 1.0 and abdominal = 1.0	lung = 1.0	86	45	13.274	31.111	4.13	1.406
degree-of-diffe = 3 and liver =	lung = 1.0	101	45	13.274	31.111	4.13	1.406

1.0 degree-of-diffe = 3 and sex = 2.0	liver = 1.0	187	44	12.979	50	6.49	1.555
degree-of-diffe = 3 and sex = 2.0	mediastinum = 1.0	166	44	12.979	47.727	6.195	1.759
neck = 1.0	supraclavicular = 1.0	4	44	12.979	45.455	5.9	2.526
degree-of-diffe = 3 and sex = 2.0	abdominal = 1.0	190	44	12.979	45.455	5.9	1.34
degree-of-diffe = 3 and sex = 2.0	pleura = 1.0	74	44	12.979	31.818	4.13	1.438
class = 1.0 and sex = 1.0 and age = 2.0	mediastinum = 1.0	225	43	12.684	60.465	7.67	2.228
class = 1.0 and sex = 1.0 and age = 2.0	bone = 1.0	216	43	12.684	39.535	5.015	1.426
bone = 1.0 and sex = 2.0	lung = 1.0	83	43	12.684	30.233	3.835	1.367
class = 1.0 and sex = 1.0 and age = 2.0	supraclavicular = 1.0	202	43	12.684	30.233	3.835	1.68
supraclavicular = 1.0 and age = 2.0	mediastinum = 1.0	52	42	12.389	50	6.195	1.842
lung = 1.0 and age = 2.0	mediastinum = 1.0	99	42	12.389	50	6.195	1.842
lung = 1.0 and age = 2.0	abdominal = 1.0	113	42	12.389	50	6.195	1.474
age = 3.0 and sex = 1.0	abdominal = 1.0	159	42	12.389	45.238	5.605	1.334
age = 3.0 and sex = 1.0	liver = 1.0	157	42	12.389	42.857	5.31	1.333
supraclavicular = 1.0 and age = 2.0	neck = 1.0	49	42	12.389	38.095	4.72	2.935
lung = 1.0 and age = 2.0	liver = 1.0	108	42	12.389	35.714	4.425	1.111
lung = 1.0 and age = 2.0	bone = 1.0	84	42	12.389	33.333	4.13	1.202
lung = 1.0 and age = 2.0	pleura = 1.0	57	42	12.389	30.952	3.835	1.399
lung = 1.0 and age = 2.0	peritoneum = 1.0	88	42	12.389	30.952	3.835	1.105
bone = 1.0 and sex = 1.0 and age = 2.0	mediastinum = 1.0	230	41	12.094	34.146	4.13	1.258
class = 1.0 and mediastinum = 1.0 and degree-of-diffe = 3	liver = 1.0	217	40	11.799	45	5.31	1.4
class = 1.0 and mediastinum = 1.0 and	abdominal = 1.0	218	40	11.799	37.5	4.425	1.105

degree-of-diffe = 3 class = 1.0 and mediastinum = 1.0 and degree-of-diffe = 3	supraclavicular = 1.0	199	40	11.799	32.5	3.835	1.806
class = 1.0 and mediastinum = 1.0 and degree-of-diffe = 3	pleura = 1.0	204	40	11.799	32.5	3.835	1.469
class = 1.0 and mediastinum = 1.0 and degree-of-diffe = 3	bone = 1.0	211	40	11.799	32.5	3.835	1.172
peritoneum = 1.0 and liver = 1.0	abdominal = 1.0	137	39	11.504	66.667	7.67	1.965
mediastinum = 1.0 and liver = 1.0	abdominal = 1.0	168	39	11.504	66.667	7.67	1.965
degree-of-diffe = 3 and abdominal = 1.0	liver = 1.0	185	39	11.504	64.103	7.375	1.994
degree-of-diffe = 3 and abdominal = 1.0	mediastinum = 1.0	164	39	11.504	61.538	7.08	2.268
degree-of-diffe = 3 and sex = 1.0 and age = 2.0	mediastinum = 1.0	238	39	11.504	61.538	7.08	2.268
lung = 1.0 and sex = 2.0	abdominal = 1.0	111	39	11.504	53.846	6.195	1.587
mediastinum = 1.0 and sex = 2.0	abdominal = 1.0	179	39	11.504	53.846	6.195	1.587
mediastinum = 1.0 and sex = 2.0	liver = 1.0	173	39	11.504	51.282	5.9	1.595
class = 5.0	abdominal = 1.0	3	39	11.504	46.154	5.31	1.361
mediastinum = 1.0 and liver = 1.0	lung = 1.0	91	39	11.504	46.154	5.31	2.086
lung = 1.0 and sex = 2.0	liver = 1.0	106	39	11.504	46.154	5.31	1.435
class = 5.0	peritoneum = 1.0	1	39	11.504	41.026	4.72	1.464
lung = 1.0 and sex = 2.0	mediastinum = 1.0	97	39	11.504	41.026	4.72	1.512
mediastinum = 1.0 and sex = 2.0	lung = 1.0	98	39	11.504	41.026	4.72	1.854
liver = 1.0 and abdominal =	lung = 1.0	210	39	11.504	41.026	4.72	1.854

