Karlsruher Institut für Technologie

Bachelorarbeit 2020 Code Dokumentation

Bachelorarbeit

Vorkonditionierung mit dem SPAI-Algorithmus

Tim Laufer

15. September 2020

Betreuung:
Dr. Markus Neher

Inhaltsverzeichnis

1 Aufruf des Algorithmus

Sind sowohl die Methode *load_ data*, als auch die zu testenden Matrizen im selben Ordner abgelegt, kann der Algorithmus über die Konsole gestartet werden. Abbildung 1 zeigt

```
>python SPAI_seriell.py 0.1 4 2 2 3
```

Abbildung 1: Programmaufruf mit Parametern

dabei den Aufruf, wo auch zu sehen ist, dass die Parameter nach dem Filenamen übergeben werden Es werden eine Gleitkommazahl, vier Integer und einen String übergeben. Diese beschreiben

1. die obere Schranke des Residuums
2. den maximalen *Fill-in* in den Spalten von M
3. die Art des Abbruchkriteriums
4. die Anzahl neuer Elemente für Expansion von J pro Iterationsschritt

5. die Art der Testmatrix
6. den Namen der *.npz*-Datei, in welcher die Daten der Matrix M gespeichert werden

2 Methoden

2.1 Start

- Übergabeparameter
  - Matrix A vom Typ *scipy.sparse.csc.csc_matrix*
  - obere Fehlerschranke *epsilon* vom Typ *float*
  - Art des Abbruchkriteriums vom Typ *int*
  - Anzahl neuer Indizes pro Iterationsschritt vom Typ *int*
  - Default-Matrix M vom Typ *scipy.sparse.csc.csc_matrix*
  - Nullmatrix vom Typ *scipy.sparse.lil.lil_matrix*
- Rückgabeparameter
  - Anzahl Iterationen vom Typ *int*
  - Dauer eines Durchlaufs des Algorithmus vom Typ *float*
  - Matrix M vom Typ *scipy.sparse.csc.csc_matrix*

Je nach Abbruchkriterium werden die verschiedenen Hauptmethoden gerufen. Die Anzahl der benötigten Schritte und die benötigte Zeitdauer werden zurückgegeben.

2.2 Initialize_J

- Übergabeparameter
  - Default-Spalte m vom Typ *scipy.sparse.csc.csc_matrix*
- Rückgabeparameter
  - Liste J vom Typ *list*

Die Zeilenindizes der Nichtnullelemente der Spalte m werden über das Attribut *m.inidces* bestimmt, da m vom Typ *scipy.sparse.csc.csc_matrix* ist.

2.3 Initialize_I

- Übergabeparameter
  - Matrix A vom Typ *scipy.sparse.csc.csc_matrix*
  - Menge J vom Typ *list*
  - Index der aktuellen Spalte vom Typ *int*
- Rückgabeparameter
  - Liste I vom Typ *list*

Die Zeilenindizes der Nichtnullzeilen der Matrix $A_{:,J}$ werden über das Attribut $A_{:,J}.inidces$ bestimmt, da A ein *scipy.sparse.csc.csc_matrix* Objekt ist. Damit das Residuum kleiner 1 werden kann, muss der Spaltenindex der Spalte m in I enthalten sein. Ist dem noch noch nicht so, wird dieser hinzugefügt.

2.4 unit_vector_tilde

- Übergabeparameter
  - Index i vom Typ *int*
  - Menge I vom Typ *list*
- Rückgabeparameter
  - i-ter Einheitsvektor vom Typ *numpy.ndarray*

Dieses Methode bestimmt den i-ten Einheitsvektor, nachdem der Algorithmus die Nullzeilen aus $A_{:,J}$ entfernt hat. Dieser Vektor hat i.A. kleinere Dimension als der i-te Standardeinheitsvektor.

2.5 retransform_residual

- Übergabeparameter
  - Residuum r in reduzierter Form als *scipy.sparse.csc.csc_matrix*
  - Menge I vom Typ *list*
  - Matrixdimension n vom Typ *int*
- Rückgabeparameter
  - Residuumsvektor der Dimension n vom Typ *numpy.ndarray*

Diese Methode transformiert den aktuellen Residuumsvektor auf die Ausgangsdimension zurück.

2.6 unite_sets

- Übergabeparameter
  - Menge I vom Typ *list*
  - Menge J vom Typ *list*
  - Menge $\tilde{I}$ vom Typ *list*
  - Menge $\tilde{J}$ vom Typ *list*
- Rückgabeparameter
  - Menge I vom Typ *list*
  - Menge J vom Typ *list*

Die Mengen I und J werden geupdatet durch die Mengen I und J, die die neuen Kandidaten beinhalten.

