Applying Machine Learning Algorithms to Oil Reservoir Production Optimization

Mehrdad Gharib Shirangi
Stanford University

Abstract

In well control optimization for an oil reservoir described by a set of geological models, the expectation of net present value (NPV) is optimized. This approach called robust optimization, entails running the reservoir simulator for all the reservoir models at each iteration of the optimization algorithm. Hence, robust optimization can be computationally demanding. One way to reduce the computational burden, is to select a subset of models and perform the optimization on the reduced set. Another popular technique, is using fast proxy models, rather than full-physics simulators. In this work, a kernel clustering technique is used to select a subset of reservoir models that capture the range of uncertainty in the response of the entire set.

In this work an adaptive strategy is used to build fast proxy models for the NPV, and then optimizing the proxy model using a pattern search algorithm. The proxy model is generated by training an artificial neural network (ANN) or support vector regression model (SVR) using some training examples. The challenge in building a proxy model is using finding good training examples. In this work, after optimizing the proxy model, new training examples are generated around the current optimal point, and a new proxy model is built and the procedure is repeated. An example is presented that shows the efficiency of kernel k-medoids clustering and building proxy models for production optimization.

Introduction

Wang et al. (2012) applied retrospective optimization to well placement problem. They applied k-means clustering for selecting the realizations at each subproblem. The focus of this work is on well control optimization. A response vector is introduced to well characterizes the dissimilarity between the response of realizations. Here, kernel k-medoids clustering is applied for choosing a small set of statistically representative realizations.

Our objective is to find an optimal well control vector that maximizes the expectation of the life-cycle NPV over the ensemble of reservoir models.

The well control vector u, can be shown by

$$u = \left[u_1^1, u_1^2, \cdots, u_1^{N_c}, \cdots, u_{N_w}^1, u_{N_w}^2, \cdots, u_{N_w}^{N_c}\right]^T, \tag{1}$$

where, each subscript denotes the well index and each superscript denotes the control index; N_w denotes the number of wells and N_c denotes the number of control steps for each well. u_j^n can be either well pressure (BHP) or oil rate or total liquid rate of the jth well at the nth simulation time step. In this work, the well controls are the BHP's. We only consider simple bound constraints.

In robust production optimization, we want to maximize $E[J(u,m)]$, where

$$E[J(u,m)] = \frac{1}{N_e}\sum_{j=1}^{N_e} J(u, m_j), \tag{2}$$

where m denotes a realization of the reservoir model, and $J(u,m_j)$ denotes the corresponding NPV.

Model Selection for Reducing Number of Realizations

A simple way to reduce the computational cost, is to select a few statistically representative models in the ensemble; these representative models are selected based on ranking or clustering some attributes. A random selection of the realizations will not provide a representative set for the ensemble.

Wang et al. (2012) used k-means clustering for selecting a set of representative realizations from a large ensemble of models in field development optimization. The attributes they used for clustering are OOIP, normalized cumulative oil production, permeability distance, and oil-water contact depth. They applied k-means clustering to obtain a number of clusters, and then one realization was chosen randomly from each cluster. Scheidt and Caers (2009) developed an effective model selection method using kernel k-means clustering in metric space. In this method, after obtaining the clusters, the model nearest to the centroid of that cluster (the medoid) is chosen.

In this work, kernel k-medoids clustering is used, where the models are divided into some clusters and the robust production optimization is applied on the reduced ensemble which contains the models corresponding to the clusters centroids. As the center of each cluster represents all the models in that cluster, a weight is assigned to each centroid. Finally, the weighted expectation of NPV of the centroids of the clusters is maximized:

$$\text{maximize } \frac{1}{N_e}\sum_{j=1}^{n_k} \omega_j J(u, m_j, y_j), \tag{3}$$

where n_k denotes the number of clusters. With N_j representing number of realizations in the jth cluster, $\omega_j = N_j$ is set as the weight of the jth centroid in the objective function of the subproblem.

In this work, the field cumulative oil and water production and the field cumulative water injection in time are used to construct a response vector for each realization.

To obtain the response vectors, all realizations are ran with the simulator using the same control strategy. At this stage, as we are only interested in calculating a dissimilarity distance, the responses can be obtained from a rank-preserving proxy-type simulator, e.g. the streamline simulator, and need not be the exact simulator.

