Experimental Design in Dynamical System Identification: A Bandit-Based Active Learning Approach

Artémis Llamosi, Adel Mezine, Florence D'Alché-Buc, Véronique Letort, Michèle Sebag

Experimental Design in Dynamical System Identification: A Bandit-Based Active Learning Approach

Artémis Llamosi[1,4], Adel Mezine[1], Florence d'Alché-Buc[1,3], Véronique Letort[2], and Michèle Sebag[3]

[1] Informatique Biologie Intégrative et Systèmes Complexes (IBISC), Université d'Evry-Val d'Essonne, France
artemis.llamosi@univ-paris-diderot.fr
{amezine,florence.dalche@ibisc.univ-evry.fr}
[2] Ecole Centrale Paris, 92295 Châtenay-Malabry Cedex
veronique.letort@ecp.fr
[3] TAO, INRIA Saclay
Laboratoire de Recherche en Informatique (LRI), CNRS, Université Paris Sud, Orsay, France
Michele.Sebag@lri.fr
[4] Laboratoire Matière et Systèmes Complexes, Université Paris Diderot & CNRS, 75013 Paris, France
INRIA Paris-Rocquencourt, Rocquencourt, 78153 Le Chesnay, France

Abstract. This study focuses on dynamical system identification, with the reverse modeling of a gene regulatory network as motivating application. An active learning approach is used to iteratively select the most informative experiments needed to improve the parameters and hidden variables estimates in a dynamical model given a budget for experiments. The design of experiments under these budgeted resources is formalized in terms of sequential optimization. A local optimization criterion (reward) is designed to assess each experiment in the sequence, and the global optimization of the sequence is tackled in a game-inspired setting, within the Upper Confidence Tree framework combining Monte-Carlo tree-search and multi-armed bandits.

The approach, called EDEN for Experimental Design for parameter Estimation in a Network, shows very good performances on several realistic simulated problems of gene regulatory network reverse-modeling, inspired from the international challenge DREAM7.

Keywords: Active learning, experimental design, parameter estimation, Monte-Carlo tree search, Upper Confidence Tree, ordinary differential equations, e-science, gene regulatory network.

1 Introduction

A rising application field of Machine Learning, e-science is concerned with modeling phenomena in e.g. biology, chemistry, physics or economics. The main goals

of e-science include the prediction, the control and/or the better understanding of the phenomenon under study. While black-box models can achieve prediction and control goals, models consistent with the domain knowledge are most desirable in some cases, particularly so in domains where data is scarce and/or expensive.

This paper focuses on the identification of dynamical systems from data, with gene regulatory network reverse modeling as motivating application [26]. We chose the framework of parametric ordinary differential equations (ODE) [18,11] whose definition is based on the domain knowledge of the studied field. Our goal is restricted to *parametric identification*. Formally, it is assumed that the structure of the ODE model is known; the modeling task thus boils down to finding its m-dimensional parameter vector $\boldsymbol{\theta}$. Setting the ODE parameter values, also referred to as reverse-modeling, proceeds by solving an optimization problem on $\mathbb{R}^m$, with two interdependent subtasks. The first one is to define the target optimization criterion; the second one is to define the experimental setting, providing evidence involved in the optimization process.

Regarding the first subtask, it must be noticed that parametric ODE identification faces several difficulties: i) the behavior described by the ODE model is hardly available in closed form when the ODE is nonlinear and numerical integration is required to identify the parameters, ii) the experimental evidence is noisy, iii) the data is scarce due to the high costs of experiments, iv) in some cases the phenomenon is partially observed and therefore depends on hidden state variables. To overcome at least partially these difficulties, several estimation methods have been employed using either frequentist [11] or Bayesian inference [27]. In case of hidden variables, Expectation-Maximization approaches and filtering approaches with variants devoted to nonlinear systems such as the Unscented Kalman Filter (UKF) [31] and the Extended Kalman Filter [37] have been applied to ODE estimation.

Overall, the main bottleneck for parameter estimation in complex dynamical systems is the non-identifiability issue, when different parameter vectors θ might lead to the same response under some experimental stimuli[1]. The non-identifiability issue is even more critical when models involving a high-dimensional parameter vector $\boldsymbol{\theta}$ must be estimated using limited evidence, which is a very common situation. To mitigate the non-identifiability of parameters and hidden states in practice, the e-scientist runs complementary experiments and gets additional observations. Ideally, these observations show some new aspects of the dynamical behavior (e.g. the *knock-out* of a gene in an organism), thereby breaking the non-identifiability of parameters. The selection of such (expensive) complementary experiments is referred to as *design of experiments (DOE)*. The point is to define the optimal experiments in the sense of some utility function, usually measuring the uncertainties on $\boldsymbol{\theta}$ (including the non-identiabilities), and depending on the experiment, the data observed from it and the quality of estimates produced by some chosen estimation procedure. For instance, the utility

[1] See [32] for a presentation of non-identifiability issues, beyond the scope of this paper.

function can refer to the (trace of) the covariance matrix of the parameter estimate. DOE has been thoroughly studied from a statistical point of view for various parameter estimation problems (see [36,16,25]), within a frequentist or a Bayesian framework [10,28]. Usual definitions of utility include functions of Fisher information in the frequentist case [35] or of the variance of the estimated posterior distribution in the Bayesian case [5]. In this work, we focus on the case of sequential experimental design [33,6], which is the most realistic situation for experimentations in a wet laboratory.

