סמינר בנושא מיוחד

קיבוץ אשכולות קשה לעומת קיבוץ אשכולות רך בניתוח תמונה

שם המנחה: דוקטור מיה הרמן

שם המגישה: מורן פיינגולד

תעודת זהות: 308208537

מספר טלפון נייד: 0545604517

תאריך הגשה: 22.7.15

תוכן עניינים

1. מבוא ............................................................................................................... 3
2. תהליך כריית מידע וניתוח אשכולות ...................................................................... 4
   1. כריית מידע מהו? ....................................................................................... 4
   2. ניתוח אשכולות .......................................................................................... 4
   3. אלגוריתמים לניתוח אשכולות ....................................................................... 5
3. קיבוץ אשכולות קשה ......................................................................................... 11
   1. מהו קיבוץ אשכולות קשה ............................................................................ 11
   2. אלגוריתמים נפוצים בקיבוץ אשכולות קשה ..................................................... 11
   3. אלגוריתם K-Means .................................................................................. 11
   4. יתרונות וחסרונות האלגוריתם ...................................................................... 12
   5. הדגמת האלגוריתם .................................................................................... 12
4. קיבוץ אשכולות רך ............................................................................................ 18
   1. מהו קיבוץ אשכולות רך ............................................................................... 18
   2. אלגוריתמים נפוצים בקיבוץ אשכולות רך ........................................................ 18
   3. אלגוריתם Fuzzy C-Means ....................................................................... 19
   4. יתרונות וחסרונות האלגוריתם ....................................................................... 19
   5. הדגמת האלגוריתם .................................................................................... 20
5. השוואת קיבוץ קשה וקיבוץ רך ............................................................................ 23
   1. מאפיינים עיקריים ....................................................................................... 23
   2. יתרונות וחסרונות ....................................................................................... 26
   3. יישומים .................................................................................................... 27
6. ניתוח תמונה .................................................................................................... 28
   1. ניתוח תמונה בעזרת ניתוח אשכולות ............................................................. 28
   2. ניתוח תמונה בניתוח אשכולות קשה .............................................................. 30
   3. ניתוח תמונה בניתוח אשכולות רך ................................................................. 31
7. סיכום ............................................................................................................. 33
8. ביביליוגרפיה ................................................................................................... 34

מבוא

עבודה זו תתמקד בניתוח אשכולות, תת נושא של כריית מידע שמתעסק בקיבוץ נתונים לקבוצות.

נעמיק בניתוח אשכולות רך לעומת ניתוח אשכולות קשה, ובפרט נחקור שניים מהאלגוריתמים הבולטים, K-Means ו Fuzzy C-Means. שני אלגוריתמים אלו מהווים את הבסיס לאלגוריתמים מבוססי מרכז כובד ולכן לכל אלגוריתם ישנם דוגמאות רבות על ההרחבות והשימושים שלו.

בחרתי בהדגמת ההבדל בין ניתוח אשכולות רך לניתוח אשכולות קשה בשני אלגוריתמים אלו מכיוון שהם דוגמאות מייצגות ונפוצות לכל קבוצה, וכן בגלל הדמיון הרב בינהם. הדמיון בינהם יראה לנו אילו יכולות נפלאות יש לניתוח אשכולות שנקודות שוני כה קטנות יוצרות תוצרים כה שונים ומגוונים.

דווקא בעולם המודרני בו קיים כל כך הרבה מידע נגיש, עולה הצורך בניתוח ועיבוד המידע הזמין לפריצות דרך טכנולוגיות שיועילו לדורות הבאים, לשמירה על הבית שלנו כדור הארץ מפני כל הסכנות המחכות לו ולשמירה עלינו ועל ילדינו. אם היינו יכולים לחזות בדיוק אילו מחלות עשויות לתקוף אותנו או מה יקרה אם מספר הקרחונים באנטרטיקה יקטן לאורך השנים, היינו יכולים להגיע מוכנים עם פתרונות חדשים לבעיות שכלל לא צפינו. לכן, לפי דעתי כה חשוב להמשיך לחקור את התפתחות נושא כריית המידע, וחיזוי ידע על פי מאגרים רבים של מידע בפרט.

במהלך העבודה נראה כיצד ההבדלים בין קיבוץ אשכולות קשה לעומת קיבוץ אשכולות קל משפיעים על בעיית ניתוח התמונה על ידי הרצת ניתוח אותה התמונה בשני האלגוריתמים השונים. מבין היישומים הרבים של קיבוץ אשכולות בחרתי בנושא ניתוח תמונה מכיוון שהוא מכיל מרחב רחב של תתי בעיות, כל בעיה נפתרת בצורה שונה ומותאמת יותר לאלגוריתם מסוים בקיבוץ אשכולות. כמו כן, ניתוח תמונה טומן בחובו יכולות אינסופיות בין אם בתחום הרפואה או בתחומים אחרים, שמביאות מידי יום את החברה שלנו קדימה.

2. תהליך כריית מידע וניתוח אשכולות

משחר ההיסטוריה, המין האנושי התעניין בהפיכת מידע לידע. החל משבטים קדומים שניסו לנבא אסונות טבע, אנליסטים בבנקים שמנסים לגלות מה המנייה הגדולה הבאה ועד לשוטרים שמנסים לזהות גניבות זהות, זיהוי דפוסים במידע זוהי אחת הבעיות המאתגרות שמלוות את הקיום האנושי. עם ההתקדמות הטכנולוגית וצמיחת נפח המידע הזמין, עלה הצורך בגישות אוטומטיות יותר. ועל כן, תהליך כריית מידע הוא היכולת להשתמש באמצעים אוטומטיים (כגון אלגוריתם או תוכנית מחשב) על כמויות מידע גדולות והסקת מסקנות מכך. בתהליך כריית המידע נשתמש בשיטות אחרות בעולם מדעי המחשב בכדי לזהות תבניות נסתרות במידע. אחת משיטות אלו היא ניתוח אשכולות.

2.1 כריית מידע מהו?

כריית מידע (Data mining) הוא תהליך ניתוח אוטומטי של מאגרי נתונים גדולים או מורכבים, על מנת לחשוף תבניות או מגמות אותן לא ניתן היה לגלות בדרך אחרת. באמצעות סריקת נתונים של מאגרי מידע גדולים, יוכל להתבצע זיהוי של מאפיינים ותוכל להיווצר מערכת קשרים בינהם. בסוף התהליך נשאף להפוך מידע לידע. כריית נתונים היא תהליך עסקי מתמשך של יישום טכניקות סטטיסטיות ואינטליגנציה מלאכותית במאגרי נתונים גדולים המבוססים על מחסני נתונים במטרה לגלות מגמות נסתרות, תבניות וקשרים בין הנתונים, כדי להפוך ידע זה להחלטות עסקיות וביצועיות. ניתן אף לומר שכריית מדע היא ניבוי העתיד באמצעות נתוני העבר, ומי שמכיר את עולם כריית המידע וראה את כוחה יודע שהגדרה זו הופכת למציאות מידי יום [13].

2.2 ניתוח אשכולות

נצלול לתחום ניתוח האשכולות (Cluster Analysis). במושג זה נתייחס לקיבוץ אובייקטים לקבוצות כך שאובייקטים דומים יהיו קרובים יותר זה לזה, כלומר– שני אובייקטים באותה הקבוצה יהיו דומים יותר זה לזה מאשר לאובייקטים מקבוצות אחרות. לדוגמא, ניקח את רשת הקשרים של הפייסבוק, ונחלק את המשתתפים לתתי קבוצות על בסיס קשרים בין אישיים. אפילו בסט האובייקטים הכי פשוט נמצא תכונות רבות שעל פיהם יהיה ניתן לקבץ את האובייקטים בקבוצות שונות, כאשר התכונות ייבחרו בתלות במטרה של ניתוח המידע. משתמשים בניתוח אשכולות בבינה מלאכותית, בזיהוי תבניות, ניתוח תמונה ועוד ועוד.



איור 1 - סט של עצמים דומה או שונה בהתאם לתכונה הנבחרת [6].

ישנם סוגים רבים של אלגוריתמים שפותחו לניתוח אשכולות, בין היתר אלגוריתמים מבוססי קישוריות (שבבסיסם המרחק בין אובייקטים שונים), אלגוריתמים מבוססי התפלגות סטטיסטית או אלגוריתמים מבוססי תורת הגרפים (כך שכל קבוצה תיוצג כעץ שבו כל צומת מקושר לכל צומת אחר בעץ, ידועים גם כאלגוריתמים היררכיים) [13].

ניתן גם לחלק קיבוץ אשכולות לשני חלקים עיקריים: קיבוץ רך וקיבוץ קשה, על פי אופן שיוך אובייקט לקבוצה. בתיאור כללי, כמות הקבוצות שאליהן יכול אובייקט להיות מקשור משייכת את האלגוריתם לסוג הקיבוץ האשכולות (כאשר בקיבוץ אשכולות רך אובייקט יכול להיות מקושר ליותר מקבוצה אחת, בניגוד לקיבוץ אשכולות קשה) [5].

2.3 אלגוריתמים לניתוח אשכולות

משימת ניתוח האשכולות (חלוקת אובייקטים לקבוצות על פי תכונות דומות) ניתנת לביצוע על ידי שלל שיטות ואלגוריתמים. נבחר אלגוריתם מסוים על פי קלט הבעיה, סוגי האובייקטים ומטרת הניתוח לאשכולות. נעמיק בסוגי האלגוריתמים השונים ונפרט על הנפוצים בהם.

