PARAMETER ESTIMATION WITH MOCK ALGORITHM

{ BOWEN DENG } DEPARTMENT OF STATISTICS

DATA OVERVIEW

The dataset we use is **Wine Quality Data Set** from UCI Machine Learning Data Repository. It has 4898 examples, with 11 features (10 predictors and 1 response) for each example. The response is quality of the wine example.

Feature	Min	Max	Mean	Eg.
Fixed acidity	3.8	14.2	6.9	7
Volatile acidity	0.1	1.1	0.3	0.27
Citric acid	0.0	1.7	0.3	0.36
Residual sugar	0.6	65.8	6.4	20.7
Free sulfur	2	289	35	45
Total sulfur	9	44	138	170
Density	0.99	1.04	0.99	1.001
pH	2.7	3.8	3.1	3
Sulphates	0.2	1.1	0.5	0.45
Alcohol	8.0	14.2	10.4	8.8
Quality	3	9	3.19	6

Table 1: Summary and example for the dataset

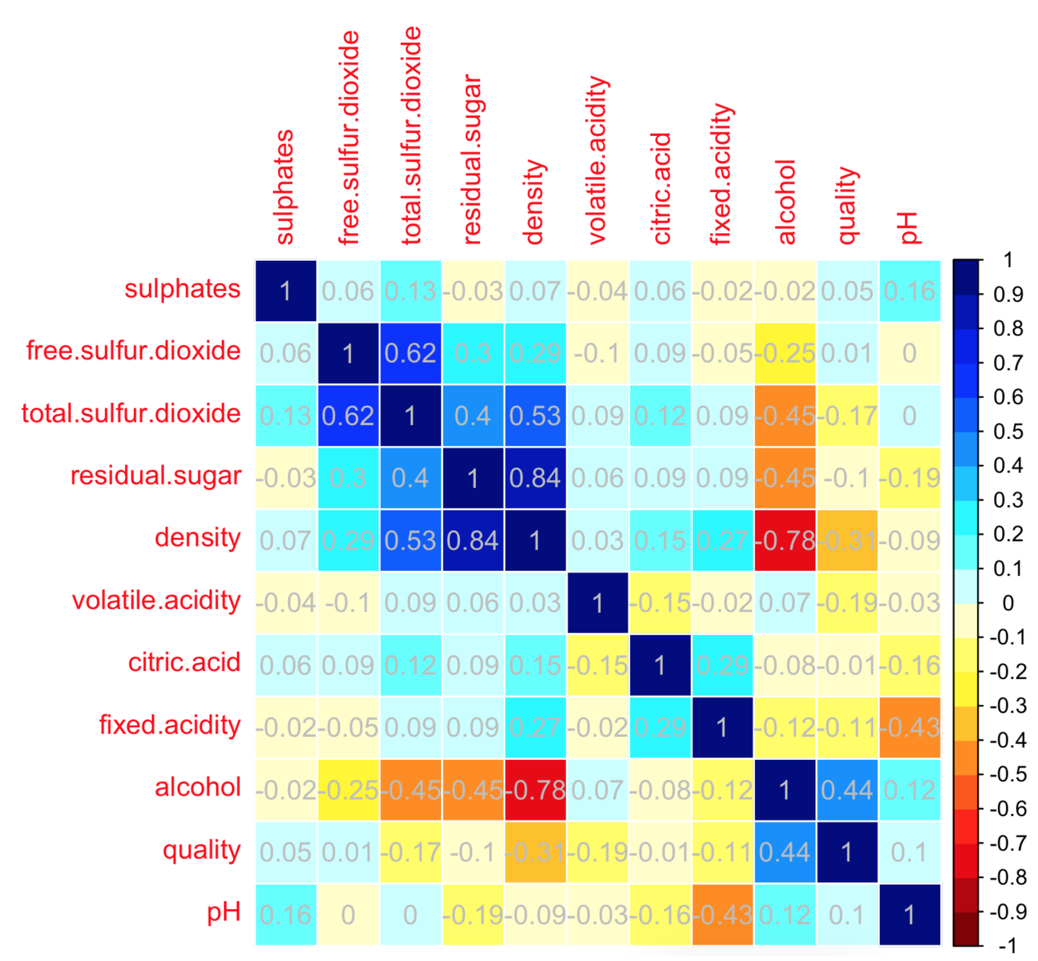


Figure 1: Heatmap for correlation matrix of wine quality data

ALGORITHM

We use linear model for the prediction for this dataset.
Instead of minimizing the mean square error loss, we use the **MOCK** algorithm:
Take Θ' a subset of Θ, and a kernel function $K_\tau : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$.