The limitations of current standard sequential DOE is twofold. Firstly, it seldom accounts for the cost of the experiments and the limited budget constraint on the overall experiment campaign. Secondly and most importantly, it proceeds along a myopic strategy, iteratively selecting the most informative experiment until the budget is exhausted.

The contribution of the present paper is to address both above limitations, formulating DOE as an *active learning* problem. Active learning [12,13,14,3] allows the learner to ask for data that can be useful to improve its performance on the task at hand. In this work, we consider active learning as a *one-player game* similarly to the work of [34] devoted to supervised learning and propose a strategy to determine an optimal set of experiments complying with the limited budget constraint. The proposed approach is inspired by the Upper Confidence Tree (UCT) [24,34], combining Monte-Carlo tree search (MCTS) [7] and multi-armed bandits [9]. Formally, a reward function measuring the utility of a set of experiments is designed, and UCT is extended to yield the optimal set of experiments (in the sense of the defined reward function) aimed at the estimation of parameters and hidden variables in a multivariate dynamical system.

The approach is suitable for any problem of parameter estimation in ODE where various experiments can be defined: those experiments can correspond to the choice of the sampling time of observation, the initial condition in which the system is primary observed or some intervention on the system itself. In this work, the approach is illustrated considering the reverse modeling of gene regulatory networks (GRN) in systems biology [11,26]. GRN are dynamical systems able to adapt to various input signals (e.g. hormones, drugs, stress, damage to the cell). GRN identification is a key step toward biomarkers identification [4] and therapeutical targeting [23].

After an introduction of the problem formalization, the paper gives an overview of the proposed approach, based on an original reward function and extending the UCT approach. A proof of concept of the presented approach on three realistic reverse-modeling problems, inspired by the international DREAM7 [15] challenge, is then presented in the application section.

2 Problem Setup

We consider a dynamical system whose state at time t is the d-dimensional vector $\mathbf{x}(t)^T = [x_1(t) \dots x_d(t)]$ and whose dynamics are modeled by the following first-order ODE:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); \boldsymbol{\theta}) \ , \tag{1}$$

where $\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}(t)}{dt}$ denotes the first order derivative of $\mathbf{x}(t)$ with respect to time, function $\mathbf{f}$ is a non linear mapping, $\boldsymbol{\theta}$ is the m-dimensional parameter vector, $\mathbf{u}(t)$ is an exogenous input to the system. Let us first assume that we partially observe its behavior given some initial condition $\mathbf{x}(0) = \mathbf{x}_0$ and with some neutral input $\mathbf{u}(t) = g_0(t)$, e.g. without any *intervention* (as defined below). Let $\mathbf{H}$ be the observation model, typically a projection of $\mathbb{R}^d$ in a lower dimensional space $\mathbb{R}^p$ $(p < d)$, $\mathbf{Y}_0 = (\mathbf{y}^0_{t_k})_{k=0,\ldots,n-1}$, a time series of n p-dimensional observations and $(\boldsymbol{\epsilon}_{t_k})_{k=0,\ldots,n-1}$, n i.i.d realizations of a p-dimensional noise. For sake of simplicity, $\mathbf{y}_{t_k}$ (resp. $\boldsymbol{\epsilon}_{t_k}$) will be noted $\mathbf{y}_k$ (resp. $\boldsymbol{\epsilon}_k$). Given these assumptions, the observations and the states of the system [31] can now be expressed as follows: given $k = 0, \ldots, n-1$:

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0 \\ \mathbf{x}(t_{k+1}) &= \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} f(\mathbf{x}(\tau), \mathbf{u}(\tau), \boldsymbol{\theta}) d\tau \\ \mathbf{y}_k &= \mathbf{H}(\mathbf{x}(t_k), \mathbf{u}(t), \boldsymbol{\theta}) + \boldsymbol{\epsilon}_k \ . \end{aligned} \tag{2}$$

This model can be seen as a special state-space model where the hidden process is deterministic and computed using a numerical integration. Different tools such as nonlinear filtering approaches such as Unscented Kalman Filtering (UKF) [31] and extended Kalman Filtering (EKF) [37] can be applied. However it is well known that nonlinearity and limited amount of data can lead to practical non-identifiability of parameters. Namely, two different parameter solutions can provide the same likelihood value. A well known way to address this issue is to intervene on the dynamical system to perform additional experiments producing observations that exhibit different kinetics. It can consist either in perturbing the system, e.g. forcing the level of a state variable to be zero, or in changing the observation model by allowing to observe different state variables. To benefit from these new data during the estimation phase, the ODE model must account for all the available experiments defined by a finite set of size E: $\mathcal{E} = \mathcal{E}_0 = \{e_1, \ldots, e_E\}$. This can be done by defining adequately the exogenous input $\mathbf{u}(t)$ among a set of intervention functions $\mathbf{g}_e(t), e \in \mathcal{E}$ as shown in the application section.

