Transaction Fee Optimization
MAS.S62 Cryptocurrency Engineering & Design

Maxime Amram & Jeremy Toledano
May 16, 2018

1 Bitcoin Transaction Fee

1.1 Fee Market

For every transaction inside a mined block, Bitcoin miners receive the **difference between the transaction's inputs and outputs**. This reward per transaction is referred to as the **transaction's fee**. Such a fee can be seen as a compensation paid by the sender to the miner in exchange for space taken inside a block. For that matter, fees are expressed in **satoshis per byte**.

Since block size is limited to 1 MB, space for transactions is scarce, and miners are well advised to prioritize transactions in order to maximize revenue. In practice, miners will always start by appending the transactions with the highest fees to their blocks. Thus, the fee, which is **chosen by the sender of the transaction**, will directly impact the time the transaction will remain in the mempool. As a result, the fee market can be seen as an **auction market**, where the highest bidder waits the shortest time. Finding the optimal compensation for a transaction is both crucial and quite challenging, even for regular Bitcoin users.

In Bitcoin, fees are not the only incentive for miners. For every block they mine, they also receive a reward, which halves every 210,000 blocks. As this block reward decreases, fees will eventually become the only **economic incentive** for miners to keep mining. Thus, fees are central to the blockchain's long-term viability. On the users' side, the fee market has a great impact on user experience: high fees, for instance, would render the system unsuitable for micropayments. For these reasons, **predicting the optimal fee** for a given transaction is essential and will require more research in the near future.

1.2 Current Fee Estimation

There is no one-size-fits-all solution for estimating transaction fees, as they are highly dependent on the user's requirements in terms of waiting time. Today, the following three main estimation methods are primarily used.

- **History-Based Methods**
  Since version 0.15, Bitcoin Core includes a built-in fee proposition feature. It allows users to choose between two modes: conservative and economical. Users have to specify a **target number of blocks** they are willing to wait for before their transaction is confirmed. Given this target, an estimate of the appropriate fee is proposed based on the most recently mined blocks' fees. Since they rely on historical data, history-based methods adapt to the volatility of the fee market, but are not reactive to sudden changes in the network.

- **Trial and Error Methods**
  If a transaction's fee has been underestimated, the sender can adjust it by broadcasting new transactions through **Child Pays for Parent** (CPFP) or **Replace by Fees** (RBF). Nonetheless, CPFP takes up unnecessary space in a block and complicates the dependency graph of transactions. As for RBF, it is double spending and thus contributes to flooding the network.

- **Mempool-Based Methods**
  Currently, the best way to estimate transaction fees is to look at the current state of the mempool. Thereby, a decision can be made using the **same data as miners**, who explore the mempool to choose which transactions to include in their blocks.

1.3 Improvement Opportunity

After taking a closer look at the mempool data (Figure 1), we noticed that over the past two months, more than 55% of the time, there was less than **1 MB of transaction data in the mempool**. Since this is below the maximum block size, in theory, any transaction in the mempool would be included in the next block, regardless of its fee rate. The goal of this project is to anticipate such events and help users optimize the fees they are paying to miners.

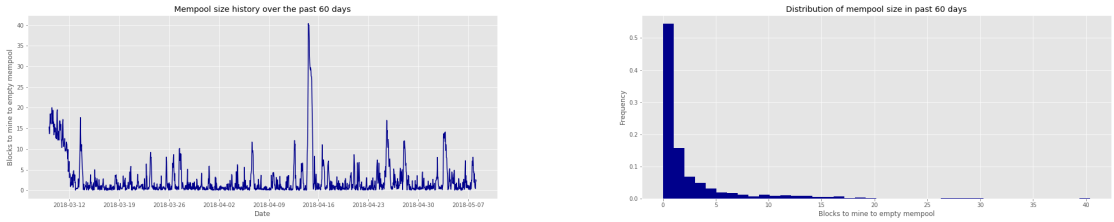

Figure 1: Mempool Exploration Data

2 Predicting Future Mempool State

2.1 Data Gathering and Hypotheses

We started off by gathering mempool data from two sources:

- Blockchain.info, which shares blockchain data and statistics;
- BlockSci, a Python library that gives access to blockchain data through AWS.

Using these sources, we were able to get enough historical data from the mempool to run our models. In the future, we would need to **run a full node** in order to adjust the granularity of the data measurements, a crucial step in verifying the results of our study.

