The GRAPPA method for accelerating annotations on Big consumer opinion datasets. Applications to sentiment modeling on COVID19 Lockdown Tweets and Amazon Reviews.

Abstract

The annotation process on Big consumer opinion datasets from Amazon Reviews or Twitter chatter can easily become a computational bottleneck. In our case annotations consist of associating sentiments and emotions to millions of opinion records from Amazon or Twitter. To tackle this problem, we suggest a method that accelerates computations by splitting the set of opinion records into chunks and applying annotation calculations in parallel on each chunk. We called this method “GRAPPA” (GeneRal Approach for Parallel Processing Annotations) it uses the R statistical systems “Map” and “Reduce” higher order functions. We compare accelerations that can be achieved with this method by using implicit or multicore parallelism (on one workstation) and cluster parallelism mobilizing the authors’ university's datacenter. The results are stunning especially with cluster parallelism where computations that lasted more than half a day, could be reduced to a few minutes and even less as calculations scale quasi linearly. We present two applications of the method one building a Barometer of sentiments and emotions expressed on Twitter during Covid19 lockdown restrictions and the other doing predictive sentiment analysis using Amazon Customer Reviews.

Introduction

Social media produce a lot of chatter, that thoroughly mined and analyzed can produce useful information, whose extraction can be automated and sometimes replace cumbersome surveys. The latter may be often less reliable as based on declarative, less spontaneous and sincere answers to predefined questions. Additionally, when using powerful and quick computation methods, such content extraction exercises can be used to build reliable opinion barometers. Mining social media content usually consists of topic modeling, sentiment analytics or social network analysis. Sentiment analysis often reduces to huge annotation exercises that associate specialized dictionaries to a given content. The paper describes the annotation processes and introduces the GRAPPA method that uses parallel computing techniques to achieve substantial acceleration. Some extensions of the method and two applications: one building a Barometer of sentiments and emotions expressed on Twitter during Covid19 lockdown restrictions and the other doing predictive sentiment analysis using Amazon Customer Reviews are also presented. Finally, alternative cluster parallelism methods like MapReduce and many-core parallelism using CUDA on GPUs are discussed.

The GRAPPA method

The annotation process consisted in extracting words or emoticons expressing sentiments and emotions from social media content and counting their frequency. For this purpose, we used several dictionaries and lexicons:

• The NRC (National Research Council Canada) emotion lexicon (Mohammad and Turney, 2013): based on the Putchnik (1980) scale of sentiments and emotions and integrated to R's suyzhet package (Jockers, 2015) served to score individual content on sentiments (positive or negative) and emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) based on word occurrence.
• The Linguistic Inquiry and Word Count (LIWC) system (Pennebaker & al., 2015) that counts the percentage of words that reflect different emotions, thinking styles, social concerns, and even parts of speech, has been applied using the quatenda dictionaries[1] R package (Benoit, 2018). It adds to our analysis three groups of variables (80 variables): those related to relatives (friend, family, humans), those related to physiology (food, body, sexuality, health) and finally those related to the spatiotemporal dimension.
• The Lexicoder Sentiment Dictionary (Young & Soroka, 2012) with its LSD French version (Duval & Pétry, 2016) has been also used with quanteda in order to count terms expressing positive and negative sentiments.
• The Emoji Sentiment Ranking (Kralj Novak, 2015) has been applied to associate sentiment scores to all emojis present in tweets.

This annotation process can easily encounter a computational bottleneck, when the collected content consists of several millions of records and can last many hours and even days.
The method we found to accelerate computations consists in splitting those records into chunks and applying the annotation calculations in parallel on each chunk. We called this method “GRAPPA” (GeneRal Approach for Parallel Processing Annotations). It uses R’s “Map” and “Reduce” higher order functions. First, we adopted implicit or multicore parallelism on one workstation with an 8 core CPU (Intel I9 microprocessor) and 16G memory (RAM) using R’s “parallel” package (R Core Team, 2020) with its “mcapply” function that allowed for a two to four-fold time reduction. As this was not enough we switched to cluster parallelism using the authors’ universities resources and R’s rslurm package (Marchand & Carroll, 2019) that submits R calculations to the cluster’s job manager SLURM (Simple Linux Utility for Resource Management). The result was stunning: computation time fell to less than ten minutes, when using 16 nodes with 2 cores each. This huge difference in computing time between multicore and cluster parallelism is due to the fact that in multicore parallelism the available cores share the same memory while in cluster parallelism each node has its own memory.
Our approach consists essentially of three phases: split, map and reduce that apply to both multicore and cluster parallelism. While the split phase uses the same R instructions for both cases the map and reduce phases use slightly different commands.
In the SPLIT phase the text column in the “big” input data frame (df) is split into a number of chunks (nchunks) that equals the available cores on one machine or the available nodes (machines) in the cluster. The chunks are kept in a list of split text vectors (svtexte) like this:

```
svtexte=split(as.character(df$text),fsplit[1:nchunks]) # splitting
```