Choosing the appropriate interventions (experiments) to apply to the system in order to produce better estimates of parameter and hidden states is the purpose of this work. We are especially interested in an *active learning algorithm* that sequentially selects at each step ℓ, the next experiment e^*_ℓ among the candidate experiments of the set $\mathcal{E}_\ell = \mathcal{E}_{\ell-1} - \{e^*_{\ell-1}\}$, that will produce the most useful dataset for the estimation task. Contrary to the purely statistical approaches of experimental design, ours aims at offering the possibility to anticipate on the fact that one given experiment will be followed by others. The search for an optimal $e^*_\ell \in \mathcal{E}_\ell$ thus depends on the potential subsequent sequences of experiments, their total number being limited by a finite affordable *budget* to account for the cost of real experiments.

Algorithm 1. EDEN or *real game*

Initialization (section 3.2)
while (budget not exhausted) and (estimates not accurate) **do**
  Design a new experiment using Upper Confidence Tree (UCT) as in Algorithm 2 (section 3.3)
  Perform the proposed experiment and re-estimate parameters with the augmented dataset (section 3.4)
  Evaluate the estimates (section 3.5)
end while

3 Game-Based Active Learning for DOE

Please note that in the following, to simplify the description of the approach, we will only talk about parameter estimates, implying hidden state and parameter estimates.

3.1 Complete Algorithm

Active learning of parameters and hidden states in differential equations is considered as a one-player game. The goal of the game is to provide the most accurate estimates of parameters and hidden states. Before the game begins, a first estimate of the hidden states and parameters is obtained using an initial dataset (here unperturbed, termed *wild type*). Then, at each turn, the player chooses and buys an experiment and receives the corresponding dataset. This new dataset is incorporated into the previous dataset and parameters are re-estimated. This procedure, described in Algorithm 1, is repeated until the quality of estimates is sufficiently high or the player has exhausted the budget.

3.2 Initialization

At the beginning of the game, the player is given:

- An initial dataset, here a time series $\mathbf{Y}_0 : \{\mathbf{y}_0^0, \ldots, \mathbf{y}_{n-1}^0\}$, corresponding to the partial observation of the wild type system measured at time $t_0, \ldots, t_{n-1}$ with given initial condition.
- A system of parametric ordinary differential equations, $\mathbf{f}$, of the form of Eq. (1) and an observation model $\mathbf{H}$.
- A set of experiments $\mathcal{E} = \mathcal{E}_0 = \{1, \ldots, E\}$ along with their cost (for simplicity and without loss of generality, the cost is assumed in this work to be equal to 1 for all experiments).
- A total budget: $\mathrm{B} \in \mathbb{R}$ (here the total number of experiments we can conduct) and an optimizing horizon T which states how many experiments we optimize jointly at each iteration of Algorithm 1.
- A *version space*, Θ, which represents all the probable parametrization of our system, compatible with the observed initial set. More precisely, it consists

of a candidate set of *hypotheses* $\Theta(\mathbf{Y}_0) = \{\boldsymbol{\theta}_1^{*(0)}, \boldsymbol{\theta}_2^{*(0)}, \ldots, \boldsymbol{\theta}_m^{*(0)}\}$: a parameter vector can be considered as a hypothesis, i.e. included in the version space, if the simulated trajectories of the observed state variables it generates are consistent with the available dataset. The initial version space is built from the means of the posterior distributions of parameters estimated from the initial dataset $\mathbf{Y}_0$. Building up on previous works [31], we learn m Unscented Kalman Filter (UKF), as described in 3.4, starting with m different initializations and flat priors.
- A reward (or utility) function used in the design procedure, described in 3.3.

3.3 Design of Experiment Using Upper Confidence Tree

The ℓ^{th} move of the *real game*, i.e. the EDEN protocol, consists of running a Monte-Carlo Tree Search (MCTS) in order to find the best first experiment to perform given it is followed by $T-1$ experiments (or less if we have a remaining budget that does not allow for $T-1$ experiments).

The utility of a sequence of experiments is inherently a random variable because of the uncertainty on the true system (the true parameter vector $\boldsymbol{\theta}_{true}$ is not known), but also because of the particular realization of the measurement noise (note that for stochastic models, additional uncertainty would come from the process noise). In addition, the utility of a sequence of experiments is not additive in the single experiments' utilities. Therefore we optimize a tuple of experiments (with size of the horizon, *i.e.* according to the available budget), even though only the first experiment of the sequence will be performed at a given iteration of EDEN. This problem is addressed by seeing the sequence of experiments as arms in a *multi armed bandit* (MAB) problem.

Upper Confidence Tree (UCT). UCT, extending the multi-armed bandit setting to tree-structured search space [24], is one of the most popular algorithm in the MCTS family and was also proposed to solve the problem of active learning in a supervised framework by [34]. Its application to sequential design under budgeted resources is to our knowledge an original proposal. A sketch is given in Algorithm 2.

UCT simultaneously explores and builds a search tree, initially restricted to its root node, along N tree-walks. Each tree-walk involves several phases:
The **bandit phase** starts from the root node (where all available experiments are represented by accessible nodes) and iteratively select experiments until arriving at an unknown node or a leaf (distance T from the root). Experiment selection is handled as a MAB problem. The selected experiment $\tilde{e}_\ell$ in $\mathcal{E}_{\ell,known}$ maximizes the Upper Confidence Bound [1]:

$$\tilde{e}_\ell = \arg \max_{node_i \in \mathcal{E}_{\ell,known}} \left(\hat{\mu}_i + C \times \sqrt{\frac{log(\sum_j n_j)}{n_i}} \right) \quad . \tag{3}$$

where :

- $\mathcal{E}_{\ell,known}$ is the set of known nodes (already visited) which are accessible from the current position (ℓ^{th} experiment in the Path).
- $\hat{\mu}_i$ is the mean utility of node i
- n_i is the number of time node i has been visited before
- C is a tuning constant that favors exploration when high and exploitation when low. Its value is problem specific and must be compared to both the number of possible experiments and the overall mean utility of a sequence of experiments. In the illustration, we used $C = \sqrt{10}$.

