Ariel Levin
ariel.lvn89@gmail.com

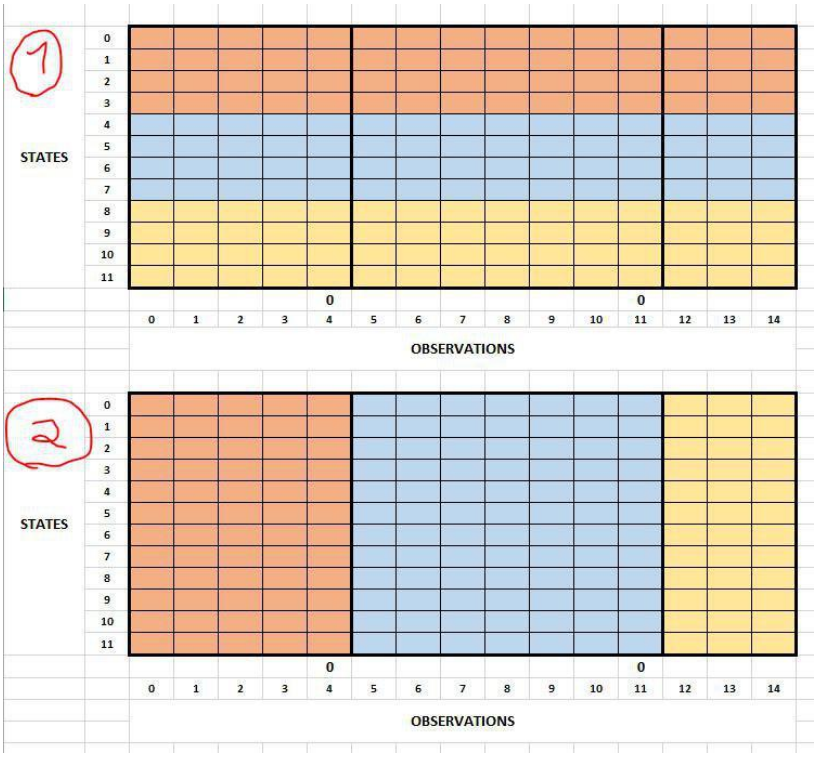
פרויקט מחשוב מקבילי ומבוזר
אלגוריתם Viterbi
תיעוד הפתרון

הפתרון שאיתו בחרתי לממש את אלגוריתם Viterbi כולל את שלושת טכנולוגיות המיקבול שבהן עסקנו במסגרת הקורס: MPI, OpenMP ו-CUDA.

להלן שתי הדרכים העיקריות לשימוש ה-MPI בפרויקט זה:

1. חלוקה של המטריצת הסתברויות לפי שורות, זאת אומרת שכל תהליך ב-MPI יהיה אחראי לחישוב חלק ספציפי וקבוע מהמצבים בעמודה הבאה.

2. חלוקה של המטריצת הסתברויות לפי אינדקסים שה-Observation בהם הוא 0, ושכל תהליך ב-MPI יטפל בחלק כזה של המטריצה.



תמונה זו היא דוגמא להמחשת הדרכים שציינתי למעלה, שבה כל צבע מביע תחום חישוב של כל תהליך MPI.

Ariel Levin
ariel.lvn89@gmail.com

אני בחרתי במימוש שלי להתמקד בדרך הראשונה.

הסיבה שבחרתי להתמקד בדרך זו, היא מכיוון שכך החלוקה בין התהליכים תמיד שווה, ללא תלות בכמות התהליכים שמפעילים, או בכמות האפסים שיהיו ב-Observations.

בדרך השנייה, יכול להיווצר מצב שלתהליך אחד יש הרבה עבודה לעומת תהליך אחר שיש לו קצת. אם יש לנו יותר תהליכים מאשר אפסים אז נוצר מצב שיש תהליכים שלא עושים כלום, ואם יש לנו יותר אפסים מאשר תהליכים אז צריך לחכות שתהליכים יתפנו לפני חישוב המשך המטריצה.