In the meantime, we made a few simplifying hypotheses:

- We assumed that miners were rational;
- We did not take into account the dependencies between transactions;
- We neglected the size of a transaction with respect to the size of the entire mempool.

2.2 Mempool Size Direct Prediction

As Figure 2 suggests, there seems to be **patterns in the variation of the mempool size** on seasonal, weekly, and daily time frames. We used **machine learning** and **deep learning** algorithms to learn these patterns and to **predict the mempool size in the following hour**.

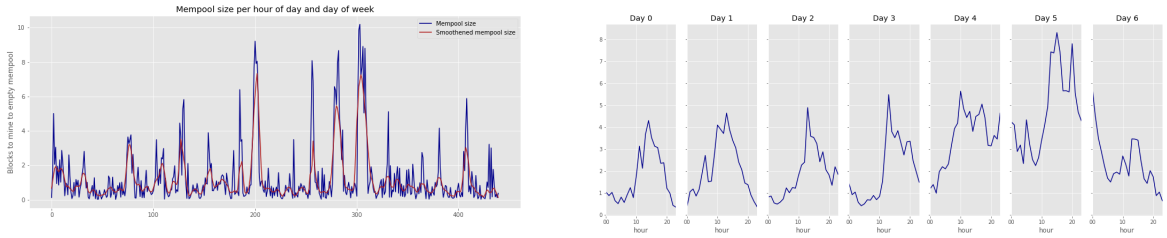

Figure 2: Mempool Size Patterns

Model	Out-of-Sample R^2
Baseline	0.7216
Random Forest	0.7089
Ridge Regression	0.7340
LSTM	0.7396

Table 1: Mempool Size Direct Prediction Results

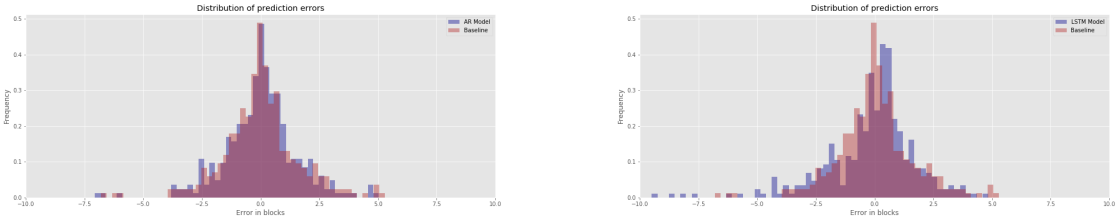

Figure 3: Distribution of Errors in Mempool Size Direct Prediction

We used **auto-regression** features to train different machine learning models, as well as a **Long Short-Term Memory** (LSTM) neural network. The results are presented in Table 1 and Figure 3, and compared to a baseline model which predicts the future mempool size as its current size.

Even though the LSTM neural network yields the best results, the prediction is not sufficiently reliable to be used in our final model. This may be due to the fact that such neural architectures typically need more data to learn complex patterns, or simply because the **randomness in block mining** blurs the signal. We thus decided to take another approach and to consider the **arrival rate of transactions in the mempool**, as they are less random by nature.

2.3 Transaction Arrival Rate Prediction

The transaction arrival rate shows **seasonal, weekly, and daily patterns** that are even clearer than in the case of mempool size (Figure 4). Let us then **predict this transaction arrival rate** and use it to **infer the future mempool state**. To this end, we trained the same models as in the previous section and recorded the results in Table 2 and Figure 5. We concluded that the best model's prediction would be sufficient for our mempool state inference.

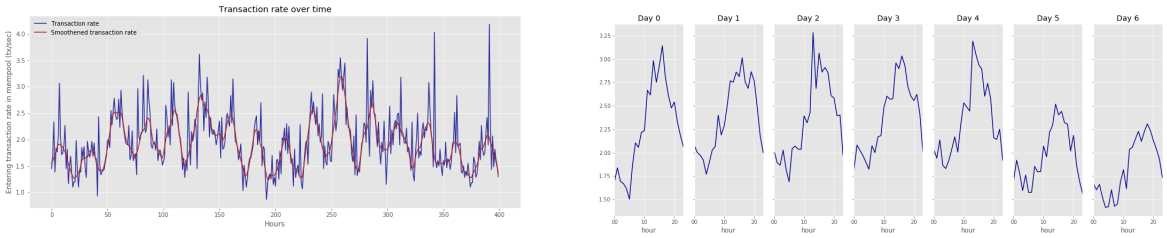

Figure 4: Transaction Arrival Rate Patterns

Model	Out-of-Sample R^2
Baseline	0.0456
Random Forest	0.3353
Ridge Regression	0.3801
LSTM	0.3745