*ניתוח אשכולות היררכי*

אלגוריתמים היררכיים מאופיינים בחלוקה לקבוצות תחת ההנחה שאובייקטים קרובים יותר קשורים יותר זה לזה. למעשה הפיזור במרחב של האוביקטים הנתונים משפיע על החלוקה לאשכולות, ואשכול יאופיין על ידי המרחק בין שני האובייקטים הרחוקים ביותר באשכול. השם אלגוריתמים היררכיים מגיע מההירככיה של קבוצת החלוקות לאשכולות וכי באלגוריתמים אלו לא קיים פיתרון יחיד אלא רמות של פיתרונות, כאשר ניתן לפצל קבוצות או לאחד קבוצות קרובות וליצור סט של פתרונות. לפיכך ישנם אלגוריתמים היררכים אשר מתחילים מסט אובייקטים וממשיכים באיטרטיביות לאיחוד כל כמה אובייקטים לקבוצות ולאחר מכן קבוצות אלו לקבוצות גדולות יותר, או אלגוריתמים שמתחילים מסך כל האובייקטים כקבוצה ומנסים לפצלם בכל איטרציה לקבוצות קטנות יותר. בנוסף כל אלגוריתם היררכי מחשב באופן שונה את המרחקים במרחב בין אובייקטים ומוציא חלוקות שונות לאשכולות, על ידי שכלול פונקציות שונות כגון המרחקים במרחב, המרחק המקסימלי האפשרי, המרחב הממוצע ועוד [6].

ביצועי האלגוריתם מאופיינים בזמן ריצה פולימיאלי (לרוב באזור O(n^3) בהתאם לקלט), איטי יחסית לקלטים גדולים ולסוגי אלגוריתמים אחרים, אולם נפוץ הודות למגוון הפתרונות והגמישות [6].

נראה דוגמא לאלגוריתם היררכי איטרטיבי אשר מתחיל מסט של אובייקטים ומאחד אותם לקבוצות לעומת אלגוריתם אשר מתחיל מקבוצות גדולות ובכל איטרציה מפצל את הקבוצות לתתי קבוצות.

קלט התחלתי של 12 אובייקטים נפרדים (משמאל לימן):

🡪

🡪

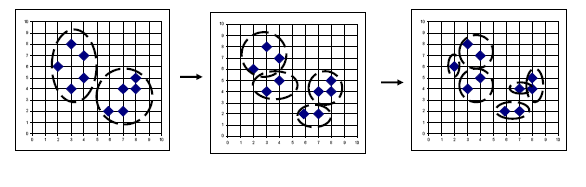
🡪

🡪

🡪

🡪

קלט התחלתי של 10 אובייקטים מחולקים לשני קבוצות:



איור 2 - חלוקת אובייקטים שונים לקבוצות [6].

סוגי האלגוריתמים ההירכיים אשר מפצלים אשכולות גדולים לתתי אשכולות (נקראים גם Top Down) נחשבים מורכבים יותר, יעילים יותר ומדויקים יותר (מכיוון שהם מסתכלים על "התמונה המלאה" – סט כל האובייקטים כקבוצה ולא על אובייקטים בודדים ואלו שלידם בלבד)

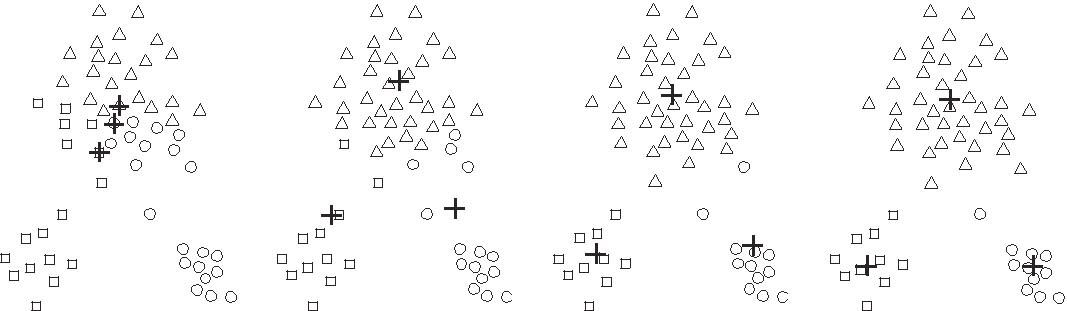
*אלגוריתמי ניתוח אשכולות מבוססי אב טיפוס (או מבוססי מרכז כובד)*

אלגוריתמים מבוססי מרחק (Centroid-based clustering) והנפוץ בהם K-Means clustering מבוססים על חילוק האובייקטים למספר נתון של אשכולות (k) לפי מרכזי כובד. מטרת האלגוריתמים היא למצוא את מרכזי הצברים כך שסכום המרחקים הריבועיים בין כל נקודה ומרכז הצבר אליו שויכה יהיה הקטן ביותר.

אלגוריתמים מסוג זה הינם יוריסטיים (ומבלי פונקציה יוריסטית משויכים לבעייות NP קשות), וידועים בביצועים של לפחות פתרון מינימום אופטימלי מקומי. במרבית המימושים מספר האשכולות מחושב מראש, וכן האשכולות הנוצרים שואפים למספר דומה של אובייקטים בתוכם.

אלגוריתם K-Means הינו איטרטיבי, כאשר בכול איטרציה בוחרים k אובייקטים בתור מדגם מתוך בסיס המידע שלנו להיות הצירים של החלוקה, כלומר מרכזי הכובד. לאחר מכן מקצים כל אובייקט למרכז הכובד הקרוב ביותר וכל אוסף של נקדה שהוקצה למרכז כובד נבחר התחלתי הוא אשכול. מרכז הכובד של כל אשכול מתעדכן בהתאם לאובייקטים המוקצים לו, לאחר מכן חוזרים באופן איטרטיבי על הצעדים עד שמרכז הכובד אינו משתנה יותר [6].

נראה דוגמא להרצה של מספר איטרציות על קלט התחלתי:



איור 3 - שימוש באלגוריתם K-Means למציאת k=3 אשכולות, משמאל לימין [13].

אנחנו רואים שלאורך האיטרציות מרכזי הכובד עוברים לשני האשכולות הקטנים בתחתית הקלט, ולאחר האיטרציה הרביעית מרכז הכובד נשאר קבוע והאלגוריתם מפסיק.

התוצאה של החישוב משתנה בהתאם לאתחול של האלגוריתם ולפיכך נהוג להריץ אותו מספר רב של פעמים (לקבל מגוון תוצאות רחב ולמצוא את האיזון בינהן). בנוסף, תהליך קביעת מספר האשכולות דורש חישובים רבים.

היבט חשוב נוסף באלגוריתמים אלו הוא חלוקת אובייקטים למרכז הכובד בצורה הטובה ביותר. ייתכן שיש מספר סוגים של חישוב קרבה שמתאימים לסוג מסוים של נתונים.

אלגוריתמים אלו נפוצים מאוד על כמויות גדולות של מידע ובתחומים של פילוח שוק, אסטרונומיה, חקלאות וכן הלאה.

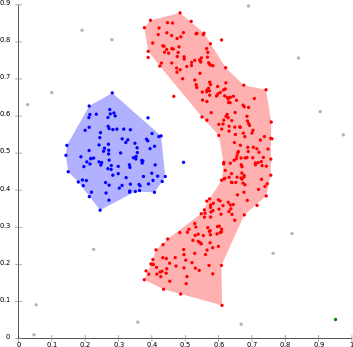
*ניתוח אשכולות מבוסס התפלגות*

מודלים של אלגוריתמי ניתוח אשכולות אשר נעזרים בשיטות סטטיסטיות מבוססים על מודלים של התפלגות כאשר אשכול מכיל אובייקטים בעלי ההתפלגות הכי דומה מבסיס המידע. בעוד הבסיס התיאורטי של שיטות אלה הוא מצוין, הם סובלים מבעיה מרכזית של התאמה מרובה מידי, כאשר מודל סטטיסטי מתאר שגיאה או רעש אקראי במקום הקשרים הרצויים, לדוגמא כאשר ישנם יותר מידי פרמטרים ביחס לכמות המידע, ובמקרים כאלו יהיו למודל יכולות חיזוי שגויות או מועטות. לפיכך אחת הבעיות בניתוח אשכולות מבוסס התפלגות היא בחירת מורכבות המודל [13].

*ניתוח אשכולות מבוסס צפיפות*

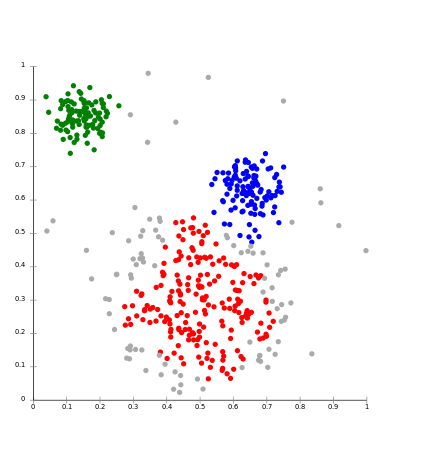
באשכולות מבוססי צפיפות, אשכולות מוגדרים כאזורים של צפיפות גבוהה יותר משאר בסיס המידע. למרות שאין גישות רבות להגדרת צפיפות כמו שיש להגדרת דמיון, קיימים מספר שיטות שונות. השיטה הפופולארית ביותר באשכולות מבוססי צפיפות היא אלגוריתםDBSCAN (Density-based spatial clustering of applications with noise). אלגוריתם זה מקבץ אובייקטים במקבץ שכנים צפוף (נקודות ליבה) ויוצר גבולות על ידי אזורים רדודים באובייקטים, והוא מבוסס חיבור אובייקטים בתוך ספי מרחק מסוימים. זמן הריצה של DBSCAN נמך למדי (פולונימיאלי בגודל הקלט, יכול להגיע עד O(n^2) כאשר n גודל הקלט). אולם החיסרון העיקרי של האלגוריתם הוא שהוא מצפה לירידה בצפיפות של אובייקטים כדי לזהות גבולות של אשכול מסוים, מה שלא תמיד ברור כל כך בסיסי המידע. OPTICS הוא אחד ההרחבות לDBSCAN שמתמודד עם גבולות טוב יותר על בסיסי נתונים שונים [13].