Each chunk will be dispatched to a CPU core or to a node (machine) in the map phase.
The MAP phase applies the annotation function, for example “get_nrc_sentiment(...)” when the NRC dictionary is used, on each chunk in the split list using the mclapply command that executes the function simultaneously on the mobilized cores like this:

```
resnrc <- mclapply(1:ncores, function(i) get_nrc_sentiment(svtexte[[i]],
language="french"))
```

Similarly the slurm_apply function, which is a wrapper in the cluster environment for the mclapply

1 https://github.com/kbenoit/quanteda.dictionaries

function, applies the annotation function to each node in the cluster like this:

```
sjobnrc <- slurm_apply(function(i) get_nrc_sentiment(svtexte[[i]],
language="french"),data.frame(i=seq_along(svtexte)),add_objects =
c("get_nrc_sentiment","svtexte"),jobname = 'nrc_16', nodes = 16,cpus_per_node = 2,
submit = TRUE)
```

The REDUCE phase regroups the resulting annotations collected from each core into one "big" table using the rbind (row bind) function:

```
resnrc <- Reduce(rbind, resnrc)
```

Similarly the "get_slurm_out(..)" function is equivalent to "Reduce" and collects annotations resulting from the SLURM jobs that were running on several machines on the cluster:

```
resnrc <- get_slurm_out(sjobnrc, outtype = 'table')
```

Some extensions of the method

Besides the parallel annotation process presented here, the GRAPPA method can be extended to many classical Data Analysis problems and allow for BigData calculations. This extension is possible because many classical data analysis algorithms, as those used in Linear Regression, Factor Analysis or Discriminant Analysis, rely on special cases of matrix multiplication that allow chunk-wise calculations. These are, for example, the multiplication of the transposed matrix with itself **X'X** or the multiplication of a transposed matrix with a vector **X'y** where, in the case of linear regression, **X** contains the values of the independent variables and **y** the values of the dependent variable. These multiplications are usually the only BigData calculations in those linear algebra algorithms as the input matrices they imply are "thin and tall" meaning that they can have millions or billions of rows and only tens or hundreds of columns. The outputs are relatively small matrices as their size depends only on the number of columns and can be solved using normal computing power available on commercial workstations or laptop computers.
The GRAPPA method splits these matrices into chunks as it did in the previous section with the text vectors, then maps those chunkwise matrix multiplications by dispatching the calculations on the available machines and/or cores and finally sums the collected chunkwise results in the reduce phase by exploiting the fact that unlike multiplication the sum of matrices is associative and commutative.

Applications of the method

Building a sentiment barometer on COVID19 Lockdown Tweets

In order to build a Lockdown COVID19 sentiment barometer data from the publicly available Twitter Rest API has been collected. The daily gathering process continued since March 17 until May 11, when lockdown restrictions in France have been loosened. Keywords used were #ConfinementJourXx hashtags. The Xx indicator means the number of days since the beginning of the lockdown.
The purpose of the study was to better understand human reactions to a brutal anthropological shock: the clash of an advanced and mobile society with a natural contingency - a virus which circulates from mouth to mouth and travels in business class. We tried to record how people experience this shock day after day, by capturing social chatter and measuring feelings, emotions and concerns.
In building the lockdown barometer all four dictionaries and lexicons mentioned before have been used and a methodology has been introduced to test convergent validity. A detailed description of the approach can be found in Benavent (2020). All the code used to extract, preprocess, annotate and analyze tweets is available on a public github repository
(https://github.com/BenaventC/BarometreConfinement).

The collected dataset and the sentiment and emotion annotations datasets are made available at https://github.com/calciu/COVID19-LockdownFr and a complete description can be found in Balech & al. (2020).
The way the GRAPPA method was applied in the annotation process using all four dictionaries has already been described in previous section as well as the stunning acceleration of calculations that has been obtained when applying the method on the authors' university computer cluster.

Predictive sentiment analysis using Amazon Customer Reviews

The predictive sentiment analysis we present uses Amazon customer reviews (courtesy McAuley et al., 2015, file size 58.3G), a file which contains 82.68 million reviews after de-duplication (142.8 million comments originally), which runs from May 1996 to July 2014.
The GRAPPA method was used here to transform (annotate) the text of customer reviews into vectors of frequencies which account for the distribution of sentiments and emotions in the texts using the NRC dictionary. The proportions of emotions by text helped explain the assigned utility score derived from the review quality perceived and evaluated by other customers through their helpfulness ratings and to predict the importance of emotions in classifying reviews as having high or low utility. Random forests classifiers have been trained on the basis of positively and negatively helpful reviews in order to identify the importance of emotion and sentiment dimensions in customer reviews. More details on the predictive model can be found in Felbermayr & Nanopoulosto (2016).

