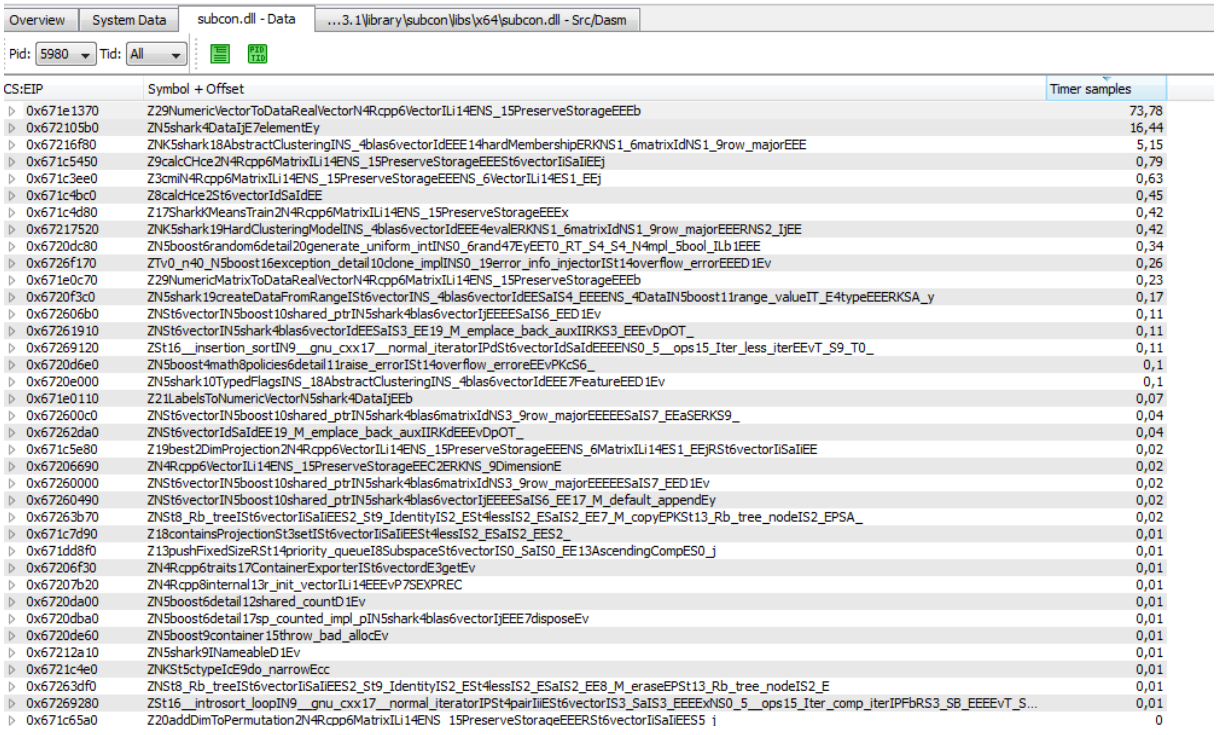
Laufzeit-Optimierung Shark-Methode

Inhalt

1. Analyse/Infos - Lange Laufzeit bei Methode “NumericVectorToDataRealVector”

- Performance-Analyse mit dem AMD CodeAnalyst ergab eine lange Laufzeit bei der Methode „NumericVectorToDataRealVector“:



Overview | System Data | subcon.dll - Data | ...3.1\library\subcon\libs\x64\subcon.dll - Src/Dasm