The bandit phase stops upon arriving at an unknown node (or leaf). Then in the **tree building phase**, a new experiment is selected at random and added as a child node of this current leaf node. This is repeated until arriving in a terminal state as determined by the size of the horizon. Overall, we can summarize this procedure as going from root to a leaf following a path of length T. When children nodes are known, the UCB criterion is applied, when they are not known, a random choice is performed and the node is created. At this point the reward R of the whole sequence of experiments is computed and used to update the cumulative reward estimates in all nodes visited during the tree-walk.

One of the great features of the proposed method is to perform a *biased* Monte Carlo tree search thanks to the UCB criterion which preserves optimality asymptotically and ensures we build an UCT. After some pure random exploration of the tree, this criterion makes a rational trade-off between exploration (valuation of untested sequences of experiments) and exploitation (improving the estimation of mean utility for an already tested sequence).

When a sufficient number of tree walks has been performed, we select the next experiment to make among the nodes (experiments) directly connected to the root. This choice is based on the best mean score (but could have been selected by taking the most visited node: when the number of tree walks is sufficiently high, these two options give the same results).

Surrogate Hypothesis. A reward function is thus required that measures how informative a sequence of experiment is. The tricky issue is that the true parameter vector $\boldsymbol{\theta}_{true}$ is not known and therefore cannot be used as a reference for evaluating the obtained estimates. As in [34], we proceed by associating to each tree-walk a surrogate hypothesis $\boldsymbol{\theta}^*$, drawn from the version space $\Theta_{\ell-1}$, that will represent the true parameter $\boldsymbol{\theta}_{true}$ in the current tree walk. The reward R attached to this tree walk is computed by i) estimating the parameters $\hat{\boldsymbol{\theta}}$ from the obtained dataset; ii) evaluating the estimate $\hat{\boldsymbol{\theta}}$.

Here we present two different approaches to evaluate this estimate and thus to calculate the reward. The reward R_1 calculates a quantity related to the (log) empirical bias of the parameter estimate. The average reward associated to a node of the tree, i.e. to a sequence of experiments, thus estimates the expectation over $\Theta_{\ell-1}$ of the estimation error yielded by the choice of this sequence, e.g. the (log) bias of the parameter estimate. The reward R_2 calculates the empirical variance of the parameter estimate and thus does not use the current surrogate hypothesis $\boldsymbol{\theta}^*$.

Algorithm 2. UCT pseudo-code

Input:
Hypothesis Space: Θ
Budget: B
Max Horizon: T
Maximal number of tree-walks: N
Initialize :
$walk = 1$
while $walk \leq N$ **do**
  $current\_node = root$
  Sample a surrogate hypothesis: $\boldsymbol{\theta} \sim \Theta$
  $Path = \{current\_node\}$
  Init virtual budget: $b = min(B, T)$
  while $b \geq min_{i \in \mathcal{E}}(cost(e_i))$ **do**
    $e = UCB(current\_node)$
    $current\_node = e$
    $Path = \{Path \cup current\_node\}$
    $b = b - cost(e)$
  end while
  $Reward = R(Path, \boldsymbol{\theta}^*)$
  Update path score: $Update(Path, Reward)$
  $walk = walk + 1$
end while
$e^* = MaxReward(root)$

Reward 1. The concept of this utility function is to quantify how well the selected experiments allow the parameters' estimation to converge towards the true parameters. At each turn ℓ, the uncertainty on the true parameters of the system is captured by the distribution of likely parameter candidates $\boldsymbol{\theta}^* \in \Theta_{\ell-1}$. The utility function for $R1$ does not require any specific assumption on the model itself and only requires an estimation method and a way to value the quality of that estimation. It is computed using the following procedure: Let $\boldsymbol{\theta}^* \in \Theta$ be the current surrogate hypothesis, and Estimation : (prior $\pi, \tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*)) \mapsto \hat{\boldsymbol{\theta}}$ be an estimation procedure, here bayesian, where π is some prior distribution on $\boldsymbol{\theta}$, $\tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*)$ is the set of simulated data according to the observation model given in the problem setting and corresponding to a sequence of k experiments, $\tilde{e}_{1:k}$, when considering the surrogate hypothesis $\boldsymbol{\theta}^*$ as the true parameters. We can evaluate this estimation by comparing the estimated parameters, $\hat{\boldsymbol{\theta}} = E[\boldsymbol{\theta}|\tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*))]$ to the current $\boldsymbol{\theta}^*$. In this work we use the following metric to measure the quality of estimate $\hat{\boldsymbol{\theta}}$, based on the DREAM 7 challenge [30]:

$$d(\boldsymbol{\theta}^*, \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{m} ln\left(\frac{\theta_i^*}{\hat{\theta}_i}\right)^2 \quad . \tag{4}$$