2.7 set_I_tilde

- Übergabeparameter
  - Matrix A als *scipy.sparse.csc.csc_matrix*
  - Menge $\tilde{J}$ vom Typ *list*
  - Menge I vom Typ *list*
  - Menge J vom Typ *list*
- Rückgabeparameter
  - Menge $\tilde{J}$ vom Typ *list*

Die Kandidaten für eine Erweiterung der Menge I werden gesetzt. Dafür werden Zeilenindizes der Matrix $A_{.,J\cup\tilde{J}}$ über das CSC-Format und *.inidces* abgerufen. Da Duplikate auftreten können, werden diese mit *numpy.unique* entfernt. Damit I und $\tilde{I}$ disjunkt sind, wird noch *numpy.setdiff1d* gerufen.

2.8 set_m_to_M

- Übergabeparameter
  - Spalte m als *scipy.sparse.csc.csc_matrix*
  - Spaltenindex i der aktuellen Spalte vom Typ *int*
  - Matrix M als *scipy.sparse.lil.lil_matrix*
  - Menge J vom Typ *list*
- Rückgabeparameter
  - Matrix M als *scipy.sparse.lil.lil_matrix*

Die Matrix M wird um die Spalte m erweitert. Aus Effizienzgründen wird LIL verwendet.

2.9 R_backwards_solver

- Übergabeparameter
  - Index der aktuellen Spalte vom Typ *int*
  - Matrix Q als Numpy-Array
  - Matrix R als Numpy-Array
  - Länge der Menge J vom Typ *int*
  - Menge I vom Typ *list*
- Rückgabeparameter
  - Spalte m vom Typ *numpy.ndarray*

Mittels *Scipy.sparse.linalg.solve_triangular* wird das LGS mit der oberen Dreiecksmatrix R gelöst. Zuvor muss noch ein Teil einer Zeile von Q extrahiert werden.

2.10 compute_QR_decomposition

- Übergabeparameter
  - Menge I vom Typ *list*
  - Menge J vom Typ *list*
  - Matrix A als *scipy.sparse.csc.csc_matrix*
- Rückgabeparameter
  - Matrix Q als Numpy-Array
  - Matrix R als Numpy-Array

Mittels *numpy.linalg.qr* wird die große QR-Zerlegung von $A_{I,J}$ berechnet. Da die komplette Zerlegung benötigt wird, wird noch „complete“ verwendet.

2.11 compose_QR_decomposition

- Übergabeparameter
  - Menge I vom Typ *list*
  - Menge J vom Typ *list*
  - Menge $\tilde{I}$ vom Typ *list*
  - Menge $\tilde{J}$ vom Typ *list*
  - Matrix A vom Typ *scipy.sparse.csc.csc_matrix*
  - Matrix Q vom Typ *numpy.ndarray*
  - Matrix R vom Typ *numpy.ndarray*

- Rückgabeparameter
  - Matrix Q vom Typ *numpy.ndarray*
  - Matrix R vom Typ *numpy.ndarray*

Die QR-Zerlegung von $A_{I \cup \tilde{I}, J \cup \tilde{J}}$ wird zusammengesetzt aus der alten schon berechneten QR-Zerlegung und einer kleinen noch zu berechnenden QR-Zerlegung. Aus Performancegründen wird $Q^T A_{I,\tilde{J}}$ anfangs einmal berechnet und dann zwischengespeichert.

2.12 control_lists

- Übergabeparameter
  - Menge I vom Typ *list*
  - Menge J vom Typ *list*
- Rückgabeparameter
  - Boolean

Wenn beide Mengen I und J leer sind, wird „False "zurückgegeben und der Algorithmus terminiert.

2.13 set_J_tilde

- Übergabeparameter
  - Matrix A als *scipy.sparse.csc.csc_matrix*
  - Residuumsvektor der Dimension n vom Typ *numpy.ndarray*
  - Menge J vom Typ *list*
  - Python-Liste vom Typ *list* bestehend aus n Python-Listen vom Typ *list*, welche die Zeilenindizes der Nichtnullelemente von A speichern
- Rückgabeparameter
  - Kandidatenenge vom Typ *list*

Methode prüft den Residuumsvektor auf noch fehlerhafte Einträge. In den noch fehlerhaften Zeilen werden die Spaltenindizes der Nichtnullelemente gesammelt, da sich diese Indizes für eine Expansion von J lohnen. Diese Menge von Kandidaten wird zurückgegeben, nachdem mittels *numpy.unique* und *numpy.setdiff1d* Duplikate entfernt wurden.