Pid: 5980 Tid: All

CS:EIP	Symbol + Offset	Timer samples
0x671e1370	Z29NumericVectorToDataRealVectorN4Rcpp6VectorILi14ENS_15PreserveStorageEEEb	73,78
0x672105b0	ZN5shark4DataIjE7elementEy	16,44
0x67216f80	ZNK5shark18AbstractClusteringINS_4blas6vectorIdEEE14hardMembershipERKNS1_6matrixIdNS1_9row_majorEEE	5,15
0x671c5450	Z9calcCHce2N4Rcpp6MatrixILi14ENS_15PreserveStorageEEESt6vectorIiSaIiEEj	0,79
0x671c3ee0	Z3cmiN4Rcpp6MatrixILi14ENS_15PreserveStorageEEENS_6VectorILi14ES1_EEj	0,63
0x671c4bc0	Z8calcHce2St6vectorIdSaIdEE	0,45
0x671c4d80	Z17SharkKMeansTrain2N4Rcpp6MatrixILi14ENS_15PreserveStorageEEEx	0,42
0x67217520	ZNK5shark19HardClusteringModelINS_4blas6vectorIdEEE4evalERKNS1_6matrixIdNS1_9row_majorEEERNS2_IjEE	0,42
0x6720dc80	ZN5boost6random6detail20generate_uniform_intINS0_6rand47EyEET0_RT_S4_S4_N4mpl_5bool_ILb1EEE	0,34
0x6726f170	ZTv0_n40_N5boost16exception_detail10clone_implINS0_19error_info_injectorISt14overflow_errorEEED1Ev	0,26
0x671e0c70	Z29NumericMatrixToDataRealVectorN4Rcpp6MatrixILi14ENS_15PreserveStorageEEEb	0,23
0x6720f3c0	ZN5shark19createDataFromRangeISt6vectorINS_4blas6vectorIdEESaIS4_EEEENS_4DataIN5boost11range_valueIT_E4typeEEERKSA_y	0,17
0x672606b0	ZNSt6vectorIN5boost10shared_ptrIN5shark4blas6vectorIjEEEESaIS6_EED1Ev	0,11
0x67261910	ZNSt6vectorIN5shark4blas6vectorIdEESaIS3_EE19_M_emplace_back_auxIIRKS3_EEEvDpOT_	0,11
0x67269120	ZSt16__insertion_sortIN9__gnu_cxx17__normal_iteratorIPdSt6vectorIdSaIdEEEENS0_5__ops15_Iter_less_iterEEvT_S9_T0_	0,11
0x6720d6e0	ZN5boost4math8policies6detail11raise_errorISt14overflow_erroreEEvPKcS6_	0,1
0x6720e000	ZN5shark10TypedFlagsINS_18AbstractClusteringINS_4blas6vectorIdEEE7FeatureEED1Ev	0,1
0x671e0110	Z21LabelsToNumericVectorN5shark4DataIjEEb	0,07
0x672600c0	ZNSt6vectorIN5boost10shared_ptrIN5shark4blas6matrixIdNS3_9row_majorEEEESaIS7_EEaSERKS9_	0,04
0x67262da0	ZNSt6vectorIdSaIdEE19_M_emplace_back_auxIIRKdEEEvDpOT_	0,04
0x671c5e80	Z19best2DimProjection2N4Rcpp6VectorILi14ENS_15PreserveStorageEEENS_6MatrixILi14ES1_EEjRSt6vectorIiSaIiEE	0,02
0x67206690	ZN4Rcpp6VectorILi14ENS_15PreserveStorageEEC2ERKNS_9DimensionE	0,02
0x67260000	ZNSt6vectorIN5boost10shared_ptrIN5shark4blas6matrixIdNS3_9row_majorEEEEESaIS7_EED1Ev	0,02
0x67260490	ZNSt6vectorIN5boost10shared_ptrIN5shark4blas6vectorIjEEEESaIS6_EE17_M_default_appendEy	0,02
0x67263b70	ZNSt8_Rb_treeISt6vectorIiSaIiEES2_St9_IdentityIS2_ESt4lessIS2_ESaIS2_EE7_M_copyEPKSt13_Rb_tree_nodeIS2_EPSA_	0,02
0x671c7d90	Z18containsProjectionSt3setISt6vectorIiSaIiEESt4lessIS2_ESaIS2_EES2_	0,01
0x671dd8f0	Z13pushFixedSizeRSt14priority_queueI8SubspaceSt6vectorIS0_SaIS0_EE13AscendingCompES0_j	0,01
0x67206f30	ZN4Rcpp6traits17ContainerExporterISt6vectordE3getEv	0,01
0x67207b20	ZN4Rcpp8internal13r_init_vectorILi14EEEvP7SEXPREC	0,01
0x6720da00	ZN5boost6detail12shared_countD1Ev	0,01
0x6720dba0	ZN5boost6detail17sp_counted_impl_pIN5shark4blas6vectorIjEEE7disposeEv	0,01
0x6720de60	ZN5boost9container15throw_bad_allocEv	0,01
0x67212a10	ZN5shark9INameableD1Ev	0,01
0x6721c4e0	ZNKSt5ctypeIcE9do_narrowEcc	0,01
0x67263df0	ZNSt8_Rb_treeISt6vectorIiSaIiEES2_St9_IdentityIS2_ESt4lessIS2_ESaIS2_EE8_M_eraseEPSt13_Rb_tree_nodeIS2_E	0,01
0x67269280	ZSt16__introsort_loopIN9__gnu_cxx17__normal_iteratorIPSt4pairIiiESt6vectorIS3_SaIS3_EEEExNS0_5__ops15_Iter_comp_iterIPFbRS3_SB_EEEEvT_S...	0,01
0x671c65a0	Z20addDimToPermutation2N4Rcpp6MatrixILi14ENS_15PreserveStorageEEERSt6vectorIiSaIiEES5_j	0