Where θ_i^* is the i^{th} component of θ_i^* and we sum over all components. Overall, this defines a semi-metric (lacking triangular inequality) on the space of parameters. This semi-metric is proportional to the mean squared logarithmic ratio

of the parameters, and so penalizes fold changes in parameters' values. This is especially relevant in estimating biological parameter values that can span several orders of magnitude and where observables may be very insensitive to some parameter values [17]. With all these notations, the utility function returned at each iteration of the MCTS is:

$$r1(\tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*), \pi, \boldsymbol{\theta}^*) = -d(\boldsymbol{\theta}^*, \hat{\boldsymbol{\theta}}) \ . \tag{5}$$

In this work, we chose the prior π as a Gaussian distribution whose mean is a randomly perturbed $\boldsymbol{\theta}^*$ (mean of $\pi = \boldsymbol{\theta}^*.\epsilon$) with $\epsilon \sim \mathcal{N}(1, 0.1)$. The prior covariance is set to the identity matrix. Because of the prior π, we only perform an estimation around the target value $\boldsymbol{\theta}^*$, which explains why this reward is called *local*. The rationale behind this being that, assuming our representation of the version space is fine enough, we will always find a sample not too far away from the true value (here, further than 10% on average for each dimension). In the end, since this function is called within a MCTS framework, the relevant utility for the selection of experiments is the average over different calls to the function. Given $Path = (\tilde{e}_{1:k})$ the sequence of chosen experiments, we have:

$$R_1(\tilde{e}_{1:k}, \boldsymbol{\theta}^*) = \mathbb{E}_\epsilon[r1(\tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*), \pi_\epsilon, \boldsymbol{\theta}^*)] \ . \tag{6}$$

Thus R_1 compares the expectation of the posterior probability defined from data $\tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*)$ produced by experiments $\tilde{e}_{1:k}$ to the parameter $\boldsymbol{\theta}^*$, coordinate by coordinate. The main interest of this reward function is that it can be straightforwardly applied to any estimation method and that its only significant assumption is that the version space is fine-grained enough. In this respect, it is said to be *agnostic*. On the other hand, its main drawback is that it is usually computationally expensive (depending on the estimation scheme used).

Reward 2. In the second reward, we also solve an estimation problem using a joint UKF starting from a Gaussian prior centered on the surrogate θ^* and with an identity covariance matrix. The reward is classically defined in relation to the evolution of the trace of the covariance of the posterior :

$$R_2(\tilde{e}_{1:k}, \boldsymbol{\theta}^*) = -\sum_{i=1}^{m} VAR[\theta_i | \tilde{\mathbf{Y}}_{1:k}(\boldsymbol{\theta}^*)] \ . \tag{7}$$

3.4 Performing Experiments and Re-estimation of Parameters and Hidden Variables

Having estimated an optimal sequence of experiments, we will perform (or simulate noisy data for the in-silico illustration) one experiment only. This allows us to subsequently choose the next experiment benefiting from the new information brought by the genuine acquired data.

An estimation procedure is required in the *virtual games* (MCTS iterations) each time a reward of a sequence of experiments has to be calculated, as well

as in the *real game*, when the real data are acquired. In the case of the *virtual game*, each experiment $\tilde{e}$ in a sequence to be evaluated corresponds to the basal model perturbed with some specific exogenous input $\mathbf{u}(t) = \mathbf{g}_{\tilde{e}}(t)$. Therefore, the learning problem turns to the joint estimation of different models sharing some parameters (and some states). To achieve this joint learning task, we propose an original strategy that consists of aggregating the different models corresponding to different interventions into a single fused model. Then, we apply a Bayesian filtering approach devoted to nonlinear state-space models, the UKF, to the new system. Parameters are estimated together with hidden states using an augmented state approach. UKF provides an approximation of the posterior probability of $\boldsymbol{\theta}$ given the multiple time series corresponding to the multiple experiments, allowing to calculate the different rewards, the bias-like reward R_1 or the variance reward R_2.
In the case of the *real game*, at each turn, different models have to be jointly learnt from the previous datasets and the new one, just acquired after a purchase of an experiment. This can be performed exactly the same way using UKF on a single fused model.

3.5 Evaluation of the Quality of Estimates in the Real Game

During the real game, the quality of the estimate is measured by the trace of the covariance of the UKF estimate.

4 Application to Reverse-Modeling of Gene Regulatory Networks

4.1 Model Setting

Let us consider a simple gene regulatory network that implements the transcriptional regulatory mechanisms at work in the cell [26]. We denote by d the number of genes and assume, for the sake of simplicity, that one gene codes for one protein. In contrast, a gene can be regulated by several genes, including self-regulation, with interactions of several possible types, additive or multiplicative: a gene j is said to regulate a gene i if the level of expression of gene j influences the level of expression of gene i. The vector $\mathbf{r}(t) \in \mathbb{R}^d$ denotes the expression levels (mRNA concentration) of the d genes at time t and the vector $\mathbf{p}(t) \in \mathbb{R}^d$, the concentration of the encoded proteins. Similarly to one of the challenges [30,15] in DREAM6 (2011) and DREAM7 (2012), we consider a problem of parameter and hidden variable estimation in a Hill kinetics model. In the numerical simulations, we apply EDEN on 3 different reverse-modeling problems of GRN of increasing size (3, 5, 7) whose graphs are represented in Fig. 1.

