Cockrell School of Engineering

Capacitance Resistance Model (CRM) as a Scikit-Learn Estimator

Akhil Potla & John T. Foster

DIRECT

06/23/21

Executive Summary

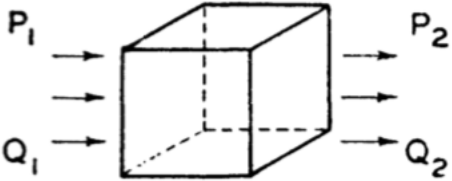
- We are implementing CRM into a Scikit-Learn estimator to easily compare various machine learning estimators and take advantage of cross validation features
- The goal is to CRM and machine learning technique to predict the oil production rate and to optimize injection rate

Motivation

- Predict the oil production rate using Capacitance-Resistance Model - Producer (CRMP)
- CRMP implemented as a Scikit-Learn Estimator
  - Common API
  - Use different cross validation tools provided by Scikit-Learn to determine the optimal parameters for CRMP

Origins of CRM

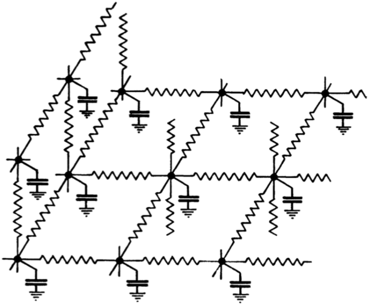
- First Proposed in 1943
- Reservoir was divided into "grid blocks"



WA Bruce et al. An electrical device for analyzing oil-reservoir behavior. In: *Transactions of the AIME* 151.01 (1943), pp. 112–124.

Origins of CRM

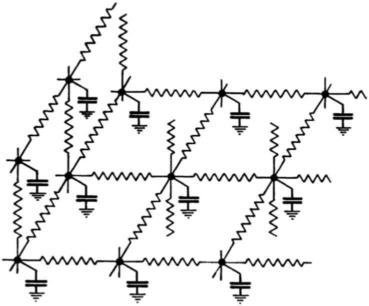
- Capacitors represent reservoir unit storage capacity
- Resistors represent inverse transmissibility



Bruce et al. 1943.

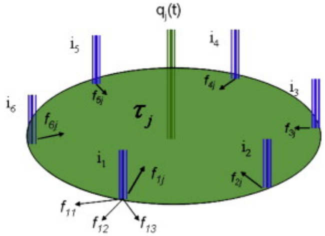
Origins of CRM

- Capacitors represent reservoir unit storage capacity
- Resistors represent inverse transmissibility



Bruce et al. 1943.

CRM Producer Volume (CRMP)



Morteza Sayarpour, Elizabeth Zuluaga, C Shah Kabir, and Larry W Lake. The use of capacitance–resistance models for rapid estimation of waterflood performance and optimization. In: *Journal of Petroleum Science and Engineering* 69.3-4 (2009), pp. 227–238.

CRMP Assumptions

- Constant temperature
- Slightly compressible fluids
- No formation gas
- Negligible capillary pressure effects
- Constant volume

Sayarpour, Zuluaga, Kabir, and Lake 2009.

Time Constant

$$\tau = \frac{c_t V_p}{J}$$

- τ, time constant $[\mathrm{days}]$
- c_t, total compressibility $\left[\mathrm{psi}^{-1}\right]$
- V_p, pore volume $[bbls]$
- J, productivity index $\left[\frac{\mathrm{bbl}}{\mathrm{day}\cdot\mathrm{psi}}\right]$

Gains

$$f_{ij} = \frac{T_{ij}}{\sum_{j=1}^{N_{prod}} T_{ij}}$$

- T_{ij}, Transmissibility $\left[\frac{\mathrm{bbl}}{\mathrm{day}\cdot\mathrm{psi}}\right]$

Sayarpour, Zuluaga, Kabir, and Lake 2009.

