Humboldt-Universität zu Berlin
Institut für Mathematik
Prof. A. Griewank Ph.D.; Dipl.Math. T.Bosse

Übungsaufgaben zur Vorlesung
COMA (S 14)
Serie 5

Abgabe bis 26.05.2014

Aufgabe 5.1: (12 Punkte)

Schreiben Sie eine Klasse 'SparseMatrix' für die effiziente Speicherung von dünnbesetzten Matrizen. Anstatt der ganzen Matrix $A \in \mathbb{R}^{m \times n}$ sollen dabei nur die von Null verschiedenen Einträge abgespeichert und bei der Matrix-Vektor Multiplikation berücksichtigt werden. Die Informationen, d.h. die Dimensionen $m, n \in \mathbb{N}$ und die Anzahl $r \in \mathbb{N}$ der nicht-null Einträge, welche durch ihre Spalten- bzw. Zeilen-Indizes $s \in \{0 \ldots, n-1\}^r$, $z \in \{0 \ldots, m-1\}^r$ und die entsprechenden reelen Werte $w \in \mathbb{R}^r$ im "Sparse Triplet Format" gespeichert sind, sollen hierbei private Variablen der Klasse und nur über entsprechende Methoden abrufbar sein. Insbesondere soll die Klasse neben dem Konstruktor

- `public SparseMatrix (int m, int n, int r, int[] s, int[] z, double[] w)`

auch die folgende Methoden beinhalten:

- `public int getm():`
  Rückgabe der Dimension $m \in \mathbb{N}$
- `public int getn():`
  Rückgabe der Dimension $n \in \mathbb{N}$
- `public int getr():`
  Rückgabe der Anzahl $r \in \mathbb{N}$ von nicht-Null Einträgen
- `public int[] gets():`
  Rückgabe der Spaltenindizes $s \in \{0, \ldots, n-1\}^r$ in einem ganzzahligen Array
- `public int[] getz():`
  Rückgabe der Zeilenindizes $z \in \{0, \ldots, m-1\}^r$ in einem ganzzahligen Array
- `public double[] getw():`
  Rückgabe der von Null verschiedenen Werte $w \in \mathbb{R}^r$
- `public void print():`
  Geeignete Ausgabe von $s \in \{0, \ldots, n-1\}^r$, $z \in \{0, \ldots, m-1\}^r$ und $w \in \mathbb{R}^r$

sowie die beiden Methoden

- `public double[] sparsematvec(int n, double[] v):`
  Multiplikation Av von einer Sparse-Matrix $A \in \mathbb{R}^{m \times n}$ und einem Vektor $v \in \mathbb{R}^n$.
- `public double[] vectranssparsemat(int m, double[] u):`
  Multiplikation $u^\top A$ von einem Vektor $u \in \mathbb{R}^m$ und einer Sparse-Matrix $A \in \mathbb{R}^{m \times n}$.

zur effizienten Auswertung von Matrix-Vektor Operationen. Letztere sollen hierbei robust implementiert sein und mögliche Fehler mit Kontrollstrukturen abfangen, z.B. inkonsistente Dimensionen. Testen Sie die letzten beiden Methoden, in dem Sie die Gleichheit (bis auf Gleitkomma-Genauigkeit)

$$u^\top (Av) = (u^\top A)v$$

für zufällige Werte $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ und $v \in \mathbb{R}^n$ überprüfen.

Aufgabe 5.2: (8 Punkte)

Wenden Sie die 'SparseMatrix'-Klasse innerhalb der `powermethod1` aus Serie 3 an, um eine effiziente Eigenwertberechnung für dünnbesetzte Matrizen zu erhalten. Benutzen Sie diese modifizierte Methode zur Berechnung eines Fussballrankings, basierend auf den Ergebnissen der ersten 12. Spieltage[1]. Wählen Sie als Wichtungsmatrizen $A \in \mathbb{R}^{n \times n}$ des Eigenwertproblems

$$Ax = \lambda x$$

die beiden Optionen:

$$a_{ij} = \text{ Anzahl der Tore, die Team i gegen Team j geschossen hat } \quad \text{bzw.}$$

$$a_{ij} = \begin{cases} 0 & \text{Team i hat gegen Team j verloren} \\ 1 & \text{Team i hat gegen Team j unentschieden gespielt} \\ 3 & \text{Team i hat gegen Team j gewonnen} \end{cases}$$

Berechnen Sie die Eigenvektoren zu den (betragsmässig) grössten Eigenwerten für beide Wichtungen. Vergleichen und interpretieren Sie die Resultate mit den offiziellen Ergebnissen, d.h. mit dem Tabellenstand nach dem 12. Spieltag, dem 17. Spieltag (Herbstmeisterschaft) und dem 34. Spieltag (Meisterschaft) der Bundesliga Saison 2013-2014. Hierbei bietet sich eine tabellarisch Darstellung an.

[1] Siehe gaggle-Homepage und Übung für das Einlesen und Bearbeiten von Daten