1.0 and sex = 2.0 mediastinum = 1.0 and sex = 2.0	pleura = 1.0	70	39	11.504	38.462	4.425	1.738
liver = 1.0 and abdominal = 1.0 and sex = 2.0	peritoneum = 1.0	233	39	11.504	38.462	4.425	1.372
mediastinum = 1.0 and sex = 1.0 and age = 2.0	abdominal = 1.0	240	39	11.504	38.462	4.425	1.134
degree-of-diffe = 3 and sex = 1.0 and age = 2.0	liver = 1.0	241	39	11.504	38.462	4.425	1.196
degree-of-diffe = 3 and abdominal = 1.0	lung = 1.0	102	39	11.504	35.897	4.13	1.623
mediastinum = 1.0 and sex = 1.0 and age = 2.0	lung = 1.0	209	39	11.504	35.897	4.13	1.623
mediastinum = 1.0 and sex = 1.0 and age = 2.0	bone = 1.0	231	39	11.504	35.897	4.13	1.295
degree-of-diffe = 3 and sex = 1.0 and age = 2.0	bone = 1.0	232	39	11.504	35.897	4.13	1.295
liver = 1.0 and abdominal = 1.0 and sex = 2.0	mediastinum = 1.0	239	39	11.504	35.897	4.13	1.323
class = 5.0	liver = 1.0	2	39	11.504	33.333	3.835	1.037
mediastinum = 1.0 and liver = 1.0	pleura = 1.0	67	39	11.504	33.333	3.835	1.507
lung = 1.0 and sex = 2.0	bone = 1.0	82	39	11.504	33.333	3.835	1.202
mediastinum = 1.0 and sex = 1.0 and age = 2.0	supraclavicular = 1.0	203	39	11.504	33.333	3.835	1.852
mediastinum = 1.0 and sex = 1.0 and age = 2.0	pleura = 1.0	208	39	11.504	33.333	3.835	1.507
lung = 1.0 and sex = 2.0	pleura = 1.0	56	39	11.504	30.769	3.54	1.391
degree-of-diffe = 3 and abdominal = 1.0	pleura = 1.0	73	39	11.504	30.769	3.54	1.391
lung = 1.0 and sex = 2.0	peritoneum = 1.0	87	39	11.504	30.769	3.54	1.098
mediastinum =	bone = 1.0	130	39	11.504	30.769	3.54	1.11

1.0 and sex = 2.0							
degree-of-diffe = 3 and abdominal = 1.0	peritoneum = 1.0	136	39	11.504	30.769	3.54	1.098
degree-of-diffe = 3 and sex = 1.0 and age = 2.0	abdominal = 1.0	242	39	11.504	30.769	3.54	0.907
lung = 1.0 and abdominal = 1.0	liver = 1.0	103	38	11.209	71.053	7.965	2.21
lung = 1.0 and abdominal = 1.0	mediastinum = 1.0	93	38	11.209	57.895	6.49	2.133
lung = 1.0 and abdominal = 1.0	pleura = 1.0	55	38	11.209	36.842	4.13	1.665
lung = 1.0 and abdominal = 1.0	peritoneum = 1.0	85	38	11.209	36.842	4.13	1.315
class = 1.0 and degree-of-diffe = 3 and sex = 1.0	mediastinum = 1.0	219	37	10.914	70.27	7.67	2.589
class = 1.0 and degree-of-diffe = 3 and sex = 1.0	bone = 1.0	214	37	10.914	37.838	4.13	1.365
class = 1.0 and degree-of-diffe = 3 and sex = 1.0	liver = 1.0	226	37	10.914	37.838	4.13	1.177
class = 1.0 and degree-of-diffe = 3 and sex = 1.0	abdominal = 1.0	228	37	10.914	37.838	4.13	1.115
lung = 1.0 and sex = 1.0	mediastinum = 1.0	95	36	10.619	50	5.31	1.842
pleura = 1.0 and peritoneum = 1.0	abdominal = 1.0	60	36	10.619	47.222	5.015	1.392
lung = 1.0 and sex = 1.0	abdominal = 1.0	109	36	10.619	47.222	5.015	1.392
degree-of-diffe = 1 and age = 2.0	peritoneum = 1.0	50	36	10.619	41.667	4.425	1.487
lung = 1.0 and sex = 1.0	bone = 1.0	81	36	10.619	33.333	3.54	1.202
lung = 1.0 and sex = 1.0	liver = 1.0	105	36	10.619	33.333	3.54	1.037
class = 1.0 and degree-of-diffe = 3 and age = 2.0	mediastinum = 1.0	220	35	10.324	74.286	7.67	2.737
age = 3.0 and degree-of-diffe = 3	liver = 1.0	153	35	10.324	57.143	5.9	1.777