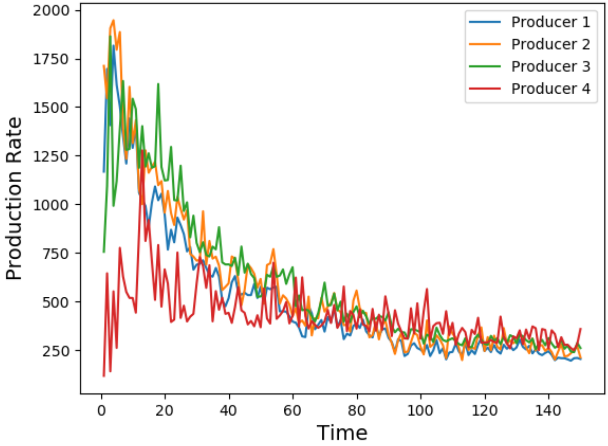
CRMP Production Rate

$$q_j^k = q_j^{k-1} exp(\frac{-\Delta t^k}{\tau}) + (1 + exp(\frac{-\Delta t^k}{\tau})) \sum_{i=1}^{N} I_i f_{ij}$$

- q_j^k, production rate at producer j at time step k $\left[\frac{\text{bbls}}{\text{day}}\right]$
- Δt^k, time that elapses during time step k $[\text{days}]$
- I_i, injection rate $\left[\frac{\text{bbls}}{\text{day}}\right]$

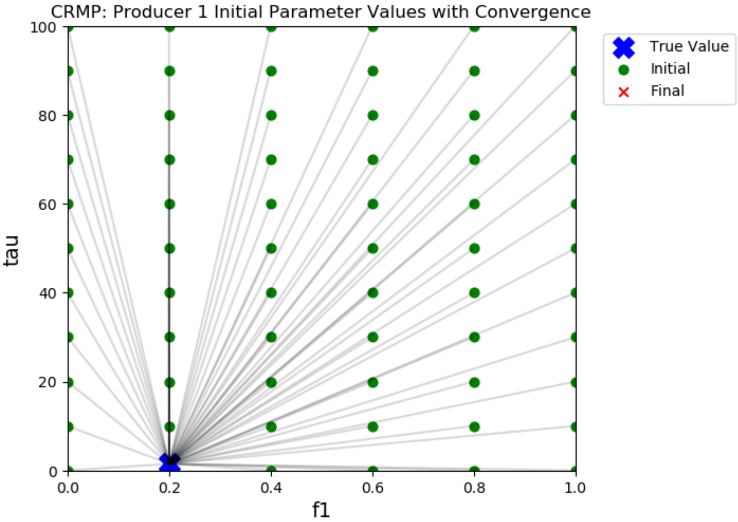
Sayarpour, Zuluaga, Kabir, and Lake 2009.