Building Fast Proxy Models

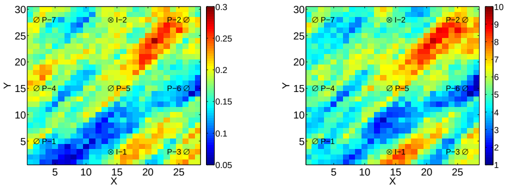
In this work, we use ANN and SVR to build fast proxy models. One challenge in building a fast proxy, is finding good training examples. Note that the input space (well controls) is very high-dimensional and continuous, hence, a randomly generation of a few training examples may not be very efficient.

Support Vector Regression as a Proxy

SVR is a regression model that can be trained to make predictions of output of a function for a given vector of input features. The general idea of SVR is to perform linear regression in a high dimensional feature space by using kernel trick (B. Scholkopf and Bartlett, 2005). In this work we use a SVR software (Chang and Lin, 2011). It is worthwhile to mention that for using the SVR, all the input features should be scaled to be between 0 and 1; same is about the output value. Hence, here all the well controls (input features) are scaled to be in $[0, 1]$, while the output values (-NPV), are also scaled to be in $[0, 1]$. In ϵ-SVR, there are two important parameters that affect the efficiency of the algorithm, which are ϵ and C. In addition, the number of training examples used also affects the predictions.

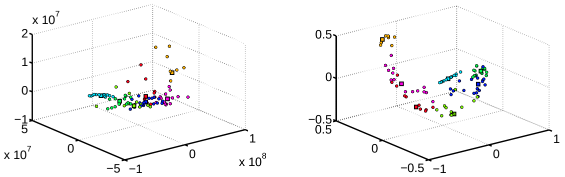
Artificial Neural Network as a Proxy

Artificial neural network (ANN) is a popular machine learning algorithm inspired by biological neural network. A neural network consists of input nodes, output nodes and interconnected groups of artificial neurons. In this work, we use the neural network toolbox of MATLAB. The idea is to build a ANN proxy of some training examples, and then optimize the ANN model rather than the computationally demanding full-physics model.



(a) true porosity field (b) true ln(k) field

Figure 1: True porosity and log-permeability fields.



(a) original space (b) kernel space

Figure 2: MDS plot of 100 realizations with 7 clusters and their centroids.

Example

This example pertains to a two-dimensional horizontal reservoir model with 28 × 30 uniform grid. True porosity and log permeability fields with well locations are shown in Fig. 1.

There are 7 producers and 2 injectors in this reservoir. $N_e = 100$ realizations of both porosity and log-permeability are generated using geostatistics and then conditioned to 300 days of production history using truncated SVD parameterization (Gharib Shirangi, 2011). Production optimization is performed for the time interval of 300 to 3000 days.

The objective is first to reduce the number of realizations in robust optimization to significantly reduce the computational costs while obtaining nearly the same final expected NPV and characterization of uncertainty in the expected NPV. Second objective is to test the efficiency of using SVR and ANN in building fast proxy models in production optimization.

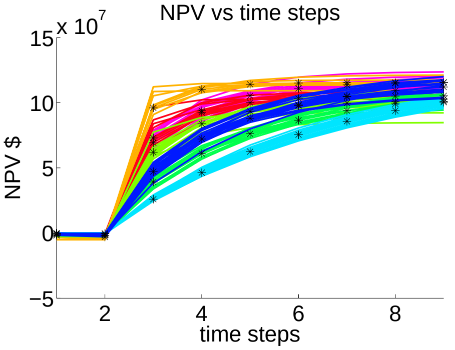


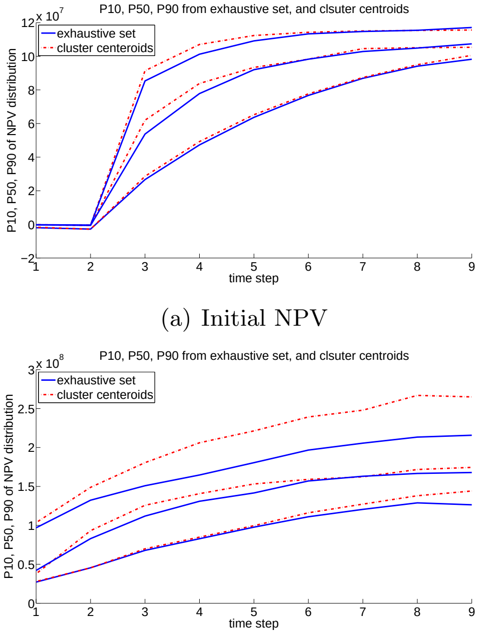
Figure 3: NPV values in time for 100 realizations colored for different clusters, markers show the curves of the cluster centroids.