age = 3.0 and degree-of-diffe = 3	abdominal = 1.0	154	35	10.324	57.143	5.9	1.684
age = 3.0 and degree-of-diffe = 3	mediastinum = 1.0	151	35	10.324	42.857	4.425	1.579
class = 1.0 and mediastinum = 1.0 and age = 2.0	liver = 1.0	222	35	10.324	40	4.13	1.244
class = 1.0 and mediastinum = 1.0 and age = 2.0	supraclavicular = 1.0	201	35	10.324	34.286	3.54	1.905
class = 1.0 and mediastinum = 1.0 and age = 2.0	pleura = 1.0	205	35	10.324	34.286	3.54	1.55
class = 1.0 and degree-of-diffe = 3 and age = 2.0	bone = 1.0	215	35	10.324	34.286	3.54	1.236
class = 1.0 and mediastinum = 1.0 and age = 2.0	abdominal = 1.0	224	35	10.324	34.286	3.54	1.011
class = 1.0 and degree-of-diffe = 3 and age = 2.0	liver = 1.0	227	35	10.324	34.286	3.54	1.066
class = 1.0 and degree-of-diffe = 3 and age = 2.0	pleura = 1.0	206	35	10.324	31.429	3.245	1.421
class = 1.0 and mediastinum = 1.0 and age = 2.0	bone = 1.0	213	35	10.324	31.429	3.245	1.133
class = 1.0 and degree-of-diffe = 3 and age = 2.0	abdominal = 1.0	229	35	10.324	31.429	3.245	0.926
lung = 1.0 and mediastinum = 1.0	abdominal = 1.0	92	34	10.029	64.706	6.49	1.907
age = 3.0 and abdominal = 1.0 and sex = 2.0	liver = 1.0	237	34	10.029	61.765	6.195	1.921
lung = 1.0 and mediastinum = 1.0	liver = 1.0	90	34	10.029	52.941	5.31	1.647
class = 1.0 and mediastinum =	liver = 1.0	221	34	10.029	41.176	4.13	1.281

1.0 and sex = 1.0 class = 1.0 and mediastinum = 1.0 and sex = 1.0	abdominal = 1.0	223	34	10.029	41.176	4.13	1.214
lung = 1.0 and mediastinum = 1.0	pleura = 1.0	54	34	10.029	38.235	3.835	1.728
lung = 1.0 and mediastinum = 1.0 class = 1.0 and mediastinum = 1.0 and sex = 1.0	bone = 1.0	80	34	10.029	35.294	3.54	1.273
	supraclavicular = 1.0	200	34	10.029	35.294	3.54	1.961
class = 1.0 and mediastinum = 1.0 and sex = 1.0	bone = 1.0	212	34	10.029	35.294	3.54	1.273
age = 3.0 and abdominal = 1.0 and sex = 2.0	mediastinum = 1.0	236	34	10.029	32.353	3.245	1.192

9.3 נספח ד'