2.14 array_of_candidates

- Übergabeparameter
  - Matrix A als *scipy.sparse.csc.csc_matrix*
  - Anzahl neue Elemente zu m pro Iteration
  - Residuumsvektor der Dimension n vom Typ *numpy.ndarray*
  - Menge J vom Typ *list*
  - Python-Liste vom Typ *list* bestehend aus n Python-Listen vom Typ *list*, welche die Zeilenindizes der Nichtnullelemente von A speichern
  - n-dim numpy-Array mit Spaltennormen
  - Menge I vom Typ *list*
  - gekürzter Residuumsvektor vom Typ *numpy.ndarray*
- Rückgabeparameter
  - Array aus Index-Werte-Paaren vom Typ *numpy.ndarray*
  - Anzahl an Kandidaten vom Typ *int*

Es wird ein Array erstellt, das sowohl den Kandidaten, als auch dessen Wert im Sinne des eindimensionalen Minimierungsproblems speichert.

2.15 new_Indizes_J_Minimization_problem

- Übergabeparameter
  - Matrix A als *scipy.sparse.csc.csc_matrix*
  - Integer Anzahl neue Elemente zu m pro Iteration
  - Residuumsvektor der Dimension n vom Typ *numpy.ndarray*
  - Menge J vom Typ *list*
  - Python-Liste vom Typ *list* bestehend aus n Python-Listen vom Typ *list*, welche die Zeilenindizes der Nichtnullelemente von A speichern
  - n-dim numpy-Array mit Spaltennormen vom Typ *numpy.ndarray*
  - Menge I vom Typ *list*
  - gekürzter Residuumsvektor vom Typ *numpy.ndarray*
- Rückgabeparameter
  - Menge $\tilde{J}$ vom Typ *list*

Zuerst wird die Methode *set_J_tilde* aufgerufen, wo die Obermenge der Kandidaten gesetzt wird. Diesen Kandidaten bekommen anschließend durch die Methode *array_of_candidates* einen Wert zugewiesen. Anschließend wird das Array mit den Kandidaten-Werte-Paaren nach den Werten sortiert. Vom kleinesten zum größten Wert, da man am kleinsten Residuum interessiert ist. Ist das Array länger, als neue Indizes zu J hinzugefügt werden, werden die besten ausgewählt. Sind weniger Kandidaten vorhanden, als zu J hinzugefügt werden dürften, werden alle ausgewählt und als $\tilde{J}$ zurückgegeben.

2.16 getcol_boost

- Übergabeparameter
  - Matrix A als *scipy.sparse.csc.csc_matrix*
- Rückgabeparameter
  - Array der Dimension n mit Spaltennormquadraten vom Typ *numpy.ndarray*

Um die Normquadrate der einzelnen Spalten zu berechnen, wird A mit A^T multipliziert. Auf der Diagonale stehen dann die gewünschten Werte.

2.17 positions_nonzeros

- Übergabeparameter
  - Matrix A als *scipy.sparse.csc.csc_matrix*
  - Dimension n vom Typ *int*
- Rückgabeparameter
  - Python-Liste vom Typ *list* bestehend aus n Python-Listen vom Typ *list*, welche die Zeilenindizes der Nichtnullelemente von A speichern

Die Positionen der Nichtnullelemente der einzelnen Zeilen werden zeilenweise in einer Pythonliste gespeichert. Die Listen selbst werden in einer einzigen Liste gespeichert. Da A keine Nullen mehr abspeichert sind alle Zeilenindizes der Nichtnullelemente in *A.indices* gespeichert. Die Positionen der ersten und letzten Nichtnulleinträge der i-ten Zeile in *A.indices* bekommt man über $A.indptr[i]$ bzw. $A.indptr[i+1]$. Die Liste aus Listen wird zurückgegeben.

2.18 save_data

- Übergabeparameter
  - Matrix M als *scipy.sparse.csc.csc_matrix*
  - String *file* mit dem gewünschten Namen zur Speicherung von M

Mittels *numpy.savez_ compressed* werden die drei Vektoren *data*, *indices* und *indptr* vom Typ *numpy.ndarray* unter den Bezeichnungen *data*, *indices* und *indptr* unter *file.npz* gespeichert.