PROOF OF OPTIMALITY FOR A DECENTRALISED EO DATA PROCESSING ARCHITECTURE

Robert Cowlishaw, Annalisa Riccardi

University of Strathclyde
Mechanical and Aerospace Engineering
Glasgow, UK

Ashwin Arulselvan

University of Strathclyde
Management Science
Glasgow, UK

ABSTRACT

Earth Observation (EO) data is large and often processed in a very centralised manner. Through the decentralisation and distribution of data processing, a more neutral and automated system can be created, while incentivising a more diverse set of data sources. This can help lower the initial barrier for new data providers and help with decreasing the time it takes for data to be created for systems such as Satellite-based Emergency Mapping. Building such architecture on a decentralised network comes with difficulties, such as merging centralised data sources together, building trust or reputation on a trustless system, and building processes and methods that require low enough computational cost to be executable on distributed networks. This paper discusses how to offload and on-load data onto a distributed network to overcome these computational challenges.

***Index Terms*—** EO Data, Data Merging, Reputation, Consensus, Decentralised and Distributed Architecture

1. INTRODUCTION

Earth Observation (EO) data processing is a key step in the EO data value chain, as it takes high density often noisy information and converts it into usable data. These computations can be simple such as merging data from different sources to more expensive tasks such as teaching machine learning models. Due to the size of EO data, data processing is a computationally expensive process no matter the complexity of the task itself. This brings with it considerations such as where EO data should be processed, onboard satellites, or on the ground, and who is allowed to process data as a single change somewhere in the computation could drastically affect the end result, allowing accidental or malicious change of the output data.

Through decentralisation of EO data creation, processing, and distribution, a fairer platform can be created with a cheaper onboarding experience for any new resource providers to join. This promotes, or incentivises, the creation of new, diverse, data and creates a more robust and automated system increasing the reliability and speed of data creation and access to such data. Without a central authority managing such a system, an impartial, neutral, method of communicating data, and data requirements, is created in a sector that holds so many different geopolitically disinclined organisations and nations.

To create a decentralised system, trust between separate entities is required, or the distribution of infrastructure to many people is required, creating consensus over a large enough group, therefore creating a trustless system. Due to the organisations and nations involved, the first method of trust being held between entities is unlikely, therefore distribution of the infrastructure is the most forthcoming option. Trust can therefore be placed in the decentralised network, through mathematical and cryptographic proofs, to prove that a process is executed based on a set of predefined rules. From this three fundamental stages in EO data processing, to overcome the cost of large scale data storage and data processing on a distributed network, appear. This paper discusses reputation and a method of overcoming difficulties of large scale data and investigates a method of reducing the cost of EO data processing on such a network. It does this by selectively offloading and on-loading computations off and on the distributed/decentralised network to allow minimise difficulty while maximising security. Figure 1 shows the basic architecture of the proposed solution.

1.1. Data Oraclisation

Oraclisation, the method of providing data to a decentralised network from multiple centralised sources, allows for non-distributed information to become distributed by creating consensus through the use of redundancy as seen in figure 1. By using an oracle, difficult computational problems can be completed on a non-distributed computer/network (off-chain) multiple times by multiple sources to provide secure provable information. Through this technique, large scale EO data processing can first be simplified off-chain, and then further processing can be done on a distributed network (on-chain) with the data that has been oraclised.

doi:10.2760/46796

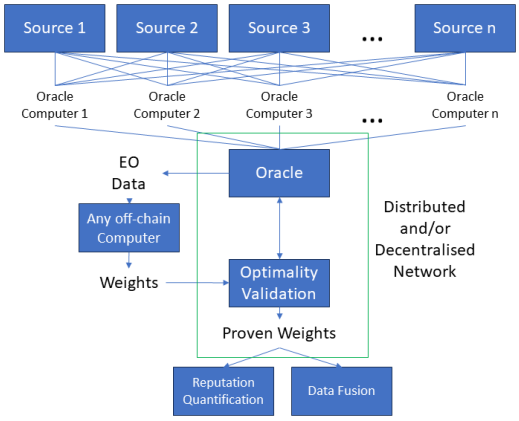


Fig. 1. Basic architecture of data processing stages to distributed and what is not required to be distributed

1.2. Optimality Validation

Computing optimal solutions to optimisation problems, often large in problem size, is another task that is usually unsuited for distributed networks, again due to the computational cost. To reduce this complexity, the more computationally intense task (*Optimisation Problem*) can be completed off-chain and then the result (*Optimal Solution*) can be validated on-chain. To prove that this works, this paper looks at an example of processing EO data and how the validation of the optimal solution is cheaper than the *Optimisation Problem*.