נראה דוגמא של חלוקה בסיס מידע לשני אשכולות על ידי DBSCAN כאשר הגבולות ברורים וישנה ירידה בצפיפות מסביב לשני האשכולות הנבחרים:



איור 4 - Density-based clustering with DBSCAN [13]

לעומת זאת, נראה דוגמא של בסיס מידע בו קשה יותר להפריד בין גבולות של אשכולות שונים ונראה כיצד ישנם אזורים אפורים בהם האלגוריתם DBSCAN לא הצליח לציין את סופם והתחלתם של אשכולות:



איור 5 - Density-based clustering with DBSCAN [13]

לסיכום, ישנם סוגים רבים ושונים של אלגוריתמי ניתוח אשכולות. כל אלגוריתם פועל באופן אופטימלי בתנאים שונים ובזמן ריצה שונה, וגם האשכולות הנקבעים שונים זה מזה. נדגים באיור אשר מראה תוצאות ריצה של שני אלגוריתמים שונים:



איור 6 - תוצאות ריצה של אלגוריתמי K-Means ו DBSCAN על בסיס מידע התחלתי, מתוך Cluster Analysis: Basic Concepts and Algorithms

3. קיבוץ אשכולות קשה

בפרק זה נתמקד בשיטת קיבוץ אשכולות קשה. בהגדרתו, קיבוץ אשכולות קשה (Hard Clustering) הוא קיבוץ אשכולות בו כל אובייקט שייך לאשכול אחד בלבד. במהלך הפרק נראה כי ישנם אלגוריתמים נפוצים מסוג זה, ונתמקד באלגוריתם K-Means, אחד הנפוצים מבינהם. בפרק הבא נחקור קיבוץ אשכולות קל, ונראה את אחד האלגוריתמים הנפוצים בראשו, Fuzzy C-Means.

3.1 מהו קיבוץ אשכולות קשה?

בפרק הקודם ראינו דוגמאות לאלגוריתמי קיבוץ שונים, ודמיון אחד בינהם הוא הפלט הסופי שלהם – קבוצה של אשכולות, כאשר לכל אשכול משויכים אובייקטים.

אולם, מעניין יהיה לדעת אם כל אובייקט מופיע אך ורק פעם אחת, או שהקיבוץ מאפשר לנו איזשהו אחוז קשירות לאשכול, כלומר אובייקט יוכל להופיע כמה פעמים, בכמה אשכולות שונים (פעם אחת בכל אשכול) לפי מידת שייכות כלשהי?

בדומה נוכל לשאול האם אדם משויך רק לקבוצה חברתית אחת על פי ניתוח חתימת העקב הדיגיטלית שלו? במקרה זה נאמר כי כנראה שלא, מכיוון שכל אחד מאיתנו מכיל הרבה יותר רבדים חברתיים מאשר כותרת אחת. בדומה, נוכל לשאול האם טקסט מסוים נכתב על ידי גבר או אישה? צעיר או בוגר? במקרים כאלו נאמר כי כנראה כל אובייקט של מידע ישויך לאשכול אחד, מכיוון שהסיכוי שאותו האובייקט ישויך ליותר מאשכול אחד נמוך יותר. בשביל לשייך אנשים למעגלים חברתיים, לדוגמא, נוכל להשתמש באלגוריתמי ניתוח אשכולות שונים, אשר ייבחרו הן על פי התאמתם לעולם הבעיה והן על פי שיוכים לניתוח אשכולות קשה או קל, כלומר האם הם ישייכו אובייקט אחד לאשכול אחד, או יאפשרו לאובייקט להופיע בכמה אשכולות שונים, במקרה שלנו אנשים שיקושרו למעגלים חברתיים שונים.

במידה ונבחר לקבץ את המידע שלנו לאשכולות בהם כל אובייקט מופיע רק פעם אחת, כלומר מצב שבו אין לנו חפיפה בין אשכולות שונים, משמע בחרנו בקיבוץ אשכולות קשה.

3.2 אלגוריתמים נפוצים בקיבוץ אשכולות קשה

האלגוריתמים הכי נפוצים בקיבכוץ אשכולות קשה הם מסוג K-Means. ישנם הרחבות רבות ואלגוריתמים משוכללים יותר תחת יסודו של K-Means, ולכן במהלך פרק זה נתמקד בגרסא הבסיסית של K-Means ונדבר על ההרחבות שלה בהמשך.

3.3 אלגוריתם K-Means

כפי שהזכרנו בפרק הקודם, K-Means (אשר לעיתים קרובות הוא גם נקרא האלגוריתם של לויד על שם סטיוארט לויד, אשר תיאר את האלגוריתם ב1957) הוא האלגוריתם הנפוץ ביותר מסוג אלגוריתמים מבוססי מרחק. בניסוח חופשי, נרצה לשייך כל אובייקט לאשכול על פי הקרבה שלו לשאר איברי האשכול ולמעשה על פי הקרבה שלו למרכז הכובד של האשכול. אלגוריתם זה מבוסס על חילוק האובייקטים למספר של אשכולות לפי מרכזי כובד, כאשר אנחנו לא יודעים מי הם מרכזי הכובד שלנו בהתחלה וכפי שצוין בפרק הקודם הוא אלגוריתם יוריסטי ששואף לפונקציית מינימום (מקומית לפחות) [5].

3.4 יתרונות וחסרונות האלגוריתם

לאלגוריתם זה ישנם יתרונות רבים:

* מאפשר להתמודד עם נתונים ללא התפלגות מוגדרת.
* מאפשר להתמודד עם מצב שבו בתצפית מסוימת חסרות חלק מהמדידות של משתני הכניסה.
* זמן עיבוד תשובה קצר ביותר.
* מאפשר לבצע לימוד "בלתי מונחה.
* אלגוריתם זה תמיד ימצא פתרון כל שהוא.
* אחרי שהוגדרו האשכולות ניתן לעבוד ללא נתוני המקור.

והן חסרונות:

* אלגוריתם זה מחייב עיבוד מוקדם ולעיתים מרובה
* אלגוריתם זה מושפע מבחירה של גבולות המשתנים
* אלגוריתם זה אינו מאפשר לטפל בנתונים שאינן מספריים ואינם אורדינליים
* לעיתים הפלט של האלגוריתם אינו פתרון יציב
* אלגוריתם זה תלוי במצב ההתחלתי של המידע
* אלגוריתם זה רגיש לגבולות (מרכזי הכובד)
* עלינו לקבוע מראש מהו הk שלנו, ובהתאם אליו תשתנה אופטימליות האלגוריתם

3.5 הדגמת האלגוריתם

מטרת האלגוריתם היא למצוא את מרכזי הכובד האופטימליים כך שסכום המרחקים הריבועיים בין כל נקודה ומרכז הכובד אליו שויכה, כלומר האשכול שלה, יהיה הקטן ביותר.

ככלל, האלגוריתם ינוע לסירוגין בין שני שלבים הבאים:

### שלב הקצאה: עלינו להקצות לכל אובייקט את אחד האשכולים. בהמשך נראה כיצד אנו מחליטים לאיזה אשכול ישויך כל אובייקט.

### שלב עדכון: עלינו לחשב את מרכזי הכובד החדשים כדי להיות במרכז הכובד הגאומטרי של האובייקטים באשכולות החדשים.

האלגוריתם עוצר כאשר לא ניתן לשנות עוד את הנתונים, ומתקבל ערך קבוע של מרכזי כובד.

שלבי האלגוריתם:

1. נבחר באקראי k אובייקטים שיהוו מרכזי כובד מתוך האובייקטים שניתנו כקלט התחלתי.
2. נשייך כל נקודה למרכז הקרוב אליה ביותר, על ידי פונקציית המרחק בין נקודות (הנורמה של ווקטור בין שני נקודות).
3. מעדכנים כל אשכול להיות הממוצע של כל הנקודות ששויכו לאשכול, על ידי חישוב המרחקים בין כל הנקודות וחלוקה במספר הנקודות לקבלת ממוצע.
4. נחזור על צעדים 2-3, עד שאין שינוי באף אחד ממרכזי הכובד.

המודל שואף לחלק את n האובייקטים (תצפיות) שלנו  ל- (k=<n) S={S1, S2,….Sk} כדי למזער את סכום  הריבועים בתוך האשכול [5].

ניתוח זמן ריצה:

זמן ריצת האלגוריתם תלוי בגורמים הבאים:

N – גודל המידע ההתחלתי, מספר האובייקטים

K – מספר האשכולות הרצוי (נתייחס אליו כאל קבוע)

I – מספר האיטרציות

D – מספר המשתנים שאנו מכירים על המידע

ממבט על האלגוריתם, נסיק כי זמן הריצה הוא מכפלה של i מכיוון שבכל איטרציה אנחנו עושים כמות עבודה דומה, והיא בעצמה תלויה בn, גודל הקלט ההתחלתי, בk, הקבוע שלנו של מספר האשכולות וכן ב d, מספר המשתנים שיש לנו באובייקטי המידע ובו נשתמש לחישוב המרחקים לאשכולות. לפיכך, זמן ריצת האלגוריתם היא O(n \* k \* i \* d) [4].

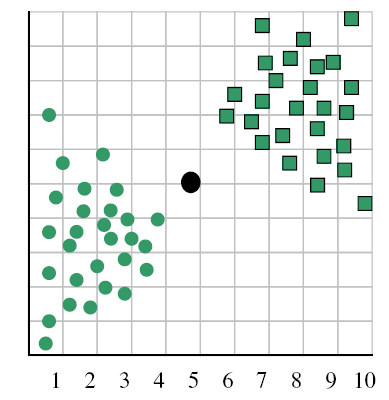
### שיטות אתחול:

שיטות אתחול בשימוש הנפוצה הן פורגי וחלוקה אקראית. שיטת פורגי בוחרת באופן אקראי k אובייקטים מהמערך הנתונים (כפי שראינו בשלב 1 בהדגמת האלגוריתם) ומשתמשת בהם כאמצעי הראשוני. שיטת החלוקה אקראית מבצעת באופן אקראי הקצאת אשכול לכל אובייקט (כלומר מבצעת את שלבים 1 + 2 בהדגמת האובייקט), ולאחר מכן ממשיכה לשלב העדכון, ובכך שהחישוב הממוצע הראשוני מהווה מרכז כובד של הנקודות באופן אקראי.