- Implementierung befindet sich in der Datei utils.cpp

Infos zum Shark-Package

http://image.diku.dk/shark/doxygen_pages/html/_base_8h.html

http://image.diku.dk/shark/doxygen_pages/html/classshark_1_1_data.html

http://image.diku.dk/shark/sphinx_pages/build/html/rest_sources/tutorials/algorithms/kmeans.html?highlight=realvector

KMeans.h liegt unter C:\R\R-3.1.1\library\RcppShark\include\shark\Algorithms

Auszug Callstack:

Datei	Methode	ruft auf:
CMI.cpp	calcCHce	SharkKMeansTrain2
CMI.cpp	SharkKMeansTrain2	NumericMatrixToUnlabeledData
CMI.cpp	SharkKMeansTrain2	kMeans
CMI.cpp	SharkKMeansTrain2	LabelsToNumericVector
utils.cpp	NumericMatrixToUnlabeledData	NumericMatrixToDataRealVector
utils.cpp	NumericMatrixToDataRealVector	std::copy-function
utils.cpp	NumericMatrixToDataRealVector	createDataFromRange

2. Alternativen

2.1. std::vector<double> statt Data<RealVector>

NumericMatrixToDataRealVector2

Direktes Befüllen des vectors mit Inhalten:

```
std::vector<double> output(size);
  for(int i=0; i<X.rows(); i++){
    for(int j=0; j<X.cols(); j++){
      output[(i*X.cols())+j] = X(i,j);
    }
  }
```

➔ Problem: kmeans-Meth. erwartet die Daten als Datentyp UnlabeledData<RealVector>

2.2. RealVector direkt aufbauen statt zuerst eine Copy anzufertigen

NumericMatrixToDataRealVector3

```
RealVector tRV(X.cols());
  for(int c=0; c<X.cols(); c++){
    tRV.push_back(X( e, c));
  }
  outputStd.push_back(tRV);
```

Ergebnis: 717 sek. (= fast 12 Min.)

```
> numClusterION <- 10
> topkSearchION <- 500
> topkOutputION <- 100
>
> ####### CMI #######
> startCmiION <- Sys.time()
> CMIResultION = CMISearch(m, numClusterION, topkSearchION, topkOutp
utION)
number of 2-dim candidates: 496
number of 3-dim candidates: 4960
number of 4-dim candidates: 817
number of 5-dim candidates: 290
number of 6-dim candidates: 83
number of 7-dim candidates: 10
> endCmiION <- Sys.time()
>
> difftime(endCmiION, startCmiION, unit="sec")
Time difference of 717.7345 secs
```

Vgl.test Originalimplementierung: 559 sec. (=9,3 Min.)

➔ Keine Performance-Verbesserung

2.3. Typecast von std::vector zu RealVector

NumericMatrixToDataRealVector4

```
outputStd[(i*X.cols())+j] = (RealVector)X(i,j);
```

➔ Problem: Programm kompiliert zwar, aber zur Laufzeit hängt es sich auf
➔ Offen: weiter ausprobieren

2.4. ALGLIB (C++-Library) einbinden statt Shark

➔ Offen: Man könnte noch versuchen die ALGLIB (C++-Library) einzubinden:

http://www.alglib.net/download.php

- besitzt auch eine kmeans-Implementierung:

clustering subpackage

Classes

ahcreport
clusterizerstate
kmeansreport

Functions

clusterizercreate
clusterizergetdistances
clusterizergetkclusters
clusterizerrunahc
clusterizerrunkmeans
clusterizerseparatedbycorr
clusterizerseparatedbydist
clusterizersetahcalgo
clusterizersetdistances
clusterizersetkmeansinit
clusterizersetkmeanslimits
clusterizersetpoints