Production Rate of Our Wells

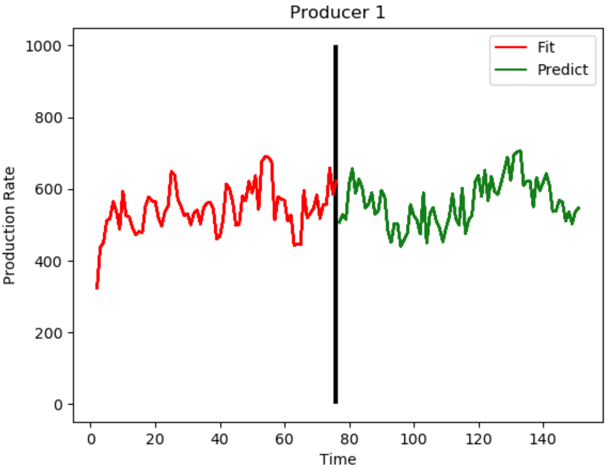


The University of Texas at Austin
Cockrell School of Engineering

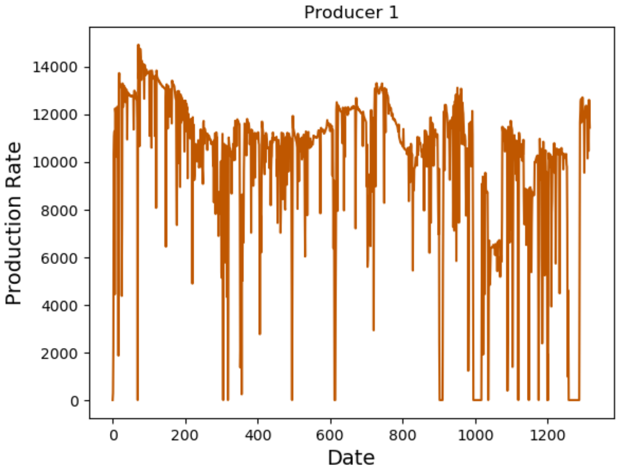
Producer 1: Convergence Across the Parameter Space



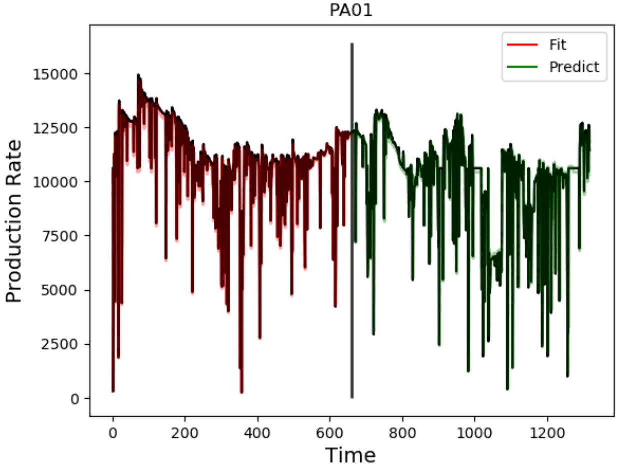
Producer 1: Convergence Across the Parameter Space



Real Data: Production Rate vs Time



Real Data: Production Rate vs Time

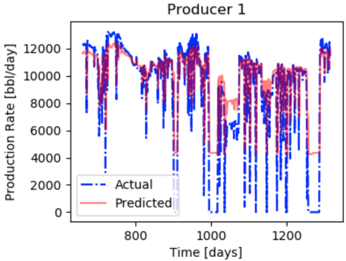


Usage

```
In [16]: # Linear Regression
         from sklearn.linear_model import LinearRegression
         model = LinearRegression().fit(X_train, y_train)
         print(model.coef_)
         y_hat = model.predict(X_test)

         [ 0.60719089 -0.00369698  0.          0.          0.        ]

Out[16]: <matplotlib.legend.Legend at 0x7fdaaf1458e0>
```

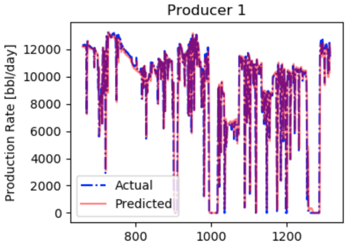


Usage

In [18]:
```python
# CRMP
from crmp import CRMP
crmp = CRMP().fit(X_train, y_train)
print('Tau: ', crmp.tau_)
print('Gains: ', crmp.gains_)
y_hat = crmp.predict(X_test)
```

```
Tau:  51.512053747256836
Gains:  [0.32532923 0.18527281 0.18527281 0.18527281]
```

Out[18]:
```
<matplotlib.legend.Legend at 0x7fdaae7cab20>
```



Future Work

- Analyzing the oil cut
  - Koval Model
  - ML Estimators
- Investigating uncertainty

DIRECT Sponsors

BAZEAN

OXY

ExxonMobil

bp

aramco

JOGMEC

JAPEX

The University of Texas at Austin
Cockrell School of Engineering