1.3. Comparison of Root Mean Square Error

Root Mean Square Error (RMSE) is often used as the metric for comparison of processed data to ground truth in many fields such as aircraft roll detection in [5] to satellite data processing [9]. Even with newer methods of satellite data processing such as [7], RMSE is a core element. As it provides a metric for difference between dataset it can also be used to measure the accuracy between two datasets. Weighted averages, similar to those shown in [9] can be used to minimise the RMSE as shown in 2.

1.4. Reputation Management

Reputation or trust is quantification of the productivity of a user towards the goal of a network or task. AS well as being used in everyday life it has become an important part of the internet [8]. On decentralised/distributed systems, reputation is often required in certain situations to allow for users to interact. [4] and [6] both design systems for reputation management across distributed systems (P2P network before Web3). Although reputation management is often considered as a metric of the network itself, such as in [4] and [6] with many more in [1], social trust [3] can be produced from how trustworthy the actors using the network are. From measuring the consensus using RMSE, within both consensus on data (data oraclisation) and optimality validation, a reputation or measure of this social trust can be produced.

2. EXAMPLE PROBLEM DEFINITION

First the *Optimisation Problem* must be defined. For this example, a method of measuring consensus across images from multiple sources of a similar data set is undertaken to generate a fused data set with increased accuracy than that of a single input image. To find a merged output image a weighted average scaling is applied to each data set to reduce RMSE over all cross comparisons. Weight averaging is used as it doesn't change the data given by each source, but just its contribution to the final output image. The following assumptions are made about the input data for each source: *i)* the EO data contains binary data, *ii)* the EO data from each source are of the same resolution, *iii)* all EO data is of the same area/region; are geographically aligned and data fusion occurs synchronously when all data has been acquired, *iv)* all EO data is of identical shape.

From these we can apply three conditions to the RMSE calculation: *i)* accurate data is promoted, *ii)* inaccurate data is penalised, *iii)* missing data is not penalised or promoted. These conditions do not impose a strict penalty on lack of precision thereby encouraging newcomers with access to cheaper satellite infrastructure to participate. True and false negatives are ignored as this would be further penalisation with a similar impact.

2.1. Data Oraclisation Problem

We have a set of sources, L, with each source $x \in L$ providing a $m \times n$ data, $\boldsymbol{A}^x$. We represent the combined dataset from all sources by $\boldsymbol{A}$. We define $\boldsymbol{\alpha} := \{\alpha_1, \ldots, \alpha_\ell\}$ as the vector of weights associated with each source, where $\ell = |L|$. The general equation for RMS disparity of two sources of data $x \in L$ and $y \in L$

$$RMS(\boldsymbol{A}^x, \boldsymbol{A}^y) = \sqrt{\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij}^x - A_{ij}^y)} \quad (1)$$

where A_{ij}^x is the data entry in row i and column j of data $\boldsymbol{A}^x$ coming from source x. Using the assumptions to achieve the conditions set out we define the relative truth of source x with respect to source y as:

$$R_y^x = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^x A_{ij}^y}{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^x} \quad (2)$$

We can now re-write the RMS function with respect to the relative truths as follows:

$$RMS(\boldsymbol{A}^x, \boldsymbol{A}^y) = \sqrt{(1 - R_y^x)(\alpha_x)^2 + R_y^x(\alpha_x - \alpha_y)^2} \quad (3)$$

doi:10.2760/46796

We define the asymmetric $\ell \times \ell$ square matrix of relative truths by $\boldsymbol{R}$, where the entry in row x and column y of the matrix is, R_y^x, the relative truth of x with respect to y. Note that we are now working with the compact relative information data $\boldsymbol{R}$, rather than the original $m \times n \times \ell$ data $\boldsymbol{A}$, where $m, n >> \ell$.

This allows for the previous steps to be completed off-chain and $\boldsymbol{R}$ to be produced on-chain from the data oraclisation. This means that the computationally difficult summation can be done off-chain as well as the data storage requirements on-chain being drastically reduced. Different methods can be taken to compute the $\boldsymbol{R}$, which may suit different types of data sources.