Kernel Clustering

Fig. 2 shows the MDS plot and the points for each of the 7 clusters. In uncertainty assessment of production optimization, capturing P10, P50 and P90 of the uncertain NPV distribution is very important. As one can see in Fig. 4(a), we could achieve this goal by using only 7 clusters. Fig. 4(b) shows the P10, P50 and P90 of the optimal NPV obtained by maximizing the expectation of NPV of cluster centroids. The optimization was performed by first generating an ANN model and then optimizing the ANN model using pattern search algorithm (will be discussed later). We ran all the models with the optimal solution, and plotted the corresponding P10, P50 and P90. As one can see, P10 and P50 values for the exhaustive set at the final time are in good agreement with those from the cluster centroids. This is of significant importance. Fig. 3 shows the unoptimized NPV (from initial guess) of all realizations versus time, colored based on clusters.

Building Fast Proxy Models

We first perform a sensitivity study on parameters of SVR (C and ϵ). To do so, we generate 700 training examples, by randomly perturbing the control vector around the mean. Out of these training examples, we set the last 200 points as testing set. Fig 5 shows the sensitivity of SVR to parameters C, ϵ, and number of training examples. As can be seen, SVR has small testing error even for small number of training examples. In addition, a larger value of C and a smaller ϵ provides smaller testing error.



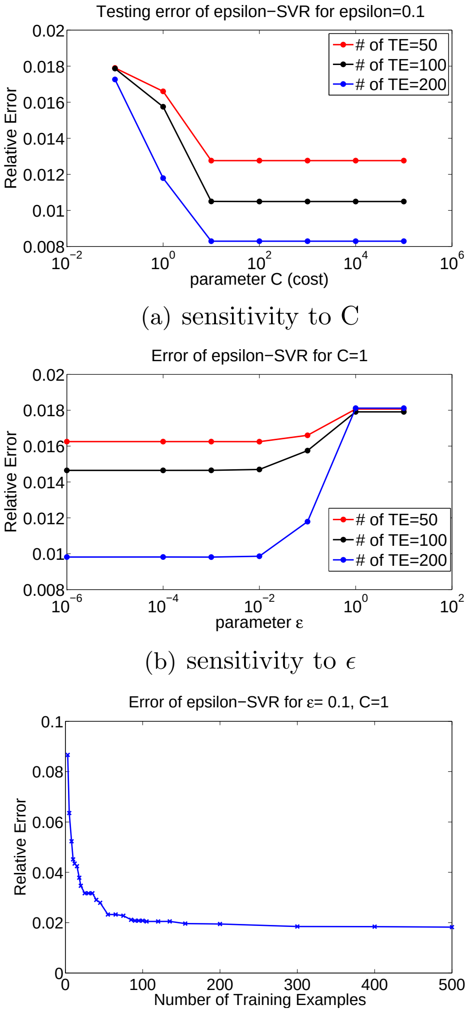
(a) Initial NPV

(b) Optimized NPV

Figure 4: P10, P50 and P90 of all 100 realizations and those calculated using only cluster centroids for the initial NPV, and optimal NPV.

Next, we build a proxy model from a number of training examples. Then we optimize the SVR model using pattern search (PS) optimization algorithm. Depending on how many training examples we use in generating the SVR model, the solution from optimizing SVR can be different. Fig. 6 shows how the optimal NPV changes by increasing the number of training examples. Fig. 7 shows the results of directly optimizing NPV using pattern search algorithm. While directly optimizing NPV takes about 800 function evaluations, we could obtain the same optimal NPV by optimizing an SVR model trained with about 100 function evaluations. Hence SVR can provide significant computational saving in production optimization.