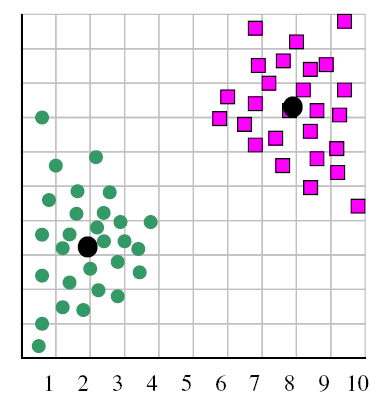
בחירת הk:

נראה כיצד בחירת הk שלנו (אשר עלינו לקבוע טרם התחלת הרצת האלגוריתם) משפיעה על פלט האלגוריתם, עבור קלט מסוים של אובייקטים:

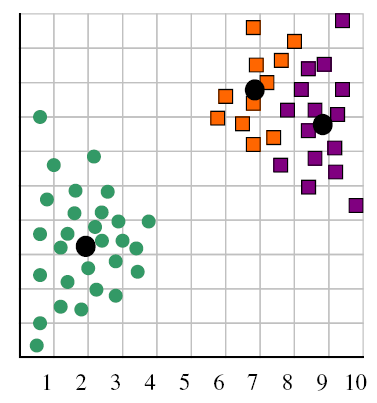
כאשר k=1, מרכז הכובד יהי 873:



כאשר k=2, מרכז הכובד ירד ל 173:



וכאשר k=3, מרכז הכובד יהיה 133:



איור 7 - בחירת מספר האשכולות,מתוך Cluster Analysis: Basic Concepts and Algorithms

כפי שראינו בתמונות, השינוי המשמעותי עבור בחירת k היה מk=1 ל k=2, ואז ככל שנעלה במספר הkים עקומת השינוי תתייצב. בחירת הk שלנו יכולה להשפיע משמעותית על "איכות" המחיצות בין המידע שלנו לאשכולות, ובפרקים הבאים נראה גם כי בחירת הk שלנו משפיעה על יעילות וזמן ריצת האלגוריתם.

הדגמת הרצה של אלגוריתם K-Means

נראה הדגמת הרצה של האלגוריתם, ניעזר בדוגמא מתוך האתר Mnemosyne\_studio.

האובייקטים ההתחלתיים שלנו יהיו הציונים עבור כל אחד משני משתנים, ב7 קטגוריות שונות

|  |  |  |
| --- | --- | --- |
| קטגוריה | משתנה A | משתנה B |
| 1 | 1.0 | 1.0 |
| 2 | 1.5 | 2.0 |
| 3 | 3.0 | 4.0 |
| 4 | 5.0 | 7.0 |
| 5 | 3.5 | 5.0 |
| 6 | 4.5 | 5.0 |
| 7 | 3.5 | 4.5 |

נבקש שk יהיה 2, כלומר נבקש פלט של שני אשכולות. למען מציאת חלוקה התחלתית טובה, נבקש שערכי הA וערכי הB של שני הציונים הכי רחוקים יהיו נקודות הציר שלנו, כלומר מרכזי הכובד ההתחלתיים.

|  |  |  |
| --- | --- | --- |
|  | אובייקט | מרכזי הכובד |
| אשכול 1 | 1 | (1.0, 1.0) |
| אשכול 2 | 4 | (5.0, 7.0) |

נעבור על שאר המידע ונקצה אותו לאחד משני האשכולות, בהתאם לקרבתם למרכזי הכובד שקבענו בהתחלה. מרכז הכובד מחושב בכל פעם בה אובייקט חדש מתווסף לאשכול. נראה את רצף הצעדים:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | אשכול 1 | | אשכול 2 | |
| מספר צעד | אובייקט | מרכז כובד | אובייקט | מרכז כובד |
| 1 | 1 | (1.0, 1.0) | 4 | (5.0, 7.0) |
| 2 | 1, 2 | (1.2, 1.5) | 4 | (5.0, 7.0) |
| 3 | 1, 2, 3 | (1.8, 2.3) | 4 | (5.0, 7.0) |
| 4 | 1, 2, 3 | (1.8, 2.3) | 4, 5 | (4.2, 6.0) |
| 5 | 1, 2, 3 | (1.8, 2.3) | 4, 5, 6 | (4.3, 5.7) |
| 6 | 1, 2, 3 | (1.8, 2.3) | 4, 5, 6, 7 | (4.1, 5.4) |

כעת החלוקה ההתחלתית השתנתה, ושני האשכולות נראים ככה:

|  |  |  |
| --- | --- | --- |
|  | אובייקט | מרכז כובד |
| אשכול 1 | 1, 2, 3 | (1.8, 2.3) |
| אשכול 2 | 4, 5, 6, 7 | (4.1, 5.4) |

כעת עלינו לוודא שכל אובייקט הוקצה לאשכול הנכון. לכן כעת נשווה את המרחק בין כל אובייקט לאשכול שלו, ולאשכול השני. להלן הממצאים:

|  |  |  |
| --- | --- | --- |
| אובייקט | מרחק למרכז הכובד של אשכול 1 | מרחק למרכז הכובד של אשכול 2 |
| 1 | 1.5 | 5.4 |
| 2 | 0.4 | 4.3 |
| 3 | 2.1 | 1.8 |
| 4 | 5.7 | 1.8 |
| 5 | 3.2 | 0.7 |
| 6 | 3.8 | 0.6 |
| 7 | 2.8 | 1.1 |

רק אובייקט 3 היה קרוב יותר לאשכול השני (מאשר לאשכול אליו הוקצה מלכתחילה). בשלב זה ווידאנו כי המרחק בין כל אובייקט לאשכול שלו הוא המינימלי מבין המרחקים לכל שאר האשכולות ,במקרה שלנו רק לאשכול הנוסף הקיים. ולכן אובייקט 3 מועבר לאשכול מספר 2. זוהי החלוקה העדכנית:

|  |  |  |
| --- | --- | --- |
|  | אובייקט | מרכז כובד |
| אשכול 1 | 1, 2 | (1.3, 1.5) |
| אשכול 2 | 3, 4, 5, 6, 7 | (3.9, 5.1) |

החלוקה האיטרטיבית תמשיך ממצב החלוקה הנוכחי (על ידי חישוב המרחקים מחדש וחיפוש מחדש של אובייקטים שקרובים יותר לאשכול אחר שאינם מוקצים אליו כעת) עד שלא נמצא עוד אובייקטים אשר קרובים יותר לאשכול אחר. בדוגמא זו, הגענו למצב הסופי מכיון שכל אובייקט נמצא באשכול אליו הוא הכי קרוב וזוהי תהיה החלוקה הסופית שלנו.

כפי שהזכרנו בהתחלה, אלגוריתם K-Means יכול להגיע לתוצאות של מינימום מקומי בלבד ולהתקע במצבים בהם אף פעם לא נגיע להזזה של אובייקט לאשכול אחר שיגרום לחישוב מרכז הכובד מחדש ויחלק מחדש את כל שאר האובייקטים בצורה אופטימלית יותר. לכן נרצה גם לעצור את מספר האיטרציות בשלב מסוים כדי להימנע מלופים שחוזרים על עצמם ולהריץ אותו מחדש עבור חלוקת מרכז כובד התחלתית שונה. אולם, אלגוריתם K-Means גם יגיע לעיתים למינימום מוחלט ומכיוון שהוא יוריסטי עבור מספר חזרות רב נגיע לממצאים מצויינים ואופטימליים.

4. קיבוץ אשכולות קל

בפרק זה נתמקד בשיטת קיבוץ אשכולות קל. בניגוד לקיבוץ אשכולות קשה, אותו ראינו בפרק 3, קיבוץ אשכולות קל (Soft Clustering) הוא קיבוץ אשכולות בו כל אובייקט שייך לכמה אשכולות שונים על פי דרגת קרבה, כל אובייקט מקסימום פעם אחת בכל אשכול. במהלך הפרק נראה כי ישנם אלגוריתמים נפוצים מסוג זה, ונתמקד באלגוריתם Fuzzy C-Means, הנפוץ מבינהם. בפרק הבא נעמוד על השינוי בין קיבוץ אשכולות קל לקיבוץ אשכולות קשה.

4.1 מהו קיבוץ אשכולות קל?