אופן המימוש:

1. אתחול:

- בתחילת התוכנית כל תהליך MPI מקצה לו את המקום הנדרש בזיכרון.
- המאסטר (Rank 0) מאתחל את הערכים למטריצות AB, ו-Transition, ומערך ה-Observation, שאמורות להיות נתונות לנו בשביל הרצת האלגוריתם. הוא עושה זאת או ע"י הרצת נתונים קבועים מראש בתוכנית, או ע"י קריאת נתונים מקובץ, או ע"י הגרלת מספרים אקראיים. בנוסף מאתחל את העמודה הראשונה במטריצת הסתברויות להיות 0 [1.1] בעזרת OpenMP.
- במידה ובחרנו להגריל מספרים אקראיים, המאסטר גם דואג לנרמל את מטריצת ה-Transition בעזרת OpenMP, מכיוון שזוהי מטריצה של הסתברויות, ועל כל סכום כל שורה או עמודה להיות שווה לאחד.
- המאסטר שולח ל-CUDA את מטריצת AB שהיא קבועה, בשביל שישמור קבוע אצלו בזיכרון בשביל חישוב ה-Emission בכל שלב.
- המאסטר מפעיל פונקציית Log [1] על כל ערכי מטריצת ה-Transition בעזרת OpenMP, ושולח את המטריצה לכל הסלייבים.

Ariel Levin
ariel.lvn89@gmail.com

2. אלגוריתם:

בשביל להקל את הסברת האלגוריתם, אתייחס למטריצת הסתברויות 'mat' כמטריצה של N שורות שהם כמות התהליכים, ו-K עמודות שהם כמות ה-Observations.

אופן פעולת האלגוריתם היא ריצה של המאסטר על כל מערך ה-Observations.

בכל איטרציה i כזו, הוא שולח את mat[i] (עמודת ההסתברויות ה-i) לכל הסלייבים ודגל שמורה להם מה לעשות באיטרציה זו.

במידה וזו איטרציה רגילה [2.1], המאסטר ישלח לסלייבים בנוסף מערך בגודל N של כל ה-Emissions שחושבו באותה איטרציה ע"י ה-CUDA.

לאחר שהסלייבים יתחילו את עבודתם, המאסטר (בזמן שמחכה להם), ישלח ל-CUDA את העמודה הבאה לחישוב ה-Emission שלה, ולאחר מכן יחכה לקבלת התוצאות מהסלייבים, שכנראה יסיימו באותו הזמן.

כאשר סלייב מקבל מהמאסטר את העמודה הנוכחית בעלת ה-N איברים, הוא יחשב את החלק שלו בעמודה הבאה, ע"י ביצוע אלגוריתם Viterbi.

הפונקציה שמבצעת את אלגוריתם Viterbi עושה זאת ע"י ריצה בלולאה חיצונית על החלק של אותו תהליך בעמודה הבאה, ולולאה פנימית שרצה על כל האיברים של עמודה הנוכחית ומוצאת מקסימום מבניהם.

על הלולאה החיצונית בפונקציית Viterbi אנו נרוץ עם OpenMP, מכיוון שמציאת כל איבר בעמודה הבאה אינו תלוי במציאת איבר אחר, בניגוד ללולאה הפנימית שמחפשת מקסימום ומגבילה אותנו מבחינת מגבול יעיל. ובכך בעצם אנחנו מייעלים ומקצרים את זמני החישוב של הסלייבים: גם כל סלייב מחשב רק את החלק שלו, וגם בחלק הזה מתחלקת העבודה ב-OpenMP.

כאשר הסלייב יסיים לחשב את החלק שלו בעמודה הבאה, הוא ישלח אותה חזרה למאסטר.

אם הסלייב מקבל דגל מהמאסטר שאומר שזו איטרצית סוף נתיב [2.2], אז הסלייב במקום לחשב את העמודה הבאה, הוא ימצא את ההסתברות המקסימלית בעמודה הנוכחית בחלק שלו, וישלח את האינדקס הזה למאסטר.

