

Prüfungsvorleistung Codierungs-Theorie

David Tarnow, Michael Narkus, Armel Siewel, Lukas Leichsenring

20. Juni 2018

Inhaltsverzeichnis

1	Einführung	1
1.1	Verwendung von C++	1
1.2	Armadillo	1
1.3	Gurobi	1
1.4	LAPACK	1
2	Schritt 1	1
2.1	Generierung von linear unabhängigen Vektoren	1
2.2	Erstellen der Matrix	2
2.3	Tests	2
3	Schritt 2	3
4	Schritt 3	3
5	Schritt 4	3
6	Schritt 5	3

1 Einführung

Einführung in die Thematik. Aufbau der Applikation, Imports (Gurobi, armadillo und co)

1.1 Verwendung von C++

Als Programmiersprache wurde in diesem Projekt C++ verwendet. Die entscheidenden Gründe für C++ waren zum einen die hohe Performance sowie die existierenden Bibliotheken (z.B. Armadillo) und Schnittstellen für die Nutzung von Gurobi.

1.2 Armadillo

Armadillo ist eine C++-Bibliothek, welche Funktionen und Klassen zur linearen Algebra bereitstellt. Angelehnt an MatLab werden hier Klassen für Vektoren, Matrizen und Kuben bereitgestellt. Im Rahmen dieses Projektes wurde Armadillo in der neusten stabilen Version (8.500.1) verwendet.

Die Einbindung erfolgte hierbei ...

1.3 Gurobi

1.4 LAPACK

2 Schritt 1

Ziel von Schritt 1 war es, anhand der Werte für k und q, eine Generatormatrix zu konstruieren.

Die Anwendung nimmt zunächst die Werte entgegen und berechnet eine Sammlung von linear unabhängigen Vektoren.

Sind alle linear voneinander unabhängigen Vektoren gefunden, wird anhand der Werte von k und q die Größe der Matrix berechnet. Die Berechnung basiert auf der, in der Vorlesung vorgestellten, Formel: $\frac{q^k - 1}{q - 1}$.

Anhand der berechneten Größe der Matrix, wird zum Abschluss eine Generatormatrix aus den zuvor erstellten Vektoren generiert und ausgegeben.

2.1 Generierung von linear unabhängigen Vektoren

Listing 1: Generierung der Vektoren

```
1 for (auto k_index = k - 1; k_index >= 0; --k_index)
{
    vec init_vec(k); // set size to k
    init_vec.fill(0); // init with zero
    init_vec.at(k_index) = 1;
6  ret.push_back(init_vec);
    calc_lin_independ_vec(ret, &init_vec, 1, q, k_index, k);
}
```

Listing 2: Berechnung der linearen Unabhängigkeiten

```
k_index++;
2 auto construct_vec = *init_vec;
  calc_lin_independ_vec(ret, &construct_vec, q_index, q,
    k_index, k);

  while (construct_vec.at(k_index) < (q - 1))
  {
7   construct_vec.at(k_index) += 1;
    calc_lin_independ_vec(ret, &construct_vec, q_index + 1, q,
      k_index, k);
  }
```

2.2 Erstellen der Matrix

Listing 3: Erstellen der Generatormatrix

```
1 arma::mat A(mat_size, mat_size);
  for (auto i = 0; i < mat_size; ++i)
  {
    for (auto j = 0; j < mat_size; ++j)
    {
6     const auto res = static_cast<int>(dot(lin_independ_vecs.
      at(i), lin_independ_vecs.at(j))) % q;
      A(i, j) = 0 == res ? 1 : 0;
    }
  }
```

2.3 Tests

Getestet wurde dieser Teil der Anwendung u.a. für die folgenden Wertepaare:

- bli
- bla
- blubb

3 Schritt 2

4 Schritt 3

5 Schritt 4

6 Schritt 5

Listings

1	Generierung der Vektoren	1
2	Berechnung der linearen Unabhängigkeiten	2
3	Erstellen der Generatormatrix	2