In the following, we introduce the ODE system and the set of experiments on the second target dynamical system composed of 5 genes. The dynamics of this network can be represented by the following system of differential equations associated to the regulation graph represented in Fig. 1B:

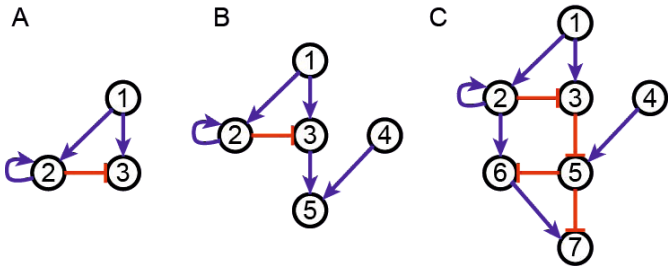

Fig. 1. Regulation graph of the 3 models. Blue arrows represent activations and red bars represent inhibitions.

$$
\begin{aligned}
\dot{r_1}(t) &= \gamma_1 - k_1^r \cdot r_1(t) \\
\dot{r_2}(t) &= \gamma_2 \cdot \left(h_{21}^+(t) + h_{22}^+(t)\right) - k_2^r \cdot r_2(t) \\
\dot{r_3}(t) &= \gamma_3 \cdot h_{31}^+(t) \cdot h_{32}^-(t) - k_3^r \cdot r_3(t) \\
\dot{r_4}(t) &= \gamma_4 - k_4^r \cdot r_4(t) \\
\dot{r_5}(t) &= \gamma_5 \cdot \left(h_{53}^+(t) + h_{54}^+(t)\right) - k_5^r \cdot r_5(t) \\
\dot{p_i}(t) &= \rho_i \cdot r_i(t) - k_i^p \cdot p_i(t), \quad \forall i = \{1, \ldots, 5\} \ . \qquad (8)
\end{aligned}
$$

where h_{ji}^+ is the Hill function for activation defined as: $h_{ji}^+(t) = \frac{p_j(t)^2}{K_{ji}^2 + p_j(t)^2}$ and h_{ij}^- is the Hill function for inhibition defined as: $h_{ji}^-(t) = \frac{K_{ji}^2}{K_{ji}^2 + p_j(t)^2}$. The parameters K_{ji} is called dissociation constant of the regulation of gene i by the protein p_j. The set of parameter to estimate is then $\boldsymbol{\theta} = [(\gamma_i)_{i=\{1,\ldots,5\}}, (K_{ji})_{\{(i,j),\ j \to i\}}]$ and the state vector: $\mathbf{x}(t)^T = [\mathbf{r}(t)\ \mathbf{p}(t)]^T$. As in the DREAM7 challenge, the initial conditions are chosen as: $\mathbf{x}_0 = [0.4\ 0.7\ 0.5\ 0.1\ 0.9\ 0.4\ 0.3\ 1.0\ 1.0\ 0.8]^T$ and are used to simulate as well the wild-type as the perturbation experiments. As for the observations, only one type of state variable, protein concentrations or mRNA concentrations, can be measured at a time, the other one being then considered as hidden state.

Two kinds of perturbations are considered: the knock-out (ko) that fully represses the expression of the targeted gene, and the over-expression (oe) that accelerates the translation of the targeted protein. In our problem, only one perturbation can be applied at a time. In order to simulate the behavior of the perturbed system, we introduce in the model two types of intervention functions, $\mathbf{g_{oe}}$ and $\mathbf{g_{ko}}$, for each gene. The wildtype system corresponds to the case of these two control variables being equal to 1. Taking $g_{ko}^i(t) = 0, \forall t \geq 0$, for gene i, simulates a knock-out on this gene by removing the production term of mRNA and protein. For instance, under a knock-out on gene 1, the equations for mRNA 1 write as:

$$\dot{r_1}(t) = g_{ko1}(t) \cdot \gamma_1$$

Taking $g_{oe}^i(t) = 2, \forall t \geq 0$, for protein i, simulates the corresponding over-expression since the production term of the protein concentration p_i is then doubled. Applying this perturbation on gene 1 gives:

$$\dot{p_1}(t) = g_{oe1}(t) \cdot \rho_1 \cdot r_1(t) - k_1^p \cdot p_1(t) \ . \tag{9}$$

Overall, 11 perturbations are considered including the wild-type, with two possible observation models (either protein or mRNA concentrations for each of them), giving in total 22 potential experiments to perform for the 5-genes network. For the 3- (resp. 7) genes network, 14 (resp. 30) experiments are considered.

4.2 Numerical Results

In this section, we describe the results we obtained on the systems described in the previous section. These simulations of an experimental design problem were performed using the two reward functions R_1 and R_2 described in (6) and (7) with an hypothesis space Θ represented by samples (1000 if not stated otherwise). A convergence criterion was used in order to limit the number of tree walks performed in the MCTS phase of the algorithm. This criterion allowed to stop tree walks as soon as the mean utility associated to all experiments did not change by more than 10% over the last 20 walks. For all details, you are encouraged to request our Matlab© code (based on the *pymaBandits* framework [20,8,19]).