Table 2: Transaction Arrival Rate Prediction Results

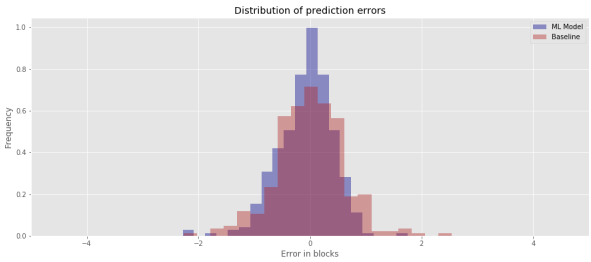

Figure 5: Distribution of Errors in Transaction Arrival Rate Prediction

2.4 Monte Carlo Simulations

Given the current mempool state and the current arrival rate of transactions in the mempool, we can estimate the evolution of the arrival rate and hence predict the future state of the mempool. To this end, we decided to model mining and transaction arrival in the mempool as **slowly time-varying Poisson processes**. Thence, the mempool state becomes a **Markovian queueing process** that can be inferred with Monte Carlo methods.

We set the time frame of a **Monte Carlo simulation** as one hour. We represented one sample simulation on the left-hand side of Figure 6, as well as a collection of 100 simulations on the right-hand side. It is interesting to note that the trajectory is piece-wise concave. Indeed, in that particular case, the estimation of the future transaction rate was lower than the current one.

Even though the precision of the simulation is not very accurate – which is not surprising given the inherent randomness of the process – the Monte Carlo model gives us key insights on what happened in the mempool during the one hour time frame. For instance, our model can estimate the **probability** of the mempool size being below 1 MB in the next m minutes for all $m \in \{0, \ldots, 60\}$, which was our initial goal.

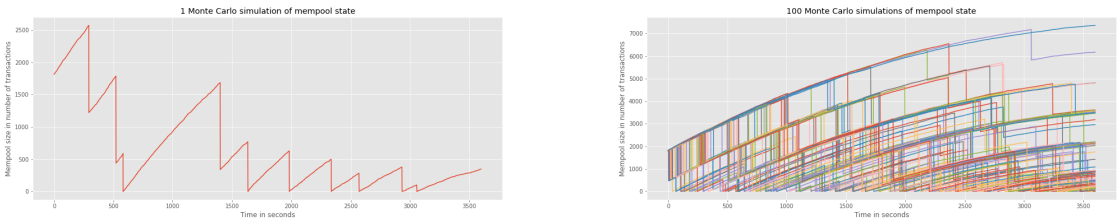

Figure 6: Monte Carlo Simulations

3 Results and Discussion

Our methodology allowed us to compute new metrics that are useful to users, and that should guide them in deciding whether to pay a high fee or to wait for the mempool to empty itself. It is interesting to note that our method is particularly suitable **after surges in the mempool**, when the transaction rate is back to normal and current methods fail to predict that the mempool size is about to decrease.

These first results are very promising, but a few comments are worth mentioning regarding their validity. First, we did not have the chance to verify the estimations of the metrics that we derived from the Monte Carlo simulations, simply because we did not have access to the mempool state within an hour. As mentioned earlier, running a full node to collect the data ourselves would address this critical issue. Similarly, the data we gathered constrained us to compute predictions on a time frame of one hour. Running a full node would allow us to choose a **shorter time frame** in order to **minimize the influence of external random events**. We believe that our models would yield better results on more granular data.

Secondly, as a further development, it could be interesting to take a closer look at the **distribution of fees in the mempool** instead of simply considering its size. This would allow us to develop an optimal framework that not only completes current fee estimators but also comprises them.

Thirdly, various improvements could be implemented, including **resistance to attacks**. For instance, miners could flood the mempool with fake transactions to simulate saturation, and our solution needs to prevent this from happening. Furthermore, it is crucial to think about how our solution scales from an economic and a game theoretical standpoint in order to fully understand how it could affect the network as a whole.

In the past, fees have not been a top concern for the Bitcoin community. Nevertheless, with the **growing popularity** of the network and the **decreasing block reward**, fee estimation is to become a fundamental element of the blockchain. We believe that more research needs to be conducted in order to develop a better understanding of this multifaceted topic.