Algorithm 1 MOnte Carlo Kernel algorithm

for $j = 1, \cdots, B$ **do**
  Sample $\theta(j) \sim \text{Unif}(\Theta')$
  for $i = 1, \cdots, m$ **do**
    $z_i \leftarrow f_{\theta(j)}(x^{(i)})$
  end for
  $w(j) \leftarrow K_\tau(\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}, \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix})$
end for
$\hat{\theta} = \frac{\sum_{j=1}^{B} w(j)\theta(j)}{\sum_{j=1}^{B} w(j)}$

For linear model, which we will be using for the regression on wine quality, the algorithm will have the following degenerations.

$$f_{\theta(j)}(x^{(i)}) = \theta(j)^T x^{(i)}$$
$$K_\tau(y, z) = \exp\left(-\|y - z\|^2/\tau\right)$$

Theorem 1. *Assume $\beta \in [-M/2, M/2]^n$, for each $x^{(i)}, y^{(i)}$ sampled from $N(\beta^T x^{(i)}, \sigma^2)$. And we use a symmetric kernel function $K_\tau(y, z) = K(\frac{y-z}{\tau})$ satisfying*

$$\int K(x) < \infty$$

then as $M, m \to \infty$,

$$\hat{\beta} \to \beta$$

In theory, the **MOCK** estimator *should* converge to the "optimal" value.

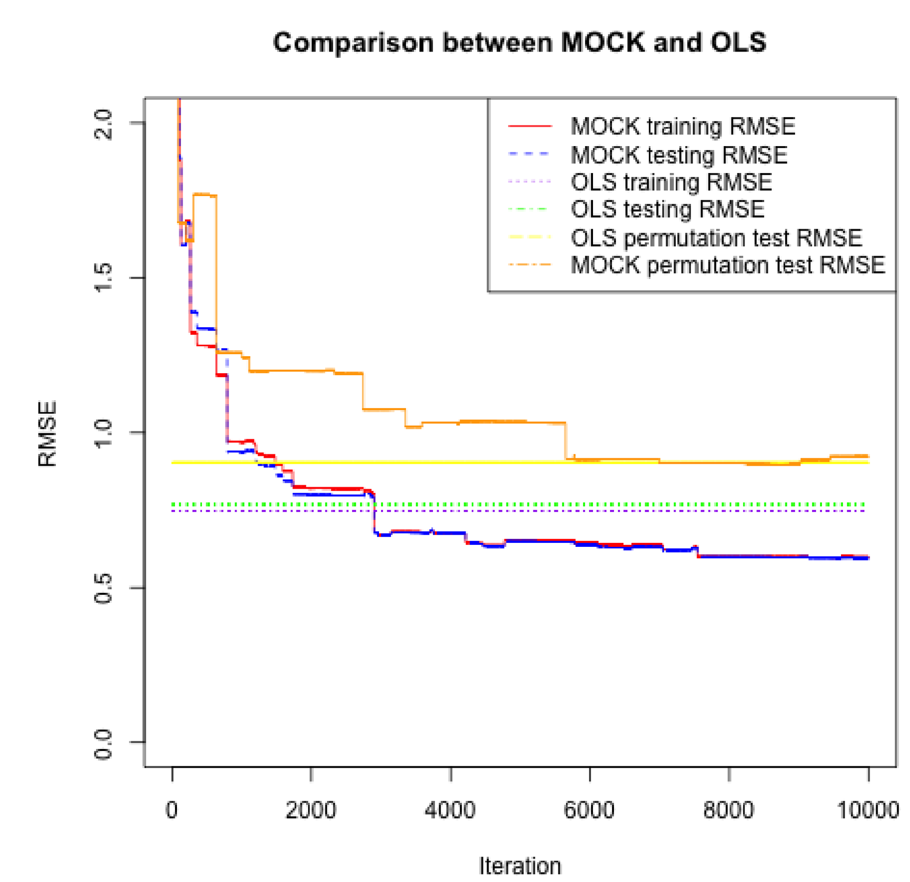
RESULTS

Set $M = 2.0$, $\tau = 3918$. We implement the **MOCK** algorithm in R.
The β we obtained by **MOCK** algorithm is:

