Big Data Analytics - Systems and Methods

Portfolioprüfung Part 2: Implementierung

Prof. Dr. Benjamin Buchwitz

—

Gegenstand der Implementierungsaufgabe ist das Verfassen einer wissenschaftlichen Arbeit in Form einer Hausarbeit, die mindestens aus den folgenden Kernbestandteilen besteht: 1) Theoretische Grundlagen der implementierten Methode(n), 2) Architektur des konfigurierten Systems, 3) programmatischer Ansatz der Implementierung mit Erläuterung der wesentlichen Codebestandteile, 4) Benchmarking der Implementierung entlang der vorgegebenen Dimension.

Der Hauptteil der Arbeit diskutiert dabei ausgewählte Teile des Programmcodes im Kontext der zugehörigen Anwendung und die erziehlten Ergebnisse. Die Arbeit schließt mit einer ausführlichen Darstellung der erarbeiteten Benchmarkingergebnisse und einer kritischen Würdigung des gewählten vorgehens. Der *vollständige* und *lauffähige* Code für das Projekt wird als technischer Anhang zur Ausarbeitung eingereicht. Die verwendete Programmiersprache kann R oder Pyhton oder eine Mischung aus beiden Sprachen sein. Sollten Sie eine andere Sprache verwenden wollen halten Sie bitte kurz Rücksprache. Da es sich bei der Arbeit um eine wissenschaftliche Arbeit handelt, sind insbesondere die Grundsätze des wissenschaftlichen Arbeitens (präzise Formulierung, hohe Inhaltsdichte, Zitation, Formatierung, Qualität von Grafiken, etc.) zu beachten. Es wird ausdrücklich empfohlen die Arbeit **in Englisch** zu verfassen.

Die Kooperation als Gruppe pro Thema ist ausdrücklich erlaubt. Jede Gruppe reicht dabei nur **eine** Ausarbeitung ein und ergänzt die Arbeit um einem Anhang mit einer Übersicht der beigetragenen Eigenleistungen der Autoren.

Regression in Apache Spark

Apache Spark stellt per MLlib unter anderem Routinen zur Schätzung von linearen Regressionsmodellen zur Verfügung. Während Apache Spark auch komplexe lineare Modelle (z.B. Generalized Linear Models) unterstützt, beschränkt sich der Methodenteil der Aufgabe auf das klassische lineare Regressionsmodell (OLS) mit dem Mean Square Error (MSE) als Loss-Funktion. Entsprechend werden nur Modelle ohne Regularisierung oder weiterführende methodische Ausgestaltungen betrachtet, sodass komplexere Schätzmethoden (ála Stochastic Gradient Descent) von der Betrachtung zunächst ausgeschlossen werden können. Die relevanten in Apache Spark verfügbaren Methoden, einige Erläuterungen und der Hinweis auf die Optimierung per Normalengleichungen (Normal Equations) via WeightedLeastSquares finden sind in der Dokumentation an folgenden Stellen:

- https://spark.apache.org/docs/latest/mllib-linear-methods.html#regression
- https://spark.apache.org/docs/latest/ml-advanced.html#optimization-of-linear-methods-developer
- https://spark.apache.org/docs/latest/api/scala/org/apache/spark/mllib/regression/index.html

Ordinary Least Squares Estimation (OLS)

Wachsende Datenmengen führen dazu, dass sich Regressionsmodelle nicht auf Single-Node Systemen lokal schätzen lassen, sondern auf Systemen mit mehreren Nodes verteilt werden müssen. Das multiple Regressionsproblem

$$y = X\beta + \epsilon$$

mit den Bestandteilen

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad X = \begin{pmatrix} x_{10} & x_{11} & \dots & x_{1p} \\ x_{20} & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n0} & x_{n1} & \dots & x_{np} \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

