Quantification of NMR Relaxometry Data with Machine Learning

Shinjer Li[1], Daniel Vasiliu[2], Tyler Meldrum[1]

[1]William & Mary Department of Chemistry
[2]William & Mary Department of Data Science

Theory

$$S(t) = \sum_{i=1}^{n} A_i * e^{-\frac{t}{T_{2,i}}}$$

NMR relaxometry data is a powerful technique that can be used to classify materials based on their "spin-spin" relaxation time constant, T_2. The data takes the form of an exponential decay that can be modeled with the equation above where the signal is a function of time and A is the weighting term. Samples typically have mono, bi, or tri exponential character where n would be 1, 2, or 3, respectively. Quantifying T_2s from relaxometry data is difficult and existing methods are not perfect.

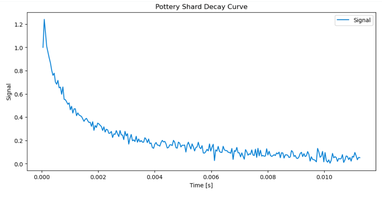
Workflow

1. Sample

We tested various samples on our single-sided NMR magnet (0.331 T or 13.325 MHz ^{1}H). Some of them included rubber, water, and pottery samples from Jamestown, VA.

Figure 1. Pottery shard samples from Jamestown, VA.

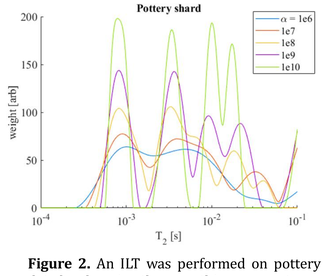
2. CPMG



A CMPG sequence is used to collect relaxometry data from our single-sided NMR. The pottery data was measured using 512 echoes, echo time of 44 µs, and repetition time of 1.3 seconds.

Figure 1. CMPG sequence of the pottery shard sample, KEC_RS_005_2. SNR ~ 55.

3. Data Processing



Inverse Laplace Transforms (ILT) have traditionally been used to quantify T_2s. However, this method requires a subjective smoothing implementation, which may impact accuracy and reproducibility. Additionally, quantifying uncertainties from ILTs are difficult as many of the peaks tend to be broad and overlapping. The matrix pencil is another method that can be used to quantify T_2s, but this approach also requires human intervention.

Figure 2. An ILT was performed on pottery data by changing the smoothing parameter, α, from 1e6 to 1e10.

Motivation

Existing methods to quantify T_2s require human intervention, which adds potential bias in results. We are proposing more systematic and reliable way of quantifying T_2s from relaxometry data through the use of machine learning. These findings can be expanded to understand other decay-based systems in the future.

Results

Method	Simulated Data	Pottery Shard Data

Sparsity Matrix

The top-down approach uses the sparsity matrix algorithm and was performed on simulated and real data. While there is agreement with simulated data and other methods, the contributions of each T_2 in this method are not yet accurate. The computational time of ~ 30 seconds is far quicker than the MCMC and provides similar findings to the ILT.

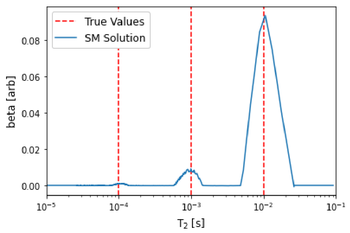


Figure 3. SM results for simulated tri-exponential data.

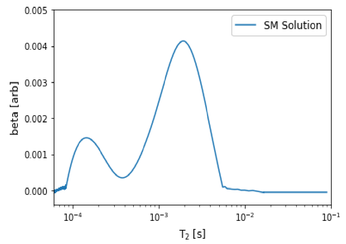


Figure 4. SM results for pottery shard data.