כפי שהזכרנו בעבר, מטרתו של ניתוח אשכולות היא לחלק סט של N אובייקטים / אלמנטים לתוך C אשכולות כך שאובייקטים בתוך אותו האשכול יהיו דומים יותר זה לזה מאשר לאובייקטים אחרים באשכולות אחרים. ניתוח אשכולות מאפשר לנו למצוא מבנים או תבניות בתוך כמויות מידע, כאשר האובייקטים בכל אשכול מראים דרגה כלשהי של דומות. זאת נקבל על ידי מגוון רחב של אלגוריתמים. אלגוריתמים אלו שונים זה מזה לפי הגדרתם למה מוכל בתוך אשכול מידע ומהי הדרך היעילה ביותר לקבל אשכולות אלו. לדוגמא, באלגוריתמים מסוג קיבוץ אשכולות רך, אלמנטי נתונים משויכים ליותר מאשכול אחד, וכל אלמנט מזוהה עם סט של רמות קרבה לאשכולות. שימוש באלגוריתמים של קיבוץ אשכול רך הפכו לאבן יסוד בהתמודדות עם האתגרים של עיבוד כמויות עצומות של מידע גולמי. הרעיון המרכזי בגישה הוא הויתור על החד ערכיות של אובייקט לאשכול. למעשה, ברוב המקרים קיבוץ רך הרבה יותר טבעי מאשר קיבוץ קשה ואובייקטים על קו התפר בין אשכולות שונים לא משוייכים בכוח לאשכול אחד בלבד [8]

4.2 אלגוריתמים נפוצים בקיבוץ אשכולות קל

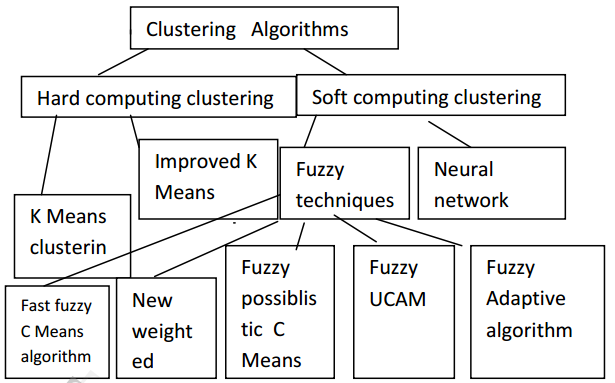
אלגוריתמים של קיבץ אשכולות רך יכולים להתחלק לשני קבוצות עיקריות: קיבוץ קלאסי (Classical Fuzzy clustering algorithms) וקיבוץ מונחה צורה (Shape based Fuzzy clustering) [8].

נמשיך ונחלק גם קבוצות עיקריות אלו לתתי קבוצות:

נושיב תחת קיבוץ קל קלאסי שלושה אלגוריתמים עיקריים: אלגוריתמים מסוג Fuzzy C-Means, אלגוריתם Gustafson-Kessel וכן אלגוריתם Gath-Geva.

תחת קיבוץ מונחה צורה נושיב את האלגוריתמים העיקריים הבאים: אלגוריתמים מעגליים, אלגוריתמים אליפטיים, ואלגוריתמיים בצורות גנריות של ניתוח אשכולות.

בתרשים הבא נוכל לראות את החלוקה הגסה בין קיבוץ קשה לקיבוץ קל ואת האלגוריתמים העיקריים תחתיהם, כאשר תחת קיבוץ קשה יושבים אלגוריתמי K-Means כפי שהזכרנו בפרק הקודם, ותחת קיבוץ קל יושבים אלגוריתמים כגון Fuzzy C-Means [5].



איור 8 - תתי קבוצות של אלגוריתמים מחולקים לפי קיבוץ קל וקיבוץ קשה, [5]

נתעמק בקבוצת אלגוריתמים הנקראת Fuzzy techniques (אלגוריתמים עמומים) ובסעיף הבא נתמקד באופן פרטני באלגוריתם Fuzzy C-Means. אלגוריתמים אלו הם שיטה מאורגנת להתמודדות עם נתונים לא מדויקים .נתוני הקלט מהם ניצור אשכולות נקראים Fuzzy set.

נעבד נתונים אלו באופן שמאפשר אפשרות שייכויות חלקיות לקבוצות במקום סט קבוצה מראש של נתונים [5].

4.3 אלגוריתם Fuzzy C-Means

אחד האלגוריתמים הנפוצים ביותר של קיבוץ אשכולות קל הוא Fuzzy C-Means אשר הוצע לראשונה בספרות על ידי ג'ו דאן (Joe Dunn) בשנת 1974 ובמקביל בגרסא כללית יותר שלו על ידי ג'ים בזדק (Jim Bezdek) בעבודת הדוקטורט שלו באוניברסיטת קורנל ב 1973. באלגוריתם Fuzzy C-Means, אובייקט נתונים יכול להשתייך לכל האשכולות עם ציוני שייכות שונים בין 0 ל -1 [8].

אלגוריתם זה עובד על ידי הקצאת שייכות לכל אובייקט מידע המתאימות לכל מרכז כובד של אשכול על בסיס המרחק בין מרכז האשכול ואובייקט המידע. ככל שהאובייקט סמוך יותר למרכז האשכול כך הוא שייך יותר לאשכול. סכימה של ציוני השייכות של כל אובייקט במטריצת רמות השייכות צריכה להיות שווה לאחד. לאחר כל איטרציה, האלגוריתם מחשב מחדש את מרכזי הכובד וציוניי השיכות של אובייקט ואשכול. בניגוד לאלגוריתמים מסורתיים יותר כגון K-Means אשר יוצרים מחיצות ברורות בין הנתונים, Fuzzy C-Means יוצר רמות שייכות שונות על ידי פונקציית שייכות [5][4].

4.4 יתרונות וחסרונות האלגוריתם

יתרונות האלגוריתם [11]:

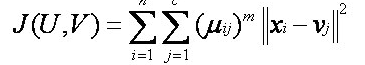
* האלגוריתם מתכנס לתוצאה סופית
* האלגוריתם מביא פתרון אופטימלי עבור קלט של נתונים עם חפיפות

חסרונות האלגוריתם:

* זמן חישוביות ארוך יחסית
* בעל רגישות לאיתחול הראשוני (מהירות ואופטימליות הפתרון מושפעים מכך)
* רגישות ל"רעש" (נתונים חריגים או בעלי סטיות תקן אשר מהווים רעש לקו האשוכולות אליו הפתרון ישאף בלעדיהם ומטים את הפתרון לכיוונים שונים)

4.4 הדגמת האלגוריתם

מטרת האלגוריתם היא להגיע למינימום של הפונקציה הבאה:



כאשר n מספר הנקודות כפי שהתקבלו בקלט, c מספר האשכולות הסופיים (קבוע שנקבע בתחילת הרצת האלגוריתם), m € [1, ∞], והנורמה של xi – vj זהו המרחק האוקלידי בין הנקודה i לאשכול j [11].

שלבי האלגוריתם יהיו כדלקמן:

קלט:

 X = {x1, x2, x3 ..., xn}סט הנקודות המהוות את בסיס הנתונים

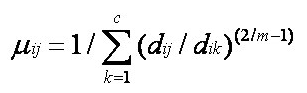
V = {v1, v2, v3 ..., vc} סט האשכולות, בתחילת האלגוריתם ריק

J – מספר מקסימלי של איטרציות

m € [1, ∞]

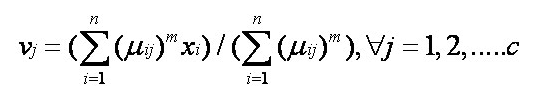
מהלך האלגוריתם [5][11]:

1. בחר c ערכים רנדומאליים, אחד עבור כל אשכול
2. חשב את רמת השייכות µij אשר מוגדרת באופן הבא:



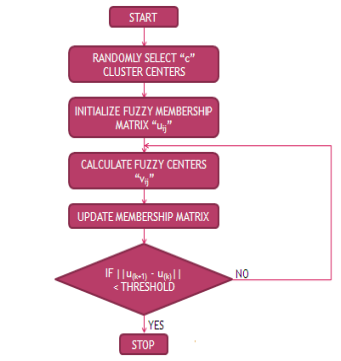
כאשר d הוא ערכו של '||xi – vj||' .

1. חשב את מרכזי הכובד של האשכולות על ידי הפונקציה הבאה:



1. חזור על שלבים 2 ו3 מקסימום J פעמים, עד אשר *||U(t+1)- U(t)|| < β* כאשר U זהו מערך דו מימדי של רמות השייכות של כל נקודה לכל אשכול *U = (µij)n\*c,* t זהו מספר האיטרציה, *0<β****<1***. בשלב זה נבדוק שההפרש בין ערכי השייכות של הנקודות לאשכולים באיטרציה זו ובין האיטרציה הקודמת קטנים מ *β* קבוע.

נציג תרשים קצר שמתאר את אלגוריתם Fuzzy C-Means.



איור 9 - תרשים סכמטי של אלגוריתם Fuzzy C-Means [10]

ראינו שאלגוריתם Fuzzy C-Means מנסה לשפר בכל איטרציה את האיתחול הראשוני שלו על ידי חישוב כל המרחקים האפשריים בין כל נקודה לבין כל אשכול ושמירת דרגת השייכות במטריצה. האלגוריתם יעצור כאשר נגיע לרמת יציבות מספיק גבוהה כך שהשוני בין כל איטרציה יהיה זניח.

ניתוח זמן ריצה:

זמן ריצת האלגוריתם תלוי בגורמים הבאים:

N – גודל המידע ההתחלתי, מספר האובייקטים

C – מספר האשכולות הרצוי (נתייחס אליו כאל קבוע)

I – מספר האיטרציות

D – מספר המשתנים שאנו מכירים על המידע

ממבט על האלגוריתם, נסיק כי זמן הריצה הוא מכפלה של i מכיוון שבכל איטרציה אנחנו עושים כמות עבודה דומה, והיא בעצמה תלויה בn, גודל הקלט ההתחלתי, ב d, מספר המשתנים שיש לנו באובייקטי המידע ובk, הקבוע שלנו של מספר האשכולות בריבוע, מכיוון שניצור מכפלה קרטית בין כל האשכולות. לפיכך, זמן ריצת האלגוריתם היא O(n \* c^ 2 \* i \* d) [4].

1. השוואת קיבוץ קשה וקיבוץ רך

התוצאה והייעילות של תהליך קיבוץ האשכולות נקבעים בדרך כלל באמצעות האלגוריתמים בהם אנחנו משתמשים. ישנם אלגוריתמים רבים ושונים בהם משתמשים כדי לפתור בעיה זו. בפרק זה נציג ונשווה בין שני אלגוריתמי אשכולות משני סוגים עיקריים וחשובים: אלגוריתמי קיבוץ קשה ואלגוריתמי קיבוץ קל, ובראשם K-Means וFuzzy C-Means אותם ראינו בפרקים הקודמים. בפרק זה נעמוד על השוני בין הביצועים והיעילות של אלגוריתמים אלו.

