Using PySpark to Estimate Value at Risk (VaR)

Devere Anthony Weaver
Towson University
Towson, USA
dweave8@students.towson.edu

***Abstract*—Risk can be defined as the uncertainty surrounding outcomes. Financial risk must be managed by financial institutions and other businesses to remain solvent. Hence, measuring financial risk is one of the most important tasks an organization can undertake. There exist a number of metrics to measure risk and just as many ways to compute each; however, this project will focus on the most prominent measure of financial risk, value at risk (VaR). For this project, to compute VaR, PySpark will be used to perform operations on historial financial data.**

I. Introduction

Financial risk management is a critical component of any organization's strategy, ensuring long-term stability, growth, and survival in volatile markets. It involves identifying, assessing, and prioritizing risks that could negatively affect an organization's financial health, such as market fluctuations, credit defaults, operational disruptions, or legal liabilities. Effective risk management helps firms mitigate losses, optimize returns, and maintain investor confidence. Additionally, regulatory environments increasingly require robust risk management practices, making it essential for legal compliance and corporate governance. By implementing a sound financial risk management strategy, organizations can safeguard assets, improve decision-making, and ensure they are better prepared to navigate uncertainties in the financial landscape.

II. Description of Data

The data used for this will consist of a portfolio of equities. The data will be pulled from Yahoo Finance's historical equity returns data. Currently, the specified size of the portfolio and specific stocks are still to be determined.

III. Preliminary Literature Review

The first paper consulted [1] provided an overview of how modern machine learning algorithms can be applied to data to clean and model financial problems. While none of the machine learning techniques from this paper may be used for this project, it was incredibly useful in providing a high-level review of the financial risk taxonomy and explaining why practitioners need to come up with various ways to model financial risk.

From there, I then began to research some of the ways that distributed computing can be useful in the field of financial risk management and stumbled upon [2] where the authors present two different methods on how to compute VaR, one of them being using Monte Carlo methods.

To find an applied example of how to parallelize Monte Carlo simulation using Spark, I then consulted [3] while also brushing up on the mathematical model for VaR in [4].

IV. Proposed Methodology

Currently, the proposed methodology consists of downloading historical returns data for a given set of equities. Once the data are downloaded successfully, they will need to be pre-processed. Pre-processing of financial returns data includes creating a dataframe to hold the daily returns by computing the fractional percent changes of each time series observation.

Next, the compounded daily rate will need to be computed based on 360 days per year. There should be no missing values in Yahoo Finance's time series of returns; however, if they are encountered, an appropriate value will be filled in (e.g. the previous day's returns since).

To compute the VaR, Monte Carlo simulation will be used. Monte Carlo simulation takes uncertainty on input variables of a model and computes probability distributions. While VaR can be computed using solely historical simulation and variance-covariance, Monte Carlo was chosen since Spark is an ideal tool for Monte Carlo simulation, because the technique is naturally massively parallelizable [1]. The final results of the VaR computations will then be tested against some yet to be determined evaluation criteria.

To obtain the data, the Python `yfinance` package will be used for ease of use. The following typical scientific Python libraries will be use for data processing, analysis, and Monte Carlo simulation: `numpy`, `pandas`, `statsmodels`.

V. Anticipated Outcome

PySpark should be able to handle this kind of computation as others have demonstrated similar Monte Carlo computations using Spark with Scala.

My goal here is to not necessarily reinvent the wheel, given the lack of Spark familiarity and time constraints. Instead, the goal is to learn how to use Spark and applied to a field I'm familiar with so that I can learn how to deal with a parallel processing system.

REFERENCES

[1] A. Mashrur, W. Luo, N. A. Zaidi, and A. Robles-Kelly, "Machine Learning for Financial Risk Management: A Survey," IEEE Access, vol. 8, pp. 203203–203223, 2020, doi: 10.1109/ACCESS.2020.3036322.

[2] Y. Li, A. Li, and Z. Liu, "Two Ways of Calculating VaR in Risk Management ——An Empirical Study Based on CSI 300 Index," Procedia Computer Science, vol. 139, pp. 432–439, 2018, doi: 10.1016/j.procs.2018.10.259

[3] S. Ryza, Ed., Advanced analytics with Spark: patterns for learning from data at scale, Second edition. Beijing: O'Reilly, 2017.

[4] J. Hull, Risk management and financial institutions, Sixth edition. in Wiley finance. Hoboken, New Jersey: Wiley, 2023.

[5] D. K. Kanungo, Probabilistic machine learning for finance and investing: a primer to Generative AI with Python, First edition. Sebastopol, CA: O'Reilly Media, Inc, 2023.