באיטרציה כזאת [2.2] המאסטר יקבל מהסלייבים את המקסימום בחלק שלהם, ימצא את המקסימום מבין כל המקסימומים שקיבל, ישמור אותו, ויאתחל את ההסתברויות של העמודה הבאה להיות 0 [1.1] בעזרת OpenMP.

המאסטר ירוץ עד סוף לולאת ה-Observations, והסלייבים יהיו בלולאה כל עוד המאסטר לא שלח להם דגל סיום.

Ariel Levin
ariel.lvn89@gmail.com

3. <u>הצגת תוצאות:</u>

בסוף הלולאה על כל ה-Observations, המאסטר יציג לנו את התוצאות, או בהדפסה, או לקבצים.
הוא יעשה זאת ע"י מעבר על כל המצבים המקסימליים שנשמרו במערך (ככמות האפסים במערך Observations + 1 של העמודה האחרונה), וימצא את הנתיב הגעה אליהם ע"י מעבר ממצב אל ההורה שלו וכן הלאה עד לתחילת הנתיב.
התוכנית תדפיס את מספריי המצבים המקסימליים, ההסתברויות שלהם (או הלוגים אם ההסתברויות מתאפסות), ואת הנתיב עד ההגעה אליהם.
במידה ונדפיס לקובץ אז כל הנתיב יוצג, במידה ונבחר להדפיס לקונסול, אם הנתיב ארוך מ-10 מצבים, אז נראה נתיב מצומצם (2 התחלתיים, 2 סופיים, ו-3 נקודות באמצע).
בנוסף התוכנית תציג כמה זמן לקח לביצוע האלגוריתם, זמן הנמדד גם ע"י MPI וגם ע"י OpenMP.

<u>הערכת סיבוכיות:</u>

סיבוכיות הריצה של תוכנית זו היא: $O(K * \frac{N}{mpi_{procs} * openmp_{procs}} * N)$, כאשר:

K – מספר ה-Observations
N – מספר ה-States
mpi_{procs} – מספר תהליכי ה-MPI
$openmp_{procs}$ – מספר תהליכי ה-OpenMP

חלוקה זו מתרחשת מכיוון שהחישוב של כל עמודת הסתברות הבאה מתחלק בין כל תהליכי ה-MPI, ובנוסף כל תהליך MPI מפעיל תהליכי OpenMP.

אך מכיוון שמס' התהליכים גם של ה-MPI וגם של ה-OpenMP הוא קבוע וקטן יחסית, ניתן לא להתחשב בו, ולכן הסיבוכיות הסופית היא:

$$\boxed{O(K * N^2)}$$

Ariel Levin
ariel.lvn89@gmail.com

הסברים נוספים:

[1] עבודה עם לוגים:

מכיוון שאנחנו מתעסקים כמות הכפלות גדולה של מספרים אשר קטנים מ-1, לאחר מספר איטרציות ההסתברויות פשוט יתאפסו, גם כאשר נעבוד עם משתני Double.
לכן במקום להכפיל את המספרים באלגוריתם Viterbi, האלגוריתם מבצע חיבור של לוגים, כך שבמטריצת ההסתברויות במקום שישמרו ההסתברויות, נשמר הלוגים שלהן. ואז בשביל להגיע מהלוג אל ההסתברות, פשוט נפעיל פונקציית exp() .
שיטה זו נבדקה ועובדת כמו שצריך.

[1.1] בנוסף, בשיטה זו נאתחל עמודה במטריצת הסתברויות עם 0 במקום 1 בגלל החישוב הפשוט $\log(1) = 0$.

[2] מצבי איטרציה:

[2.1] איטרציה רגילה – מצב שבו ה-Observation שונה מ-0 ואנחנו לא בעמודה האחרונה. במצב זה הסלייבים יחשבו בעזרת אלגוריתם Viterbi את החלק שלהם בעמודה הבאה (i+1) ויחזירו למאסטר.

[2.2] איטרצית סוף נתיב – מצב שבוא ה-Observation שווה ל-0 או שאנחנו בעמודה האחרונה. במצב זה הסלייבים ימצאו את המקסימום בחלק שלהם מהעמודה הנוכחית (i) ויחזירו למאסטר.