5.1 מאפיינים עיקריים

פלט האלגוריתם

כפי שראינו, אלגוריתם K-Means מפיק מערך בעל k תאים, כאשר כל תא מייצג אשכול ואת הנקודות (אובייקטים) המקושרים לאשכול.

אלגוריתם Fuzzy C-Means מפיק מטריצת שייכויות בין כל נקודה במישור לבין כל אשכול במישור (חלק מערכי רמות השייכות יכולים להיות בעלי הערך 0).

שלבי הרצת האלגוריתמים

ראינו כי שני האלגוריתמים מתחילים ע ם k מספר אשכולות רצוי קבוע, ואיתחול רנדומאלי התחלתי של מרכזי הכובד של כל אשכול.

כמו כן, שני האלגוריתמים רצים באיטרציות ומחשבים מחדש את ערכי המשתנים שלהם (מרכזי כובד ומרחקים מכל נקודה עבור K-Means ורמות שייכות בFuzzy C-Means), ומפסיקים את האיטרציות עבור תנאי מסוים.

באלגוריתם K-Means התנאי בודק שבאיטרציה האחרונה לא הצלחנו לשנות שום מרכז כובד או לשייך נקודה לאשכול אחר (ולכן בכל איטרציה אחריה התוצאות ימשיכו להישאר קבועות), ואילו עבור Fuzzy C-Means נבדוק שההפרש בין רמות השיכות של האיטרציה הנוכחית לאחרונה קטן מקבוע קטן כלשהו, וכן שלא עברנו t איטרציות. הסיבה להגבלה על t נובעת מזמן הריצה הארוך יותר של Fuzzy C-Means שיכולה לגרום לכמות איטרציות ששואפת לאינסופית במידה ולא נצליח להגיע לערכים מתאימים. [11] [8] [4].

שיטת חישוב מרחקים בין אובייקטים ואשכולות:

בשני האלגוריתמים נשתמש בחישוב שייכות לאשכול (מציאת האשכול הכי מתאימה לנקודה מסוים בK-Means או מציאת רמת שייכות עבור Fuzzy C-Means) נחושב על ידי המרחק האוקלידי במישור בין הנקודה לבין מרכז הכובד של האשכול.

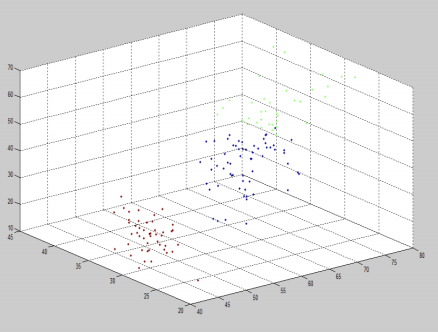
אופטימליות פתרונות האלגוריתמים:

ראינו בפרקים הקודמים כי שני האלגוריתמים שואפים למציאת מינימום מקסימלי לפונקציית הפתרון שלהם. כמו כן, ציינו בעבר כי מספר האשכולות הנבחר להרצה, ובחירת האשכולות הראשוניים בצורה ראנדומאלית משנים את הפתרון של האלגוריתם ויכולים למנוע ממנו למצוא את הפתרון האופטימלי. ראינו שניתן להתמודד עם בעיה זו על ידי הרצה מרובה של האלגוריתמים עבור מספר אשכולות שונה ועבור בחירת אשכולות ראשוניים שונה (בחירת האשכולות הראשוניים ראנדומאלית ולכן בכל הרצה נקבל תוצאות שונות). בכך, שני האלגוריתמים דומים.

אולם, מקובל להגביל את מספר האיטרציות המקסימלי של אלגוריתמים Fuzzy C-Means, שכן מספר האיטרציות שלו לרוב גדול יותר ממספר האיטרציות על המידע של K-Means [4].

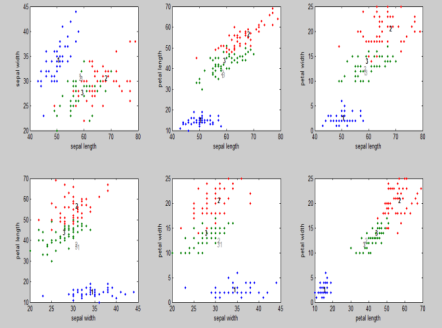
השוואת זמני הריצה של שני האלגוריתמים:

לצורך השוואת זמני הריצה של שני האלגוריתמים, נציג תוצאות הרצה של שני האלגוריתמים בmatlab כפי שהורצו והוצגו במאמר International Journal of Advanced Computer Science and Applications [4], בשימוש בסביבת העבודה UCI Machine Learning, מאגר של מאגרי מידע רבים שנועדו כסביבת הרצה במחקרים אודות בינה מלאכותית וכריית מידע. בהרצות אלו השתמשו במאגר המידע Iris plant Dataser (מאגר מידע אוודת פרחי איריס) אשר מורכב מ5 משתנים שונים: ארבעה מהם מספריים (אורך עלה גביע, רוחב עלה גביע, אורך הכותרת ורוחב הכותרת), והחמישי אינו מספרי ומורכב משלוש מחלקות (Iris Setosa, Iris Versicolour, Iris Virginica). המחלקה הראשונה לינארית ושני המחלקות האחרות לא נפרדות באופן לינארי אחת מהשנייה.

עבור אלגורתם K-Means (כאשר למדידת המרחקים השתמשנו במרחק האוקלידי), לאחר 13 איטרציות של חישוב ועדכון מרכזי הכובד של האשכולות, הגענו לסכום מרחקים כולל של 7897.88, תוך 0.44 שניות.

איור 10 - חלוקה ל3 אשכולות על ידי אלגוריתם K-Means כפי שהורץ על מאגר פרחי איריס [4].

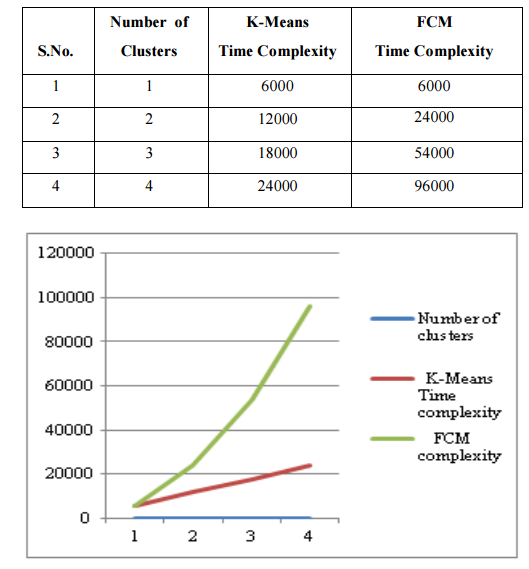
עבור אלגורתם Fuzzy C-Means (האלגוריתם מקבל מספר רצוי של אשכולות ומחזיר מרכזי כובד אופטימליים וציוניי שייכות של אובייקט לאשכול), לאחר 30 איטרציות, הגענו לערך מינימלי של 6058.68, תוך 0.781 שניות.



איור 11 - חלוקה ל3 אשכולות על יד אלגוריתם Fuzzy C-Means כפי שהורץ על מאגר פרחי אירוס, בתחילת ההרצה במצב האיתחול הראשוני ובסוף ההרצה [4]

מממצאי ניסוי זה אלגוריתם K-Means (שהוא אלגוריתם קיבוץ אשכולות קשה) רץ מהר יותר מאלגוריתם Fuzzy C-Means (אלגוריתם קיבוץ אשכולות קל). תוצאות אלו נתמכות על ידי ניתוח זמני הריצה של האלגוריתמים כפי שראינו בפרקים הקודמים.

נמחיש מסקנות אלו באמצעות הגרף הבא:



איור 12 - זמן ריצה משוער עבור אלגוריתמי ניתוח אשכולות K-Means ו Fuzzy C-Means, בהינתן k מספר אשכולות משתנה [4].

שיערנו שההבדל בין זמני הריצה של שני האלגוריתמים הוא סדר גודל של מכפלה של גודל המידע במספר האשכולות הרצוי לעומת מכפלה של גודל המידע במספר האשכולות הרצוי בריבוע. לפיכך עבור k =1 מספר אשכולות נשער כי לא נראה הבדלים בזמני הריצה בין שני האלגוריתמים, אולם עבור k=4 לדוגמא, ההבדל בין זמני הריצה קופץ פי 4 ומהווה עליה משמעותית באלגוריתם Fuzzy C-Means [4].

5.1 יתרונות וחסרונות

כפי שהזכרנו בעבר, המוטיבציה לשימוש בקיבוץ אשכולות קל או בקיבוץ אשכולות קשה מגיעה מעולם הבעיה. אם אנחנו מתמודדים עם בסיס המידע שכל נקודה בו שייכת לאיזשהו חלק בודד, וניתן לגדר אותה באופן ברור עם נקודות אחרות כדמותה, נבחר בK-Means. במידה ואנחנו מתמודדים עם כמויות רבות של מידע בהן אין שיוך חד ערכי לכל נקודה אלא "רמות קרבה והשערה של שייכות", נבחר בFuzzy C-Means.

כמו כן, נוכל להסיק מהמאפיינים של שני האלגוריתמים ששניהם רגישים לנקודות "רעש" – נקודות בטווח הטעות אשר אינן נמצאות באותו המקום במישור עם נקודות דומות מסוגן, ולכן משנות את מרכז הכובד המחושב של כל אשכול. כלומר, שני האלגוריתמים יפיקו תוצאות עם סטיות תקן במידה והקלט מכיל נקודות "רעש". אולם, קיימים אלגוריתמים שונים מסוגי קיבוץ אשכולות קל וקיבוץ אשכולות קשה אשר פחות רגישים לנקודות רעש ומכך נסיק שלא ניתן להתייחס לכך כאל חסרון או יתרון בכל אחת משיטות הקיבוץ.