- קיים קשר בין גרורות בחלל הבטן לגרורות בכבד בסבירות של 57.391%
- קיים קשר בין גרורות בכבד לגרורות בחלל הבטן בסבירות של 60.55%
- במקרה ומדובר על גידול סרטני בדרגת חומרה הכי גבוהה שישנה.קיים סיכוי של 52% לגרורות בחלל הבטן.
- במקרה וישנן גרורות בריאות קיים סיכוי של 50.667% להתפשטות לחלל הבטן.
- לנשים בעלות גרורות בחלל הבטן ישנו סיכוי של 57.353% להתפשטות הגרורות לכבד
- לנשים בעלות גרורות בכבד ישנו סיכוי של 65% להתפשטות הגרורות לחלל הבטן.
- לנשים מעל גיל 60 שחולות בסרטן קיימת סבירות של 52% לגרורות בחלל הבטן
- לנשים מעל גיל 60 שחולות בסרטן קיימת סבירות של 50.769% לגרורות בכבד
- לגברים החולים בסרטן בעלי גרורות בחלל הבטן קיים סיכוי של 57.447% לגרורות גם בכבד.
- לגברים בגילאי 30-59 החולים בסרטן בעלי גרורות בחלל הבטן קיים סיכוי של 53.448% לגרורות גם בכבד.
- לחולי סרטן בגילאי 30-59 בעלי גרורות בחלל הבטן ישנו סיכוי של 50% לגרורות בצפק[83].
- לחולי סרטן בעלי גידול בריאות בגילאי 30-59 ישנו סיכוי של 61.404% לגרורות בחלל החזה(mediastinum).
- במקרה ומדובר על גברים בעלי גידול סרטני בדרגת חומרה הכי גבוהה שישנה.קיים סיכוי של 55.35% לגרורות בחלל החזה(mediastinum).
- במקרה ומדובר על גידול סרטני בריאה בדרגת חומרה הכי גבוהה שישנה.קיים סיכוי של 72.727% לגרורות בחלל החזה(mediastinum).
- לגברים בעלי גידול בריאות קיים סיכוי של 61.818% להתפשטות הגרורות לחלל החזה(mediastinum).
- במקרה ומדובר על גברים בגילאי 30-59 בעלי גידול סרטני בדרגת חומרה הכי גבוהה שישנה.קיים סיכוי של 56.364% לגרורות בחלל החזה(mediastinum).
- לחולי סרטן בגילאי 30-59 בעלי גרורות בכבד ישנו סיכוי של 57.407% לגרורות בחלל הבטן.
- לחולי סרטן בגילאים מעל ל 60 בעלי גרורות בחלל הבטן קיים סיכוי של 62.26% לגרורות בכבד.
- לחולים בעלי גידול סרטני בדרגת חומרה הכי גבוהה שישנה(דרגה 3) עם גרורות בחלל החזה, קיים סיכוי של 50% לגרורות בכבד.

[83] צפק הינו קרום דק ושקוף העוטף את דפנות חלל הבטן ואת חלקם הגדול של האיברים בחלל הבטן.

- לחולי סרטן בגילאים מעל ל 60 בעלי גרורות בכבד קיים סיכוי של 64.7% לגרורות בחלל הבטן.
- לגברים חולי סרטן בעלי גרורות בכבד ישנו סיכוי של 55.1% להתפשטות הגרורות לחלל הבטן.
- לגברים חולי סרטן בעלי גרורות בחלל הבטן קיים סיכוי של 51.06% של התפשטות בחלל החזה(mediastinum).
- במקרה של גרורות בחלל החזה(mediastinum) , קיים סיכוי של 52.174% לגרורות בריאות.
- במקרה של גרורות בצפק ובחלל הבטן קיים סיכוי של 57.778% לגרורות בכבד.
- במקרה של גידול ברמת חומרה הכי גבוהה עם גרורות בכבד קיים סיכוי של 57.778% לגרורות ב חלל החזה(mediastinum).
- במקרה של גרורות ב חלל החזה(mediastinum) ובחלל הבטן קיים סיכוי של 57.778% לגרורות ב כבד.
- במקרה של נשים בעלות גרורות בצדר[84] קיים סיכוי של 55.556% לגרורות בכבד.
- במקרה של גידול ברמת חומרה הכי גבוהה עם גרורות בכבד קיים סיכוי של 55.556% לגרורות בחלל הבטן.
- במקרה ומדובר על נשים בעלות גידול סרטני בדרגת חומרה הכי גבוהה שישנה(דרגה 3), קיים סיכוי של 50% לגרורות בכבד.
- במקרה של גברים בגילאים מעל 60 עם גידול בראות קיים סיכוי של 60.465% לגרורות בחלל החזה.
- לחולי סרטן בגילאי 30-59 בעלי גרורות באזור שמעל עצם הבריח קיים סיכוי של 50% לגרורות בחלל החזה.
- לחולי סרטן בגילאי 30-59 בעלי גרורות בריאות ישנו סיכוי של 50% לגרורות בחלל החזה ובבטן.
- לחולים בעלי גרורות בצפק ובכבד / בחלל החזה ובכבד, קיים סיכוי של 66.667% להימצאות גרורות גם בחלל הבטן.
- במקרה ומדובר על גידול ברמת חומרה 3 (הכי חמור) וקיימות גרורות בחלל הבטן ישנו סיכוי של 64.103% להתפשטות הגרורות לכבד.
- במקרה ומדובר על גידול ברמת חומרה 3 (הכי חמור) וקיימות גרורות בחלל הבטן ישנו סיכוי של 61.538% להתפשטות הגרורות לחלל החזה.
- במקרה ומדובר על גידול ברמת חומרה 3 (הכי חמור) והחולה היא בת בגילאים 30-59 ישנו סיכוי של 61.538% להתפשטות הגרורות לחלל החזה.