2.2. Optimality Validation Problem

RMS of the cross comparison of $\boldsymbol{A}$ can be calculated from the RMS in equation 3 for all ordered pairs of sources (i, j) in L

$$f(\boldsymbol{\alpha}, \boldsymbol{R}) = \sqrt{\frac{1}{\ell(\ell-1)} \left(g(\boldsymbol{\alpha}, \boldsymbol{R}) + h(\boldsymbol{\alpha}, \boldsymbol{R})\right)} \tag{4}$$

$$g(\boldsymbol{\alpha}, \boldsymbol{R}) = \sum_{i=0}^{\ell} \sum_{j=0}^{\ell} \left((1 - R_j^i)(\alpha_i)^2\right) \tag{5}$$

$$h(\boldsymbol{\alpha}, \boldsymbol{R}) = \sum_{i=0}^{\ell} \sum_{j=0}^{\ell} \left((R_j^i)(\alpha_i - \alpha_j)^2\right) \tag{6}$$

To minimise $f(\boldsymbol{\alpha}, \boldsymbol{R})$, $\boldsymbol{\alpha}^*$ is found to be given by equation 7 where $i \neq j$, $0 \leq i < \ell$, $0 \leq j < \ell$ and is repeated ℓ times with unique i and j combinations.

$$\frac{df}{d\alpha_i} = \frac{df}{d\alpha_j}, \quad \forall i, j \tag{7}$$

where $\frac{df}{d\alpha_x}$ is the derivative of f with respect to α_x. Solving for $\boldsymbol{\alpha}$ satisfying (7) is equivalent to solving a system of linear inequalities

$$\boldsymbol{B\alpha} = 0 \tag{8}$$

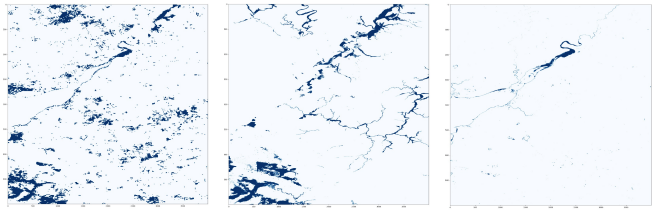
where the matrix $\boldsymbol{B}$ is determined by the relative truth matrix $\boldsymbol{R}$.

Therefore, by using a proven $\boldsymbol{\alpha}$, from $\boldsymbol{R}$ given by data oraclisation with a high enough level of consensus to be assumed to be true, the accuracy or consensus between each EO data source is quantified in a provable manner on a decentralised architecture.

3. *OPTIMISATION PROBLEM* CASE STUDY

3.1. Real Data Test

To test the system, and its optimality, three flood data sources were merged and their accuracy quantified, the data coming from a private company[1], the local government database[2] and an EU commision database[3] in this case. This range of sources gives varied data on what is considered flooded, with different levels of accuracy. In figure 2, the inputs data can be seen. This area of 3°to 2°West, 51°to 52°North was chosen due to the high level of flooding that occurs in this area.



(1) (2) (3)

Fig. 2. Input images from varying sources, of coordinates 3°to 2°West, 51°to 52°North with a resolution of 4000x4000 pixels or a Ground Sample Distance (GSD) of 27.75m. (1. Private Company Source, 2. Local Government Source, 3. EU Commission Source)

From this, $\boldsymbol{R}$ and $\boldsymbol{\alpha}$ can be calculated from equation 2 and equation 8 respectively.

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0.25613333 & 0.05081417 \\ 0.33729138 & 1 & 0.03541658 \\ 0.52717319 & 0.27902039 & 1 \end{bmatrix} \tag{9}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} 0.31327176 & 0.31492591 & 0.37180233 \end{bmatrix} \tag{10}$$

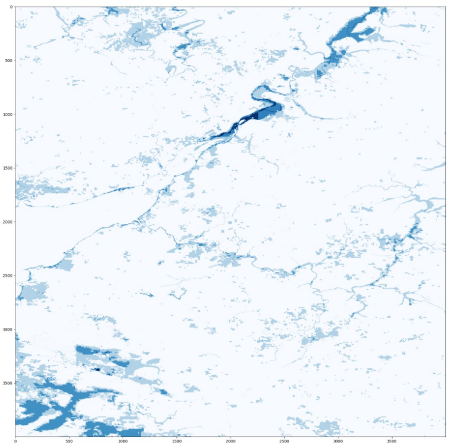


Fig. 3. The final output image with the given weights applied to each input

[1] https://global-flood-database.cloudtostreet.ai/ Combined Specific Flood Event Data

[2] https://www.data.gov.uk/ Historical Flood Data

[3] https://global-surface-water.appspot.com/ Source: EC JRC/Google Maximum Water Extent Data

doi:10.2760/46796

Through data oraclisation, the storage requirements on-chain for the input images change from 11.4MB (3 4000x4000 pixel images with a bit depth of 2) to 0.5625KB (9 256bit integers) reducing the positional data from the images but storing the ratio of overlapping values, all that is required for the *Optimisation Problem*. From figure 2 and the given $\boldsymbol{R}$ in equation 9 determined for these inputs, the lack of data provided by the third source can be easily seen by the values in the 3rd column. However, the data it provides is accurate, as seen by the large values reported in the 3rd row. This is supported also by the values of $\boldsymbol{\alpha}$ given in equation 10 where the corresponding value (3rd value) is larger than the others. In figure 3 the merged datasets with the weight averages applied can be seen. The darker areas show where the highest consensus lies, while the lighter shades show where there is lesser consensus. The white shows where no sources reported any data.