MCMC

An MCMC was performed on simulated and real data. The simulated data served as a proof of concept and allowed us to test how our results change depending on SNR and T_2 separation. Parameters used for Figures 3 and 4 were a tri-exponential fit, 4000 steps, and 60 independent walkers. The mean acceptance rate was consistently around 20-30% which an autocorrelation rate of ~ 60 and an average of 4 minute computational time.

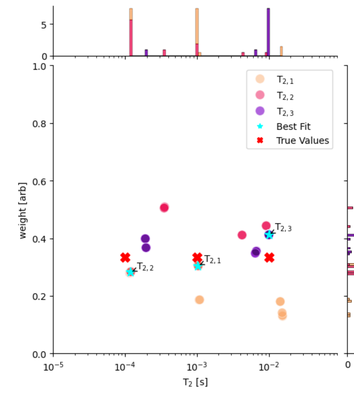


Figure 5. MCMC results for simulated tri-exponential data.

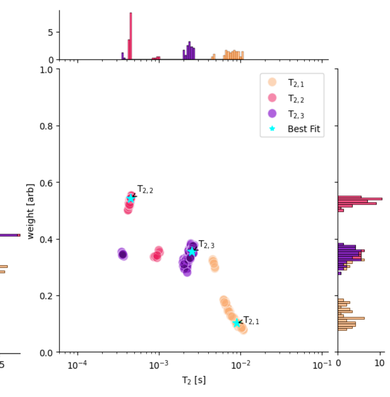


Figure 6. MCMC results for pottery shard data. T_2s were calculated to be 0.45 ms, 2.5 ms, and 8.9 ms with weights of 54%, 36%, and 10%, respectively.

ILT

An ILT was performed on simulated and pottery shard data. The results show good agreement between the top down and bottom-up methods. The smoothing factor used was 1e8.

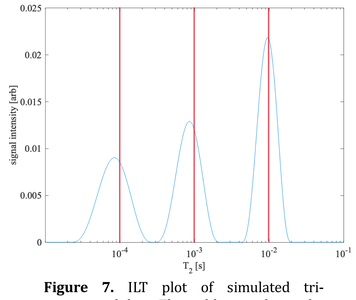


Figure 7. ILT plot of simulated tri-exponential data. The red lines indicate the true T_2 values.

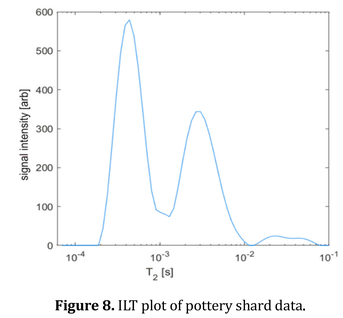


Figure 8. ILT plot of pottery shard data.

Conclusion

Our top-down and bottom-up approaches look promising with simulated data, but for real data, there is not yet complete agreement with all of our methods. We are unsure if this a question regarding SNR, T_2 separation, or our algorithms, especially since we do not know the true T_2s for real samples. The bottom-up approach is able to resolve T_2s that are closer together compared to existing methods and can provide numerical values for each T_2, its weight, and uncertainty. On the other hand, the top-down approach can be used as another method to quickly quantify T_2s in a sample. To improve our results, we are looking into refining our current algorithms while adding more algorithms, like the Cuckoo grid search. We hope to apply these findings to understand other decay-based systems in the future.

References

1. Berman, P. (**2013**) "Laplace Inversion of Low-Resolution NMR Relaxometry Data Using Sparse Representation Methods"
2. Fricke, S.N. (**2020**) "Data processing in NMR relaxometry using the matrix pencil"
3. Vasiliu, D. (**2017**) "Penalized Euclidian distance regression"
4. Roberts, C. P. (**2016**) "The Metropolis-Hastings algorithm"

Background

Top-down approach: Sparsity Matrix

The top-down approach is a variable selection problem that relies on a design matrix, X. Our algorithm finds optimal solutions of β to minimize the error of our loss function. The solution should take the form of a sparsity pattern with larger values contributing more weight to our overall signal.