Figure 4.2 shows that some well chosen experiments provide a significant (more than 100 folds) reduction of the uncertainty on parameters' value. But some of the additional experiments can lead to only marginally decreasing the quality of estimation. The experiments chosen with R_1 or R_2 are not the same. We also see in Figure 4.2 B that the number of samples forming the version space can change significantly the performance as the algorithm takes into account uncertainty on the system which is related to the number of samples. Although the results after 5 experiment purchases are similar, the same performance can be achieved with only 3 experiments if uncertainty is properly accounted for.

Figure 3 A reports the scores obtained by applying the reward R_1 for 3 sizes of network. All were using an horizon of $T = 3$ experiments and a version space represented by 1000 samples. Concerning the scores, the more complex problem leads nearly only to a reduction on the uncertainty but could not improve significantly the estimations. This is because increasing complexity implies usually more non-identifiability and requires a larger budget. These results also illustrate the complexity of experimental design: since less genes means less means to acquire data indirectly on a gene's parameters, the 3-gene network is not significantly better estimated than the 5 genes network with the same learning horizon. Concerning the computation scaling, networks of 3, 5 and 7 genes have respectively required 6, 14 and 24 hours on a quad-core Intel i7 processor at 4,5 GHz.

Current methods for sequential experimental design generally optimize one experiment at a time. Even if we only acquire one experiment per iteration of

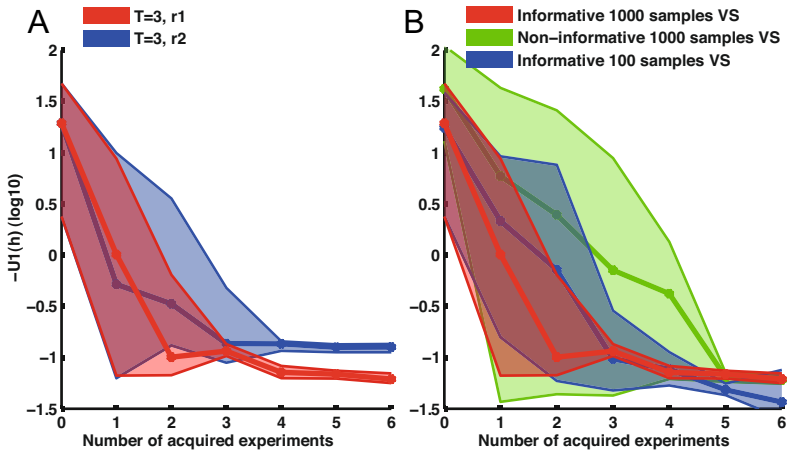

Fig. 2. A. Impact of the reward function: evolution of the (log10) scores for the 5-gene network, $d(\theta_{true}, \theta)$ for $\theta \in \Theta$ (1000 samples) using either R_1 or R_2. Are plotted the minimum, the maximum and the average over all Θ. Starting from a large Θ which only included information from the unperturbed (*wildtype*) observations, a well chosen sequence of experiments can lead to very significant improvement of the estimation of the parameters. B. Effect of the version space representation: We compare the performance on the 5- gene model, with an horizon T=3 for EDEN run starting from either the typical 1000 samples Θ which contains the information from the *wildtype* observation, the same but using 1000 uniformly distributed samples (non-informative) or an 100 sample version space (taking the best 100 samples in terms of mean squared deviation from the prior information) from the informative version space.

EDEN, we optimize sequences of experiments up to a given *learning horizon*, T. This obviously implies sampling in a much bigger space of possible designs ($O(|\mathcal{E}|^T)$) but allows at the same time to consider experimental strategies that mitigate the risk of individual experiments when the outcome is uncertain. As we can see on figure 3 B, a different learning horizon can change importantly the speed of reduction of uncertainty and estimation quality. Interestingly for that particular problem, an horizon of 3 lead the algorithm to take some risk (given the uncertainty on the system) that did not pay-off as a greedy version (T=1) performs better. But with a larger horizon (T=5), the risk mitigation is differently considered by the algorithm and an excellent performance is achieved in 2 experiments only.

5 Conclusion

We developed an active learning approach, EDEN, based on a one-player game paradigm to improve parameters and hidden states estimates of a dynamical system. This setting is identical with that of active learning [34] for supervised

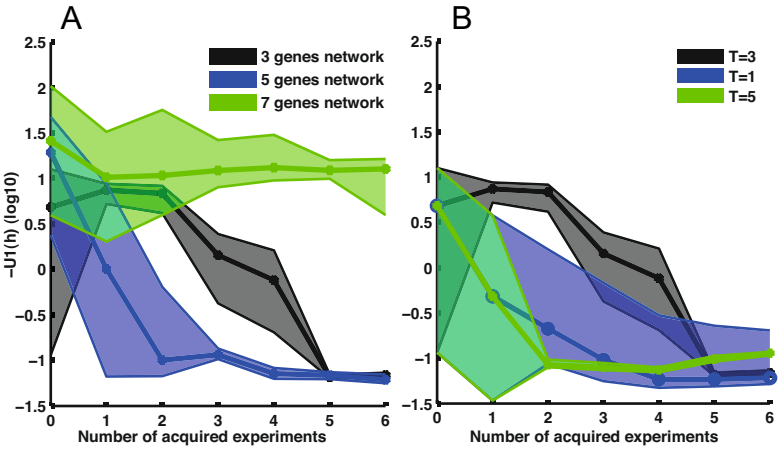

Fig. 3. A. Performance on problems of increasing complexity: evolution of the (log10) scores $d(\theta_{true}, \theta)$ for the 3,5 and 7-gene networks, for $\theta \in \Theta$ (1000 samples) using R_1. Are plotted the minimum, the maximum and the average over all Θ. B. Performance for various learning horizon T=1, 3 and 5 for the 3 genes model, using R_1 and a 1000 samples version space.