zeigt deutlich, dass die Komplexität des Problems in den Dimensionsn n (Anzahl der Beobachtungen) und p (Anzahl der Regressoren) wachsen kann. Elementaren Operationen zur Handhabung großer Datenmengen und zur Anwendung mathematischer Operatoren stellt Apache Spark zur Verfügung. Aus Gründen numerischer Stabilität und ggf. algorithmischer Präzision werden Regressionsschätzer in der implementierungstechnischen Praxis jedoch nicht direkt über die in einführenden Lehrbüchern hergeleitete analytische Lösung $\hat{\beta} = (X^T X)^{-1} X^T y$ berechnet, sondern über Dekompositionsverfahren. Die verwendeten Verfahren sind insbesondere die LU Dekomposition, die QR Dekomposition und die Berechnung der OLS-Schätzer über die Moore-Penrose-Pseudoinverse und damit mittelbar über die Singularwertzerlegung (SVD).

Eine Einführung in das Thema liefern Hansen et al. und Higham. Zur Effizienz der Algorithmen führen Golub und Van Loan Argumente an. Druinsky und Toledo äußern sich zur Präzision bei der Lösung von Gleichungssystemen mittels Matrizeninversion.

- Hansen, Per Christian; Pereyra, Victor; Scherer, Godela (2013): Least Squares Data Fitting with Applications, Johns Hopkins University Press.
- Higham, Nicholas J. (2002): Accuracy and Stability of Numerical Algorithms, Second Edition, SIAM.
- Golub, Gene H.; Van Loan, Charles F. (2013): Matrix Computations, Fourth Edition, Johns Hopkins University Press.
- Druinsky, Alex; Toledo, Sivan (2012): How Accurate is inv(A)*b?, arXiv Preprint, https://arxiv.org/abs/1201.6035.

Implementierungsaufgabe: Distributed OLS

Kern der praktischen Anwendung ist die grundlegende Implementierung der konkurierenden Verfahren zur OLS Schätzung und die Beurteilung der Performance der Verfahrensanwendung und Präzision der Ergebnisse. Dabei sind u.a. die folgenden Schritte zu berücksichtigen:

- Implementierung der zugewiesenen OLS-Verfahrensvariante(n) unter Nutzung der in Apache Spark zur Verfügung stehenden **verteilten** Rechenoperationen. Die Dimension der Datenmatrix X ist $n \times k$, wobei n die Anzahl der Beobachtungen und k die Anzahl der Variablen ist. Die implementierte Verfahrensvariante muss sehr große Dimensionen von X für die Berechnung der OLS-Schätzer unterstützen, d.h. $n \gg 10^9$ und $k \gg 10^5$.
- Simulation eines großen Beispieldatensatzes für ein Regressionsproblem mit bekannten Parametern $\hat{\beta}$ für die Darstellung und den Vergleich der Performance der Implementierung der Verfahrensvarianten. Die simulierten k Variablen sollen dabei eine moderat ausgeprägte Kovarianzstruktur haben und sich hinsichtlich ihrer Varianz unterscheiden. Eine eklatante Verletzung der Gauss-Markov-Annahmen soll nicht vorliegen.
- Anschließend Durchführen des Vergleichs und Darstellung der Rechenzeit auf einem Cluster in Abhängigkeit der Mächtigkeit von n und k. Der Apache Spark Cluster kann dabei auf einem System simuliert und mit Docker (z.B. Docker Desktop bzw. Docker Swarm) oder einer anderen Containerlösung betrieben werden. Die Ergebnisse des Vergleiches werden auch grafisch dargestellt.

#————————-

Task Description

The exercise for the implementation is to generate a forecast for a collection of time series or a single time series. The dataset is either your own proposed time series dataset or the common dataset below.