Next we use ANN as a proxy. We use 1 hidden layer with 10 artificial neurons. Increasing the number of hidden layers or neurons didnot have a significant effect on the efficiency of ANN model. Here, we use an adaptive strategy for generating training examples. We first gener-



(a) sensitivity to C

(b) sensitivity to ϵ

(c) number of training examples

Figure 5: Sensitivity of SVR (testing error) with respect to parameters.

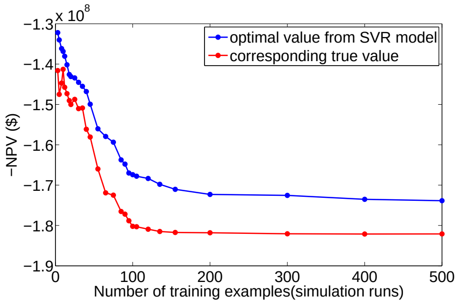


Figure 6: Change of optimal NPV from optimizing SVR with number of training examples.

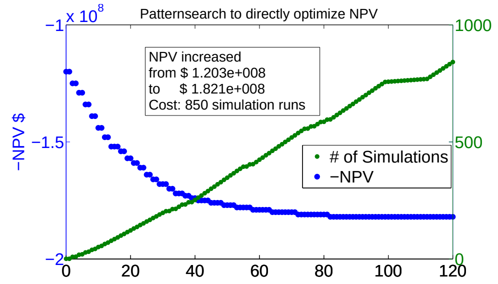


Figure 7: Directly optimizing NPV using pattern search algorithm (without a proxy model). Left axis shows -NPV while right axis shows the number of function evaluations

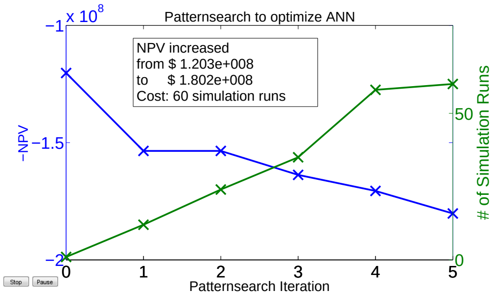


Figure 8: Optimizing ANN proxy of NPV using pattern search algorithm. Left axis shows -NPV while right axis shows the number of function evaluations

ate a few training examples, and we optimize the ANN model. Having obtained the optimal point, we run the simulator with the new point, and generate new training examples around the current point either by randomly perturbing the current point or by polling in different directions around the current point. Fig. 8 shows the results of using ANN as a proxy. According to the results of this figure, with only 60 function evaluations, the NPV increased to about the value obtained by directly optimizing NPV.

Conclusions

In this work, unsupervised learning algorithm is applied to the robust production optimization problem to choose a set of representative realizations. Kernel clustering technique applied to NPV curves, provided a set of representative models that capture the uncertainty in the response of the entire set of models. At least for the examples here, we showed that optimizing the mean NPV of only cluster centroids, would optimize the P10 and P50 of the entire set of models, and the P10 and P50 of the cluster centroids at optimal solution are in close agreement with those of the exhaustive set.

Water flood optimization under geological uncertainty, even over a few representative realizations, can be computationally expensive. In this project, support vector regression and artificial neural network was applied to build proxy models to use in production optimization. Both SVR and ANN provided significant computational savings, compared to optimizing without using a proxy model.

References

B. Scholkopf, R. C. W., A. J. Smola and P. L. Bartlett, 2005, New support vector algorithms, *Neural Computation*, **12**, 1207–1245.

Chang, C.-C. and C.-J. Lin, 2011, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, **2**, 27:1–27:27.

Gharib Shirangi, M., 2011, *History Matching Production Data With Truncated SVD Parameterization*, Master's thesis, The University of Tulsa.

Scheidt, C. and J. Caers, 2009, Representing spatial uncertainty using distances and kernels, *Mathematical Geosciences*, **41**(4), 397–419.

Wang, H., D. Echeverra-Ciaurri, L. Durlofsky, and A. Cominelli, 2012, Optimal well placement under uncertainty using a retrospective optimization framework, *SPE Journal*, **17**(1), 112–121.