Conclusions and discussions

More's law, which states the doubling of transistors on a chip every two years was the greatest technological prediction of the last half century. It defined the miniaturization process that was digging into the "computing Microcosm" until recently when apparently a physical limit has been attained. This limit forced computing technology to address the alternative growth vector by exploring the "computing Macrocosm" especially with multi-core, many-core and cluster parallelism.
The data scientist and the marketing analyst can no longer rely on a classical workstation for their BigData calculations. They must take advantage of opportunities the above-mentioned Macrocosm brings about and eventually go out over the Internet in order to use and/or exploit capabilities offered by data-centers who essentially represent huge computing infrastructures with lots of computers organized in clusters or grids.
Several technologies are currently available that exploit multiple levels of parallelism (e.g. multi-core, many-core, GPU, cluster, etc). Depending on the application demands they trade-off aspects such as performance, cost, failure management, data recovery, maintenance and usability. For example, a preference for speed at the expense of a low fault tolerance and vice versa (Reyes-Ortiz & al., 2015)
The GRAPPA method is a rather intuitive approach that can be extended to parallelize several kinds of data analytic problems. It splits the data into chunks and then uses R's higher level Map and Reduce functions in order to achieve multicore and cluster parallelism. The latter uses one of the most commonly used cluster computing frameworks for big data analytics OpenMP/MPI which efficiently exploits multi-core clusters architectures and combines the MPI paradigm with shared memory multiprocessing. MapReduce is an alternative share-nothing framework particularly adapted for cloud computing which comes with additional fault tolerance support and data replication. It significantly contributed to democratize cluster parallelism for BigData calculations by hiding all complexities of parallel computing and limiting the analyst's task to Map functions of key and value pairs and select the Reduce functions that aggregate the parallel computation results.
A further research direction is to explore other parallel programming models using many-core Graphics Processing Units (GPUs) which present an enormous and more affordable computation resource. With CUDA (Compute Unified Device Architecture) analysts can access to GPU memory and utilize parallel

computation not only for graphic applications but for general purpose processing. They can take the parallelism advantage presented by the multi-core CPUs and many-core GPUs (Mivule & al., 2014).

References

Balech, S., Benavent C., Calciu M. (2020) The First French COVID19 Lockdown Twitter Dataset, *arXiv e-prints*, May, https://arxiv.org/ftp/arxiv/papers/2005/2005.05075.pdf

Benavent C. (2020) Laboratoire du confinement, https://benaventc.github.io/BarometreConfinement/, last visited on 24/09/2020.

Benoit, K., Watanabe K., Wang H., Nulty P., Obeng A., Müller S., Matsuo A. (2018)
quanteda: An R package for the quantitative analysis of textual data, *Journal of Open Source Software*, 3, 30, doi:10.21105/joss.00774}, https://quanteda.io

Duval, D. and Pétry, F. (2016) "L'analyse automatisée du ton médiatique : construction et utilisation de la version française du Lexicoder Sentiment Dictionary", Revue canadienne de science politique, 49(2), pp. 197–220.

Jockers, M.L. (2015) Syuzhet: Extract Sentiment and Plot Arcs from Text,
https://github.com/mjockers/syuzhet

Kearney, M.W. (2019), rtweet: Collecting and analyzing Twitter data, Journal of Open Source Software,4,42, doi: 10.21105/joss.01829, https://joss.theoj.org/papers/10.21105/joss.01829

P. Kralj Novak, J. Smailovic, B. Sluban, I. Mozetic, Sentiment of Emojis, *PLoS ONE*, 10(12): e0144296, doi:10.1371/journal.pone.0144296, 2015.

Marchand P., Carroll I. (2019), Submit R Calculations to a 'Slurm' Cluster}, https://CRAN.R-project.org/package=rslurm

Mivule, K., Harvey B., Cobb C., El Sayed H. (2014), A Review of CUDA, MapReduce, and Pthreads Parallel Computing Models, *JISET - International Journal of Innovative Science, Engineering & Technology*, 1,8, 208-217

Mohammad, S.M. & Turney P.D. (2013), Crowdsourcing a Word-Emotion Association Lexicon, *Computational Intelligence*, 29, 3, 436–65.

Reyes-Ortiz, J.L., Oneto L., Anguita D. (2015), Big Data Analytics in the Cloud: Spark on Hadoop vs MPI/OpenMP on Beowulf, *Procedia Computer Science* (INNS Conference on Big Data), 53, 121–130

Pennebaker, J.W., Booth, R.J., Boyd, R.L., & Francis, M.E. (2015). Linguistic Inquiry and Word Count: LIWC2015. Austin, TX: Pennebaker Conglomerates (www.LIWC.net).

Plutchik, R. (1980), A General Psychoevolutionary Theory of Emotion, *Theories of Emotion*, 1.

R Core Team (2020), *R: A Language and Environment for Statistical Computing*, https://www.R-project.org

Tekumalla, R., Banda J. (2020), Social Media Mining Toolkit (SMMT). Under review. Genomics and Informatics.

Young, L., Soroka S. (2012), Lexicoder Sentiment Dictionary. Available at lexicoder.com.