package com.twitter.simclusters\_v2.scalding.embedding.abuse

import com.twitter.scalding.typed.TypedPipe

import com.twitter.scalding.Args

import com.twitter.scalding.DateRange

import com.twitter.scalding.Execution

import com.twitter.scalding.UniqueID

import com.twitter.scalding.Years

import com.twitter.simclusters\_v2.scalding.common.matrix.SparseMatrix

import com.twitter.simclusters\_v2.scalding.embedding.abuse.DataSources.NumBlocksP95

import com.twitter.simclusters\_v2.scalding.embedding.abuse.DataSources.getFlockBlocksSparseMatrix

import com.twitter.simclusters\_v2.scalding.embedding.abuse.DataSources.getUserInterestedInTruncatedKMatrix

import com.twitter.scalding\_internal.dalv2.DALWrite.D

import com.twitter.scalding\_internal.dalv2.DALWrite.\_

import com.twitter.simclusters\_v2.scalding.embedding.common.EmbeddingUtil.ClusterId

import com.twitter.simclusters\_v2.scalding.embedding.common.EmbeddingUtil.UserId

import com.twitter.simclusters\_v2.scalding.embedding.common.EmbeddingUtil

import com.twitter.simclusters\_v2.scalding.embedding.common.ExternalDataSources

import com.twitter.simclusters\_v2.thriftscala.AdhocCrossSimClusterInteractionScores

import com.twitter.simclusters\_v2.thriftscala.ClustersScore

import com.twitter.simclusters\_v2.thriftscala.ModelVersion

import com.twitter.wtf.scalding.jobs.common.AdhocExecutionApp

import com.twitter.wtf.scalding.jobs.common.CassowaryJob

import com.twitter.simclusters\_v2.hdfs\_sources.AdhocCrossSimclusterBlockInteractionFeaturesScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.AdhocCrossSimclusterFavInteractionFeaturesScalaDataset

import java.util.TimeZone

/\*

To run:

scalding remote run \

--user cassowary \

--submitter hadoopnest1.atla.twitter.com \

--target src/scala/com/twitter/simclusters\_v2/scalding/embedding/abuse:cross\_simcluster-adhoc \

--main-class com.twitter.simclusters\_v2.scalding.embedding.abuse.CrossSimClusterFeaturesScaldingJob \

--submitter-memory 128192.megabyte --hadoop-properties "mapreduce.map.memory.mb=8192 mapreduce.map.java.opts='-Xmx7618M' mapreduce.reduce.memory.mb=8192 mapreduce.reduce.java.opts='-Xmx7618M'" \

-- \

--date 2021-02-07 \

--dalEnvironment Prod

\*/

object CrossSimClusterFeaturesUtil {

/\*\*

\* To generate the interaction score for 2 simclusters c1 and c2 for all cluster combinations (I):

\* a) Get C - user interestedIn matrix, User \* Cluster

\* b) Get INT - positive or negative interaction matrix, User \* User

\* c) Compute C^T\*INT

\* d) Finally, return C^T\*INT\*C

\*/

def getCrossClusterScores(

userClusterMatrix: SparseMatrix[UserId, ClusterId, Double],

userInteractionMatrix: SparseMatrix[UserId, UserId, Double]

): SparseMatrix[ClusterId, ClusterId, Double] = {

// intermediate = C^T\*INT

val intermediateResult = userClusterMatrix.transpose.multiplySparseMatrix(userInteractionMatrix)

// return intermediate\*C

intermediateResult.multiplySparseMatrix(userClusterMatrix)

}

}

object CrossSimClusterFeaturesScaldingJob extends AdhocExecutionApp with CassowaryJob {

override def jobName: String = "AdhocAbuseCrossSimClusterFeaturesScaldingJob"

private val outputPathBlocksThrift: String = EmbeddingUtil.getHdfsPath(

isAdhoc = false,

isManhattanKeyVal = false,

modelVersion = ModelVersion.Model20m145kUpdated,

pathSuffix = "abuse\_cross\_simcluster\_block\_features"

)

private val outputPathFavThrift: String = EmbeddingUtil.getHdfsPath(

isAdhoc = false,

isManhattanKeyVal = false,

modelVersion = ModelVersion.Model20m145kUpdated,

pathSuffix = "abuse\_cross\_simcluster\_fav\_features"

)

private val HalfLifeInDaysForFavScore = 100

// Adhoc jobs which use all user interestedIn simclusters (default=50) was failing

// Hence truncating the number of clusters

private val MaxNumClustersPerUser = 20

import CrossSimClusterFeaturesUtil.\_

override def runOnDateRange(

args: Args

)(

implicit dateRange: DateRange,

timeZone: TimeZone,

uniqueID: UniqueID

): Execution[Unit] = {

val normalizedUserInterestedInMatrix: SparseMatrix[UserId, ClusterId, Double] =

getUserInterestedInTruncatedKMatrix(MaxNumClustersPerUser).rowL2Normalize

//the below code is to get cross simcluster features from flockblocks - negative user-user interactions.

val flockBlocksMatrix: SparseMatrix[UserId, UserId, Double] =

getFlockBlocksSparseMatrix(NumBlocksP95, dateRange.prepend(Years(1)))

val crossClusterBlockScores: SparseMatrix[ClusterId, ClusterId, Double] =

getCrossClusterScores(normalizedUserInterestedInMatrix, flockBlocksMatrix)

val blockScores: TypedPipe[AdhocCrossSimClusterInteractionScores] =

crossClusterBlockScores.rowAsKeys

.mapValues(List(\_)).sumByKey.toTypedPipe.map {

case (givingClusterId, receivingClustersWithScores) =>

AdhocCrossSimClusterInteractionScores(

clusterId = givingClusterId,

clusterScores = receivingClustersWithScores.map {

case (cluster, score) => ClustersScore(cluster, score)

})

}

// get cross simcluster features from fav graph - positive user-user interactions

val favGraphMatrix: SparseMatrix[UserId, UserId, Double] =

SparseMatrix.apply[UserId, UserId, Double](

ExternalDataSources.getFavEdges(HalfLifeInDaysForFavScore))

val crossClusterFavScores: SparseMatrix[ClusterId, ClusterId, Double] =

getCrossClusterScores(normalizedUserInterestedInMatrix, favGraphMatrix)

val favScores: TypedPipe[AdhocCrossSimClusterInteractionScores] =

crossClusterFavScores.rowAsKeys

.mapValues(List(\_)).sumByKey.toTypedPipe.map {

case (givingClusterId, receivingClustersWithScores) =>

AdhocCrossSimClusterInteractionScores(

clusterId = givingClusterId,

clusterScores = receivingClustersWithScores.map {

case (cluster, score) => ClustersScore(cluster, score)

})

}

// write both block and fav interaction matrices to hdfs in thrift format

Execution

.zip(

blockScores.writeDALSnapshotExecution(

AdhocCrossSimclusterBlockInteractionFeaturesScalaDataset,

D.Daily,

D.Suffix(outputPathBlocksThrift),

D.Parquet,

dateRange.`end`),

favScores.writeDALSnapshotExecution(

AdhocCrossSimclusterFavInteractionFeaturesScalaDataset,

D.Daily,

D.Suffix(outputPathFavThrift),

D.Parquet,

dateRange.`end`)

).unit

}

}