Intercept	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	
5.89	-0.03	0.01	-0.06	0.02	
free.sulfur	total.sulfur	density	pH	suphates	alcohol
-0.01	-0.00	0.14	0.12	0.00	-0.02

Table 2: β obtained by **MOCK** algorithm with $\tau = 3918$, $M = 2.0$.

We compare the **RMSE** and runtime of **MOCK** with those of **OLS**.



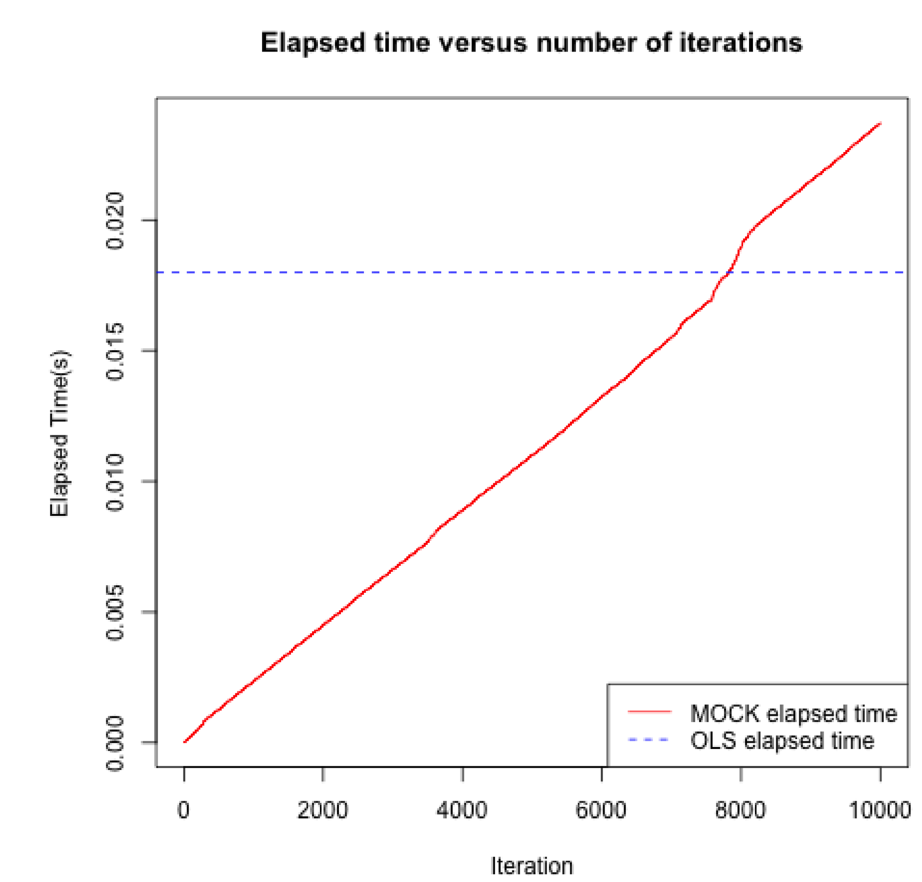


Figure 2: Comparison of **RMSE** (left) and runtime (right) for textbfMOCK and **OLS**

There is a clear tradeoff between the accuracy (**RMSE**) and the runtime. However, 6000 is a sweet spot where accuracy is almost converged and it's 18% lower than that of **OLS**, and the runtime is 25% smaller than that of **OLS**.
By looking at the error for perturbed dataset, the **RMSE** obtained by both **MOCK** and **OLS** are significantly lower than that of perturbed dataset. Hence, we can safely draw the conclusion that both algorithms are capturing the signal rather than noise.
In practice, the **MOCK** estimator performs no worse than **OLS** with appropriate parameter.