The specific task is to choose and recommend a model or forecasting process that fits to your data *in the best possible way* and evidently generates suitable forecasts for a practically relevant horizon. Try all applicable forecasting methods you know including simple benchmarks. Do not forget to discuss the forecasting methods, examine their residuals and forecasting errors and present evidence that outlines why one model should be favored over another. The following bullet points aide in guiding you through the analysis:

- Your work should start with an introduction to the topic and answer the following questions:
  - Why are the topic and data set - you work with - of interest? Why is it important to make forecasts in this domain?
  - What is the exact forecasting problem and what distinguishes a good from a bad forecast (evaluation metrics)?
  - What is the aim of your work?
  - What is already known (from literature or practice)? From this and your forecasting goals infer a hypothesis. What do you assume and expect concerning the models behavior and their forecasts?
  - Give a *very brief* overview of the steps you will take in the following presentation/paper/poster.
- Describe the dataset including relevant variables and time series' you will examine:
  - Which variables are of interest? Describe at least the y and t variable.
  - Visualize the time series. What is the range of values? What is the granularity? Are the observations equidistant? How many observations exist? Are there any missing values in the time series (how many)? Are there any further peculiarities?
  - How is the time series characterized? Trend? Seasonal patterns? Cycles?
- Then, try different forecasting methods and examine the results. Do not forget to split your data set in a training and test set before evaluating the forecasts. The chosen split needs to be justified by proper argumentation (especially if you are not using time series cross validation).
  - In order to select an *appropriate* forecasting method start with the simple forecasting methods: Is one of the simple forecasting methods already appropriate? Why or why not? What is the benefit of more complex methods and how do you use them to your advantage? Provide empirical and theoretical arguments. Plot all values of interest to your argumentation. Always examine the results and consider a plethora of evaluation measures (eg. RMSE, MAE, MAPE and MASE).
- In the end of your presentation/paper/poster, give a conclusion.

Remarks

Depending on your specific dataset not all of the bullets above may be fully appropriate. So it is up to you which points are the most relevant to your data and support your argumentation in the best way possible. Generally there are two possible paths.

- **Single Time Series:** And in depth analysis and forecasting a single time series is usually done when the forecast are of *higher value*. This means your work should go into all the details of the series and in fine tuning of the models. This is especially true if the series of interest is (relatively) short. So, when dealing with a single series the analysis and the corresponding work can feature every little aspect in great detail.
- **Collection of Time Series:** When working with a collection of time series each time series in the collection and therefore each forecast is of *lower value*. It is also unlikely that one model fits all time series as they usually exhibit some differences that can only be leveraged by specialized models. So when dealing with a collection of time series the emphasis is on the process of model selection rather than on the peculiarities of a single model. Results and evaluation measures are usually highly aggregated, plots contain multiple series/forecasts/measures and the whole analysis is supported by adequate (representable) examples.

General Dataset

If you have not proposed a dataset or your proposed dataset is not suitable for the domain of the course, you must use the following dataset:

German Petrol Prices https://dev.azure.com/tankerkoenig/_git/tankerkoenig-data

To generate a subset of time series please follow these steps:

1) Download the dataset
2) Choose a selection criteria for a subset of petrol stations (eg. all stations in a specific region or a specific brand)
3) Convert the event based observations into a time series and choose a suitable granularity (eg. hourly, daily, ... the resulting series should exhibit substantial movement)
4) Aggregate the generate time series into a small subset by choosing an aggreation level (eg. type of petrol) and an aggregation measure (eg. arithmetic mean)
5) Use the result small collection for the case study

Formal Criteria

Each case study has to be solved **individually** and **autonomously**. Your results must be submitted via the link in the Moodle course and must follow the specified format. The submitted **zip**-file (*#######.zip*) must contain at least the following:

- The **raw** data including a comprehensive description of the dataset and the variables used for the case study including all code and functions that were used to generate the time serie(s) for the actual forecasting task.
- The complete code for all analyses including descriptive statistics, model selection and estimation as well as derived evaluation results.
- Proper documentation for the complete code and all steps that guide the reader through the flow of your analyses.
- The poster in the final **pdf**-format as well as its editable source files.
- A signed declaration of authorship (Ehrenwörtliche Erklärung).