בנוסף, ראינו כי זמן הריצה של אלגורתם Fuzzy C-Means ככלל ארוך יותר מזמן הריצה ש אלגוריתם K-Means (ובפרט באופן לינארי עבור גודל הקלט n שמייצג את מספר נקוודת המידע). לכן, כאשר עולם הבעיה יאפשר גם אלגוריתם קיבוץ קשה (ולא ידרוש מאיתנו את התמונה המלאה של שייכות כל נקודה לכל אשכול), נבחר באלגוריתם קיבוץ קשה (ובמידה ובחרנו באלגוריתם קיבוץ קשה אחר שאינו K-Means, הבחירה תצדיק את עצמה בהנחה שזמני הריצה שלו באותו סדר גודל של זמני הריצה של K-Means)

חסרון נוסף לאלגוריתם Fuzzy C-Means הוא הצורך להעביר מספר איטרציות מקסימלי (בכדי להימנע מאיטרציות אינסופיות על הקלט). דבר זה ידרוש מאיתנו לבדוק מהו מספר איטרציות הולם עבור הקלט לאופטימליות הפלט ויוכל להשפיע על איכות הפלט.

5.2 יישומים

קיבוץ אשכולות הוא אחד מהתחומים פורצי הדרך של כריית מידע וכיום משולבים במרבית האזורים בתעשייה. אם נסתכל על האלגוריתמים Fuzzy C-Means ו K-Means, נראה שמשתמשים בהם עבור מגוון רחב של בעיות כגון עיבוד תמונה, תבניות ברצפים ביולוגיים, זיהוי רשתות, חיזוי שימוש ושימוש חוזר באפליקציות ועוד ועוד ועוד.

לאורך המאמרים שקראתי, ראיתי שהעתיד של קיבוץ אשכולות קל ובפרט לאלגוריתם Fuzzy C-Means צופה:

* השמשת האלגוריתם על ניתוח סוגי וכמויות הפעולות שמתבצעות באינטרט
* השתלבות בחיזוי תנועה עתידי (חיזוי מצב הכבישים עד כשעה קדימה עוד לפני שתחזית מסוימת מתחילה מתרחשת)
* הרחבת אלגוריתם Fuzzy C-Means עבור ניתוח תמונה של תמונות שרופות בקצוות
* השמשת האלגוריתם בניתוח רקמת מוח מתמונות MRI
* חיזוי טמפרטורות באזורים שונים בכדור הארץ

ובדומה גם ניתוח אשכולות קשה מתפתח ומתרחב וכעת עובדים על:

* חיזויים פיננסיים
* זיהוי פנים מבוססים K-Means
* חיזוי הצלחה מקצועית ואקדמית של סטודנטים

6. ניתוח תמונה

בשנים האחרונות, עיבוד תמונה הפך להיות תחום מתרחב והצורך בו הפך ליום יומי. אם נחשוב על הדברים הכי בסיסיים שמוכרים לנו כיום, כגון ניתוח תמונות MRI, גרפיקה במשחקי מחשב, זיהוי פנים של פייסבוק ועוד ועוד ועוד, נבין מהו הצורך הרב שלנו בניתוח כמויות מידע עצומות אודות תמונות בעזרת כריית מידע ובפרט בניתוח אשכולות. בפרק זה נראה מהם ההתקדמויות והבעיות השונות בנושא ניתוח תמונה, וכיצד זה משתנה באלגוריתמי קיבוץ אשכולות קל לעומת קיבוץ אשכולות קשה.

6.1 ניתוח תמונה בעזרת ניתוח אשכולות

כפי שהזכרנו מעלה, עיבוד תמונה עבר תהפוכות רבות בניתוח אשכולות. כל תמונה היא עולם ומלואו של אלפי נקודות בצבעים שונים, וכל אלגוריתם בתחום מציג יתרונות וחסרונות משלו. קיבוץ אשכולות בניתוח תמונה התחיל בתמונות שחורות, לבנות ואפורות והתרחב הלאה לתמונות צבעיוניות (בעיקר בשיטת RGB - Red (R), Green (G) , Blue (B), שלושת הצבעים הבסיסיים אדום כחול וירוק כמשתנים של הנקודות) [3].

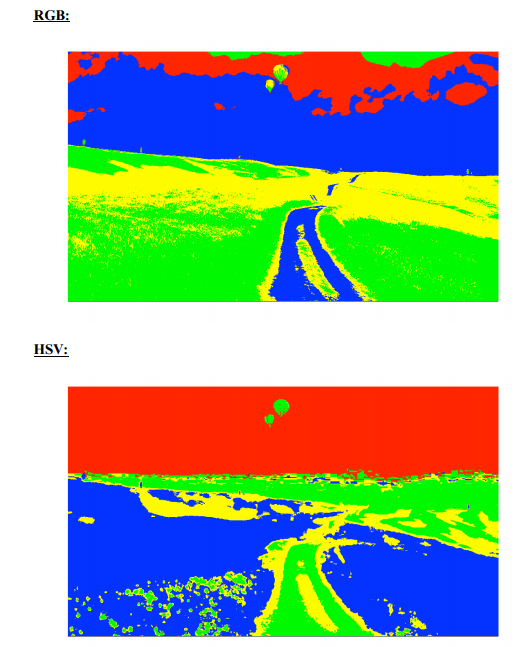
ישנם שתי גישות עיקריות להתייחסות לתמונה אותה נרצה לנתח:

* ניתוח מונחה ניגודיות – ניתוח שמשתמש בשוני ובהבדלין בין הניגודיות והבהירות של פיקסלים בתמונה [2]
* ניתוח מונחה צורה – שמתייחס למה שמופיע על התמונה (לדוגמא צורות גאומטריות) [2].

אחד החסרונות בשימוש באלגוריתמים כגון K-Means הוא שעלינו לדעת מראש מהו מספר האשכולות שנרצה בכדי להגיע לפתרון אופטימלי, על מריץ האלגוריתם להכיר את טיב התמונה איתה הוא עובד, מה שלא תמיד מתאפשר [3]. לא רק כך, אלא שפילוח תמונה הוא נושא סובייקטיבי יחסית ואנשים שונים יראו את התמונה באופן שונה וייצרו פלט שונה עבור אלגוריתמי קיבוץ אשכולות (על ידי הזנת k שונה).

לפיכך, נעשים היום מחקרים רבים בשוק לצורך קביעה אוטומטית של k עבור כל תמונה, כגון חילוץ k באמצעות אלגוריתם חיזוי של בינה מלאכותית שמזהה לבד את סוג התמונה ומצמיד לה k נפוץ (כגון נופים, משחקי מחשב, תצלום ועוד) או שימוש בשונות של ערכי RGB בפיקסלים של התמונה. אלגוריתמים רבים כיום פונים לשימוש בייצוג HSV עבור כל פיקסל, ייצוג חלופי ל RGB, שבמקרים מסוים מניב חיזוי k טוב יותר (ובמקרים מסוימים חיזוי k גרוע יותר).

נראה השוואה קצרה של פלט האלגוריתם עבור ייצוג פיקסלים בRGB ועבור ייצוג פיקסלים ב HSV(ייצוג הפיקסלים באופן שונה נעשה רק בשביל קביעת הk שלנו באופן אוטומטי):

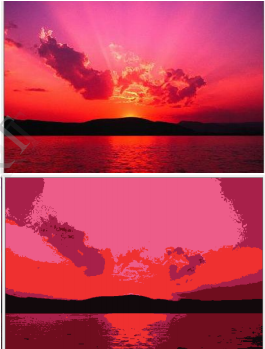


איור 13- ניתוח אותה התמונה על ידי ייצוג פיקסלים שונה, הייצוג העליון RGB והיצוג התחתון HSV.

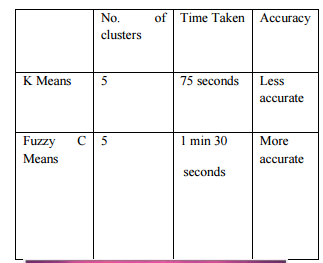
עתיד ניתוח התמונה מלא בתקוות גדולות כגון שיפור איכות תמונה עבור תמונות עם נקודות רעש, שיפור איכות תמונה בעולם הרפואי של MRIים, רנטגנים ועוד [5].

נתעמק קצת בקיבוץ אשכולות קשה לעומת קיבוץ אשכולות קל בניתוח תמונה, ונציג ניסוי שנערך במאמר Comparitive Analysis Of K Means And Fuzzy C Means Algorithm [5].

הריצו את האלגוריתמים K-Means ו Fuzzy C-Means עבור התמונה הבאה כאשר היה ידוע מספר האשכולות האופטימלי (k=5):

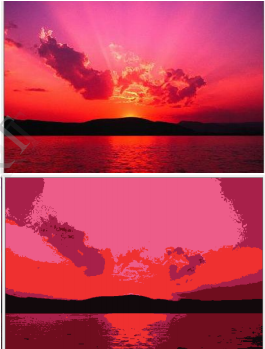


איור 14 - התמונה המקורית עליה הריצו את אלגוריתמי קיבוץ האשכולות, [5]

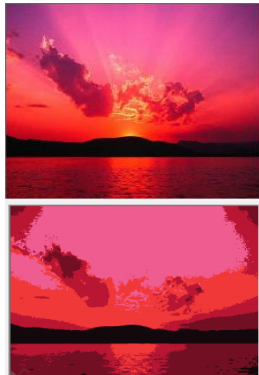


איור 15 - זמן הריצה שנלקח עבור כל אחד משני האלגוריתמים K-Means ו Fuzzy C-Means [5]

נסתכל על זמני הריצה ונראה שהם תואמים את המסקנה אליה הגענו בפרק הקודם. אלגוריתם K-Means הוא בעל זמן ריצה קצר יותר מאלגורתם Fuzzy C-Means, אולם הוא פחות מדויק מכיוון שהוא מחייב כל נקודה לאחד מחמשת האשכולות בדוגמא ולא מאפשר לכל גמישות.