Sparsity Matrix Algorithm

Our signal function can be redefined in terms of the equation below.

$$S(t) = X \cdot \beta + noise$$

Where

$$X = \begin{bmatrix} e^{-\eta t_0} & e^{-2\eta t_0} & e^{-3\eta t_0} & \cdots & e^{-k\eta t_0} \\ e^{-\eta t_1} & e^{-2\eta t_1} & e^{-3\eta t_1} & \cdots & e^{-kt_1} \\ e^{-\eta t_2} & e^{-2\eta t_2} & e^{-3\eta t_2} & \cdots & e^{-kt_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e^{-\eta t_p} & e^{-2\eta t_p} & e^{-3\eta t_p} & \cdots & e^{-k\eta t_p} \end{bmatrix}, \quad \beta = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_k \end{bmatrix}, \quad and\ noise\ \triangleq \sigma * \epsilon$$

We are seeking a solution β, such that

$$\underset{\beta}{\operatorname{argmin}}\ [L(s, X, \beta) + P(\beta)]$$

Where L is the loss function, the square root of the sum of square errors, $\sqrt{SSE}$. $P(\beta)$ is the Penalized Euclidian Distance (PED) regularization term.

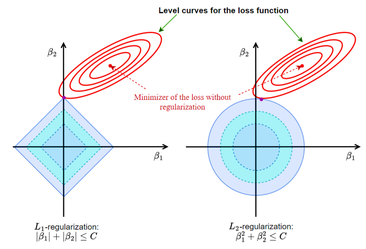
Using $\sqrt{SSE}$ allows our loss function to be independent of σ:

$$\sqrt{SSE} = \|y - X \cdot \beta\|_2 = \sqrt{\textstyle\sum_{i=1}^{n}(y_i - X \cdot \beta)^2}$$
$$y = X \cdot \beta + \sigma * \epsilon$$

So

$$\nabla Loss = \frac{1}{2} \frac{-2(y - X \cdot \beta)\, X}{\sqrt{(y - X \cdot \beta)^2}} \rightarrow independent\ of\ \sigma$$

PED's regularization uses the geometric mean between the L1 and L2 norm, which allows it to capture a combination of both norms. This results in both grouping effects and sparsity patterns, allowing efficient searches for unique solutions.



Bottom-up Approach: MCMC

Markov Chain Monte Carlo (MCMC) is a sampling algorithm that uses Markov Chains to perform Monte Carlo estimations. We utilize MCMC's Bayesian properties to define prior probabilities and use the algorithm to minimize the error between a proposed fit and the real data. Our MCMC implementation uses the Metropolis-Hasting algorithm.

Metropolis-Hastings Algorithm

1. Generate $x^* \mid x_n \sim Normal(x_n, \sigma^2)$
2. Take

$$x_{n+1} = \begin{cases} x^* & if\ u \leq A(x_n \rightarrow x^*) \\ x_n & otherwise \end{cases}$$

where

$$A(x_n \rightarrow x^*) = \min\left\{\frac{p(x^*)}{p(x^n)}\frac{q(x_n \mid x^*)}{q(x^* \mid x_n)}, 1\right\}.$$

A is the acceptance probability, u is a uniformly distributed random number between 0 and 1, $p(x)$ is the probability density function, and q(x) is proposal distribution.

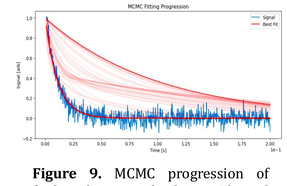


Figure 9. MCMC progression of finding the optimal solution. The red lines represent the best fit.

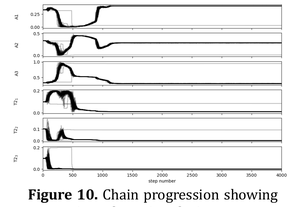


Figure 10. Chain progression showing parameter values at each step.