[84] ציפוי של הראות ושל המשטח הפנימי של דופן בית החזה. מסייע בהורדת החיכוך בין חלקים שונים בריאה בזמן הנשימה (קרוי גם אדר).

- לנשים בעלות גרורות בריאה / בחלל החזה קיים סיכוי של 53.846% להתפשטות הגרורה לחלל הבטן.
- לנשים בעלות גרורות בחלל החזה קיים סיכוי של 51.282% להתפשטות לכבד.
- לחולים בעלי גרורות בריאות ובחלל הבטן קיים סיכוי של 71.053% להתפשטות לכבד.
- לחולים בעלי גרורות בריאות ובחלל הבטן קיים סיכוי של 57.895% להתפשטות לחלל החזה.
- לגברים בעלי גידול בריאות עם רמת חומרת גידול – 3 (הכי חמור) ישנו סיכוי של 70.27% להתפשטות הגידול (כלומר לגרורות)בחלל הבטן.
- לחולים בגילאי 30-59 בעלי גרורות בצדר, קיים סיכוי של 54.054% להתפשטות הגרורות לצפק.
- לגברים בעלי גרורות בריאות קיים סיכוי של 50% להתפשטות הגרורות לחלל החזה.
- לחולים בגילאי 30-59 בעלי גידול בריאות בדרגת חומרה 3 (הכי חמור) קיים סיכוי של 74.286% לגרורות בחלל החזה.
- לחולים מעל גיל 60 בעלי גידול ברמת חומרה 3 (הכי חמור שיש) קיים סיכוי של 57.143% להתפשטות הגרורות לכבד ולחלל הבטן.
- לחולים בעלי גרורות בריאות ובחלל החזה קיים סיכוי של 64.706% לגרורות בחלל הבטן.
- לנשים חולות מעל גיל 60 עם גרורות בחלל הבטן ישנו סיכוי של 61.765% להתפשטות הגרורה לכבד.
- לחולים בעלי גרורות בריאות ובחלל החזה קיים סיכוי של 52.941% להתפשטות הגרורות לכבד.

Abstract

In this work we review sub area in data mining – Association Rules Mining. Association rules mining is a primary data mining tool. Using association rules mining it's possible to mine data from various databases. Association rules gives the ability to find and analyze relations between fields in the database. Based on those relations it will be possible to achieve new data which was not known before.

This document has four parts:

In the first part, some basic definitions and terms in data mining and association rules mining were presented.

In the second part, six basic algorithms in association rules mining were reviewed and compared each other:

- The naïve algorithm
- FP – Growth
- Eclat
- DIC
- Carma

The primary difference between those algorithms is not only in the way the association rules are mined, but also in the data structures that are used in those algorithms.

In the third part, some Medical – Biological applications of Association rules mining is presented.

The main problem that is being discussed in this document is the mining of medical association rules from medical database which contains data about tumor and metastasis locations on different patients. Using association rules we want to find relations between the fields in the given database.

In the fourth part an implementation to solve this problem is presented. Using the SPSS – Clementine program. After the data mining process is finished, a comparative analysis process will be performed to the algorithms.

As a further research we can perform the data mining process on a bigger database. It also will be interesting to add more properties to the database and analyze the relationship between those properties.

Table of Contents

The Open University of Israel
Department of Mathematics and Computer Science

Implementing association rules to find connection between tumor and metastasis location

Thesis submitted as partial fulfillment of the requirements
towards an M.Sc. degree in Computer Science
The Open University of Israel
Computer Science Division

by
Tzuriel Cohen
039076070

Prepared under the supervision of Dr. Maya Herman

December 2012