learning, where theoretical guarantees have been given along the following lines. The active learning (here experimental design) problem is equivalent to a reinforcement learning problem that can be expressed formally in terms of a Markov Decision Process; this problem is intractable but approximation with asymptotic guarantees are provided by the UCT algorithm [24,29,34]. Future work will focus on lightening these guarantees in the framework of dynamical system identification.

Furthermore, to our knowledge, this is the first application of UCT-based approaches to sequential experimental design for dynamical nonlinear systems, opening the door to a very large number of potential applications in scientific fields where experiments are expensive. The versatility of the proposed framework allows to extend it in various ways. Different dynamical models including stochastic ones can be in principle used with this strategy while other rewards can be designed. An interesting perspective is also to link the theoretical guarantee brought by UCT with the framework of Bayesian experimental design [28]. Finally, considering the scalability issue, we notice that the learning horizon (the number of experiments we jointly optimize) does not need to scale with the size of the model. In fact, the relevant horizon is the number of experiments allowing to eliminate the non-identifiability for the set of parameters of a given model. This means that the approach can be in principle extended to larger systems. Although automated experimental design approaches are still an exception in wet laboratories, some pioneering works on the robot scientist *Adam* [21,22] show the immense potential offered by realistic and practice-oriented active learning in biology and other experimental sciences.

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning 47(2-3), 235–256 (2002)
2. Asprey, S.P., Macchietto, S.: Statistical tools for optimal dynamic model building. Computers & Chemical Engineering 24:2017, 1261–1267 (2000)
3. Azimi, J., Fern, A., Fern, X.Z., Borradaile, G., Heeringa, B.: Batch Active Learning via Coordinated Matching. In: ICML 2012. Omnipress (2012)
4. Baldi, P., Hatfield, G.W.: DNA microarrays and gene expression: from experiments to data analysis and modeling. Cambridge University Press (2002)
5. Bandara, S., Schlöder, J.P., Eils, R., Bock, H.G., Meyer, T.: Optimal experimental design for parameter estimation of a cell signaling model. PLoS Computational Biology 5(11), e1000558 (2009)
6. Blot, W.J., Meeter, D.A.: Sequential experimental design procedures. Jour. of American Statistical Association 68(343), 343 (1973)
7. Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Colton, S.: A survey of Monte-Carlo tree search methods. Intelligence and AI 4(1), 1–49 (2012)
8. Cappé, O., Garivier, A., Maillard, O.A.: Kullback-Leibler upper confidence bounds for optimal sequential allocation. Annals of Statistics, 1–56 (2012)
9. Coquelin, P.A., Munos, R.: Bandit algorithms for tree search. In: Proc. of Int. Conf. on Uncertainty in Artificial Intelligence (2007)
10. Chaloner, K., Verdinelli, I.: Bayesian experimental design: a review. Statistical Science 10(3), 273–304 (1995)
11. Chou, I.C., Voit, E.O.: Recent developments in parameter estimation and structure identification of biochemical and genomic systems. Math. Biosci. 219(2), 57–83 (2009)
12. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Mach. Learn. 15(2), 201–221 (1994)
13. Dasgupta, S.: Analysis of a greedy active learning strategy. In: NIPS 17, pp. 337–344. MIT Press (2005)
14. Hanneke, S.: A bound on the label complexity of agnostic active learning. In: ICML 2007, pp. 353–360. ACM (2007)
15. The dream project website: `http://www.the-dream-project.org/`
16. Franceschini, G., Macchietto, S.: Model-based design of experiments for parameter precision. Chemical Engineering Science 63(19), 4846–4872 (2008)
17. Gutenkunst, R.N., Waterfall, J.J., Casey, F.P., Brown, K.S., Myers, C.R., Sethna, J.P.: Universally sloppy parameter sensitivities in systems biology models. PLoS Computational Biology (10), 1871–1878 (2007)
18. Hirsch, M.W., Smale, S.: Differential Equations, Dynamical Systems, and Linear Algebra. Academic press (1974)
19. Kaufmann, E., Cappé, O., Garivier, A.: On bayesian upper confidence bounds for bandit problems. In: Proc. AISTATS, JMLR W&CP, La Palma, Canary Islands, vol. 22, pp. 592–600 (2012)
20. Garivier, A., Cappé O.: The KL-UCB Algorithm for bounded stochastic bandits and beyond. In: COLT, Budapest, Hungary (2011)
21. King, R.D., Whelan, K.E., Jones, F.M., Reiser, P.G., Bryant, C.H., Muggleton, S.H., Kell, D.B., Oliver, S.G.: Functional genomic hypothesis generation and experimentation by a robot scientist. Nature 427(6971), 247–252 (2004)