איור 16 - תוצאות הרצת אלגוריתם K-Means על תמונה עבור k=5, [5]

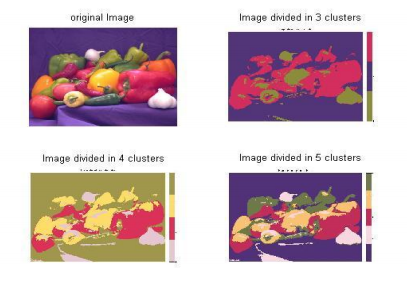


איור 17 - תוצאות הרצת אלגוריתם Fuzzy C-Means על תמונה עבור k=5, [5]

* 1. ניתוח תמונה בניתוח אשכולות קשה

במאמר "Performance based analysis between K-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data " [9] תואר אלגוריתם K-Means מורחב אשר נקרא Fast SQL K-Means הצליח לשפר משמעותית את זמן הריצה של ניתוח תמונות בגודל 400X250 (100 אלף פיקסלים) פי 90.

במאמר “A Comparative Analysis of Fuzzy C-Means Clustering and K Means Clustering Algorithms” [12] ראינו הדגמה של ניתוח תמונה עבור אלגוריתם K-Means, עבור 3 אפשרויות שונות של מספר אשכולות: עבור k=3, k=4 ו k=5. התוצאות מובאות באיור הבא:



איור 18 - הדגמת ניתוח תמונה באמצעות אלגוריתם K-Means עבור 3 אפשרויות שונות של k, [12].

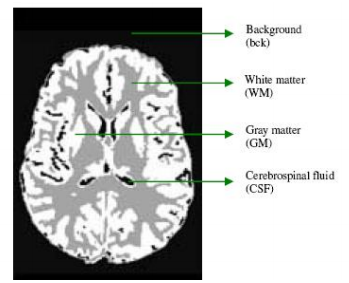
* 1. ניתוח תמונה בניתוח אשכולות רך

אף על פי שאלגוריתמים מסוג Fuzzy C-Means "רגישים" לנקודות רעש בקלט שלהם, ישנם הרחבות רבות של אלגוריתם זה שמצליחות להוציא תוצאות מוצלחות יותר עבור תמונות "רועשות" מאלגוריתם K-Means. במידה ומחקרים אלו ימשיכו להניב תוצאות פורות, ניתן יהיה להשתמש בניתוח תמונה גם עבור תמונות עדינות כגון תמונות CT ( סריקת מוח) או סריקות PET.

הזכרנו מעלה ניתוח תמונה עבור MRI אשר מכיל פחות נקודות רועשות ולרוב מורכב מצבעי שחור לבן ואפור ולכן משתמשים כבר היום באלגוריתמי ניתוח אשכולות קלים עבורו [1].

כאשר עובדים על ניתוח תמונת MRI, משתמשים בשיטה הנקראת Low depth of Field. שיטה זו מאפשרת להקנות רמת חשיבות מסוימת לחלק מחלקי התמונה בכדי שהם ייצאו מפוקסים יותר. בתמונת MR לדוגמא, ניתן יהיה לבקש להקנות חשיבות לחלקים הספציפיים בתמונת הMRI המעניינים (לדוגמא גירוי אזור מסוים במוח) [1].

שיטה נוספת בניתוח תמונות בדומה לתמונת MRI היא Self Organizing Map (SOM). בשטה זו התמונה מחולקת לשני חלקים. חלק ראשון הוא התמונה כפי שהיא במאגר הנתונים, והחלק השני מורכב מזיהוי מדויק בתמונה של רקמות ספציפיות שאלגוריתם ניתוח התמונה יודע לחפש [1].



איור 19 - פילוח תמונה עבור MRI [1].

במאמר “A Comparative Analysis of Fuzzy C-Means Clustering and K Means Clustering Algorithms” [12] ראינו הדגמה של ניתוח תמונה עבור אלגוריתם Fuzzy C-Means, עבור 3 אפשרויות שונות של מספר אשכולות: עבור k=3, k=4 ו k=7. התוצאות מובאות באיור הבא:



איור 20 - הדגמת ניתוח תמונה באמצעות אלגוריתם Fuzzy C-Means עבור 3 אפשרויות שונות של k, [12].

כפי שניתן לראות התוצאות מאוד מרשימות ומדמות בצורה חדה ואיכותית את התמונה המקורית. כזכור ניסוי דומה נערך גם עבור K-Means בתמונה דומה, ומהשוואת תוצאות שני הניסויים, ניתן לראות כי אלגוריתם Fuzzy C-Means מביא לתוצאות חדות וברורות יותר. לא בכדי בחרו בו בתחום הרפואה.

7. סיכום

בעבודה זו התעמקנו בקיבוץ אשכולות, אחת השיטות השמישות של כריית מידע. קיבוץ אשכולות מאפשר לנו לפנות למאגר של אובייקטים ולפרוס אותם על מישור ולפצל אותם לתתי קבוצות של אשכולות – כאשר אובייקטים תחת אותו אשכול דומים יותר זה לזה מאשר לאובייקטים מאשכול אחר.

ראינו שישנם שני סוגים עיקריים של קיבוץ אשכולות: קיבוץ אשכולות קל, וקיבוץ אשכולות קשה. קיבוץ אשכולות קשה הוא נוקשה יותר ומקשר כל אובייקט במישור רק לאשכול אחד בלבד. קיבוץ קל לעומת זאת מקשר כל אובייקט למספר אשכולות שונים ומציין מהי מידת השייכות של כל אובייקט לכל אשכול כזה.

בעבודה זו התמקדנו בשני אלגוריתמים מייצגים של קיבוץ אשכולות קל וקיבוץ אשכולות קשה, Fuzzy C-Means וK-Means. ראינו כיצד עובד כל אלגוריתם ובמה הוא תלוי, וכן מהו זמן הריצה של כל אלגוריתם.

כמו כן חקרנו מהם ההבדלים בין שני האלגוריתמים וגילינו כי אלגוריתם K-Means בעל ביצועים טובים יותר מבחינת זמן ריצה. כמו כן, הוא אינו מצריך מהמשתמש לחשב מידע נוסף טרם הרצתו, בניגוד לאלגוריתם Fuzzy C-Means.

לאחר מכן התמקדנו בנושא ניתוח תמונה, וראינו שם ממצאים מעניינים. בעוד ששני האלגוריתמים מאוד נפוצים בתחום, דווקא אלגוריתם Fuzzy C-Means מדויק יותר ומביא לדרגות אמינות גבוהות יותר, ולכן נבחר להשתמש בו בין היתר בניתוח תמונה בעולם הרפואה.

בהמשך המחקר העתידי הייתי שמחה לחקור את ההרחבות הקיימות של שני האלגוריתמים, ולראות כיצד הן פותרות את הבעיות שיש היום. כמו כן הייתי שמחה לראות כיצד כל שיפור והרחבה של Fuzzy C-Means מאפשרת לנו לנתח תמונות באופן אוטומטי מלא, מבלי הצורך להכיר את סוג התמונה והמאפיינים שלה לפני התחלת הניתוח שלה.

1. ביביליוגרפיה

[1] Mahesh Yambal and Hitesh Gupta. “Image Segmentation using Fuzzy C Means Clustering: A survey”. *International Journal of Advanced Research in Computer and Communication Engineering.* Vol. 2, Issue 7, (July 2013)

[2] Dipak Punekar and Dinesh Waditake, Nitish Pawar. “Analysis of Image Matching Towards the Hard Clustering”. *International Journal of Scientific and Research Publications*. Volume 4, Issue 4, (April 2014). ISSN.

[3] Siddheswar Ray and Rose H. Turi. “Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation”. School of Computer Science and Software Engineering, Monash University, Wellington Road, Clayton, Victoria, 3168, Australia.

[4] Soumi Ghosh and Sanjay Kumar Dubey. “Comparative Analysis of K-Means and Fuzzy CMeans

Algorithms”. *((IJACSA) International Journal of Advanced Computer Science and Applications*. Vol. 4, No.4, (2013).

[5] Poonam fauzdar and Sujata Kindri. “Comparitive Analysis Of K Means And Fuzzy C Means Algorithm”. *International Journal of Engineering Research & Technology (IJERT)*. Vol. 2 Issue 6, June, (2013)

[6] Maoguo Gong, Member, IEEE, Yan Liang, Jiao Shi, Wenping Ma, and Jingjing Ma. “Fuzzy C-Means Clustering With Local Information and Kernel Metric for Image Segmentation”. *IEEE Transactions on Image Proccessing*. Vol 22, No. 2, (February 2013).

[7] Mavroeidis Dimitrios and Marchiori Elena. “Feature selection for K-Means clustering stability: theoretical analysis and an algorithm”. *Data Mining and Knowledge Discovery*. Volume 28, Issue 4 , pp 918-960, (June 2014).

[8] Suganya, R and Shanthi, R. “Fuzzy C- Means Algorithm- A Review”. *International Journal of Scientific and Research Publications.* Volume 2, Issue 11, (November 2012).

[9] Velmurugan T. “Performance based analysis between K-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data”. *Applied Soft Computing*. Volume 9, Pages 134-146, (June 2014).

[10] Singh, T and Mahajan, M, “Performance Comparison of Fuzzy C Means with Respect to Other Clustering Algorithm”. *International Journal of Advanced Research in Computer Science and Software Engineering.* Volume 4, Issue 5, (May 2014)

[11] “Fuzzy C-Means clustering algorithm”, *Data Clustering Algorithms. https://sites.google.com/site/dataclusteringalgorithms/Fuzzy-C-Means-clustering-algorithm.*

[12] Bharati R, .Gohokar V. “A Comparative Analysis of Fuzzy C-Means Clustering and K Means Clustering Algorithms”. *International Journal Of Computational Engineerin.* Vol. 2, Issue No.3, Pages 737-739, (May-June 2012)

[13] “Cluster\_analysis", *Wikipedia*

https://en.wikipedia.org/wiki/Cluster\_analysis