3.2. Computational Requirements/Gas Measuring

To test the computational requirements of the optimality validation vs the calculation of the *Optimisation Problem*, smart contracts were designed for testing these algorithms on a distributed network. The major blockchain network for distributed application development as of writing this paper is Ethereum [4]. The smart contracts are written in Solidity and are openly accessible at this link [5]. To test the computational requirements, a measure of Gas (the metric for execution difficulty on Ethereum) is used. Ethereum has a maximum gas of 30 million and therefore a smart contract that requires more than this to execute will never work [6]. In figure 4 it can be seen that Optimality Validation (Validate Alpha) uses much less gas as well as handling a lot higher number of sources before the maximum size of a block is reached, than that of *Opimisation Problem* solution (Solve for Alpha).

4. CONCLUSION

This paper has discussed a methodology to reduce the computational and storage cost on-chain by selectively offloading certain tasks to be run off-chain while still remaining secure by distribution. An example of reputation quantification on a decentralised/distributed network architecture has also been discussed as well as how consensus can be measured through EO data. This decentralised data processing is the next step towards collaborative efforts to provide data in a faster, fairer, more neutral and scalable way while lowering the barrier for EO data providers from all nations and organisations.

[4] Ethereum https://ethereum.org/

[5] https://github.com/strath-ace/smart-contracts

[6] Ethereum creates a block approximately every 12 seconds (https://ycharts.com/indicators/ethereum_average_block_time)

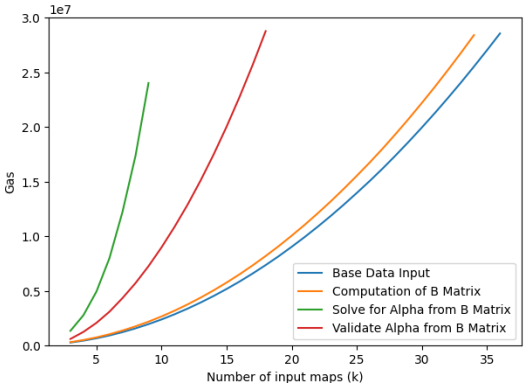


Fig. 4. Gas Required for Solving Alpha vs Validating Alpha, tested on a distributed network (Ethereum test-net [4])

REFERENCES

[1] J. Cho, A. Swami, and I. Chen. A survey on trust management for mobile ad hoc networks. IEEE Communications Surveys Tutorials, 13(4):562–583, 2011. doi: 10.1109/SURV.2011.092110.00088.

[2] A.H. Robinson and C. Cherry. Results of a prototype television bandwidth compression scheme. Proceedings of the IEEE, 55(3):356–364, 1967. doi: 10.1109/PROC.1967.5493.

[3] S. Trifunovic, F. Legendre, and C. Anastasiades. Social trust in opportunistic networks. In 2010 INFOCOM IEEE Conference on Computer Communications Workshops, pages 1–6, 2010. doi: 10.1109/INFCOMW.2010.5466696.

[4] Swamynathan, G., Almeroth, K.C. Zhao, B.Y. The design of a reliable reputation system. Electron Commer Res 10, 239–270 (2010). doi: 10.1007/s10660-010-9064-y.

[5] H.P. Bruckner, C. Spindeldreier, H. Blume, E. Schoonderwaldt, and E. Altenmuller. Evaluation of inertial sensor fusion algorithms in grasping tasks using real input data: Comparison of computational costs and root mean square error. In 2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks, pages 189–194, 2012. doi: 10.1109/BSN.2012.9.

[6] J. Jaramillo and R. Srikant, Darwin: Distributed and adaptive reputation mechanism for wireless ad-hoc networks. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking, pages 87–98, 2007.

[7] S. Kim, A. Sharma, Y. Y. Liu, and S. I. Young. Rethinking satellite data merging: From averaging to snr optimization. IEEE Transactions on Geoscience and Remote Sensing, 60:1–15, 2022. doi: 10.1109/TGRS.2021.3107028.

[8] C. Tennie, U. Frith, and C. Frith. Reputation management in the age of the world-wide web. Trends in Cognitive Sciences, 14(11):482–488, 2010. doi: 10.1016/j.tics.2010.07.003.

[9] O. Yahia, R. Guida, and P. Iervolino, Weights based decision level data fusion of landsat-8 and sentinel-l for soil moisture content estimation. In IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pages 8078–8081, 2018. doi: 10.1109/IGARSS.2018.8518027.