33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Generation Of Intervention Strategy For A Genetic Regulatory Network Represented By A Family Of Markov Chains

Noah Berlow and Ranadip Pal

Abstract— **Genetic Regulatory Networks (GRNs) are frequently modeled as Markov Chains providing the transition probabilities of moving from one state of the network to another. The inverse problem of inference of the Markov Chain from noisy and limited experimental data is an ill posed problem and often generates multiple model possibilities instead of a unique one. In this article, we address the issue of intervention in a genetic regulatory network represented by a family of Markov Chains. The purpose of intervention is to alter the steady state probability distribution of the GRN as the steady states are considered to be representative of the phenotypes. We consider robust stationary control policies with best expected behavior. The extreme computational complexity involved in search of robust stationary control policies is mitigated by using a sequential approach to control policy generation and utilizing computationally efficient techniques for updating the stationary probability distribution of a Markov chain following a rank one perturbation.**

I. Introduction

Genetic Regulatory Networks are often represented as Markov Chains such as Stochastic Master Equation models that are a form of continuous time Markov Chains [1], [2] or Probabilistic Boolean Network models which are coarse-scale Markov Chain models [3]. One of the objectives of Genetic Regulatory Network (GRN) modeling is to design and analyze therapeutic intervention strategies aimed at moving the network out of undesirable states, such as those associated with disease, and into desirable ones. However, limited experimental data prevent accurate inference of the mathematical model of the GRN. For the success of a mathematically designed intervention strategy for genetic diseases, it is critical that the designed intervention strategy posses some degree of robustness to the modeling uncertainties. In this paper, we will consider the case of the GRN being modeled by a family of Markov Chains and our interest is in generating a stationary control policy that will provide the best expected performance over the family of Markov Chains.

Optimal control of Markov Processes has a long history starting from the works of Richard Bellman and Lev Pontryagin. However, most of the development in optimal control of Markov Processes are based on the perfect knowledge of the underlying Markov process of the system. Optimal control approaches has been developed for finite and infinite horizon control for both perfectly observable and partially observable states with the assumption of the knowledge of the underlying Markov process [4]. In real life, we are often faced with the scenario of uncertainty in estimating the parameters of the underlying Markov process which necessitates development of Robust Dynamic Programming approaches. Robust dynamic programming from the perspective of worst case or min-max approach has been recently studied [5]. However, worst case approach is often conservative, giving too much importance to events that have extremely small chance of occurrence. Thus, an expected or Bayesian approach to optimal control design has to be pursued when our objective is to improve the expected chances of success. In this article, we will consider the design of computationally in-expensive sub-optimal solutions of Bayesian robust control for a family of Markov Chains.

The paper is organized as follows. Section 2 provides the mathematical description of the control problem; Section 3 presents the algorithm for robust control policy generation; the complexity analysis of the algorithm is presented in Section 4; empirical results are presented in Section 5 while the conclusions are presented in Section 6.

II. Problem Description

To explain infinite-horizon control of Markov Processes, let us consider a finite state Markov chain described by the control-dependent one-step transition probability $p_{ij}(u) := P(z_{t+1} = j | z_t = i, u_t = u)$ where, for all t, the state z_t is an element of a space S and the control input u_t is an element of a space C. When the transition probabilities are exactly known, the states make transitions according to $\omega := (P^u)_{u \in C}$. Let $\mu = (u_1, u_2, ...)$ represent a generic control policy and Π represent the set of all possible μ's, i.e., the set of all possible control policies. Let $J_{\mu,\omega}$ denote the expected total cost for the average cost per stage infinite-horizon problem [6] under control policy μ and transitions ω:

$$J_{\mu,\omega}(z_0) = \lim_{M\to\infty} \frac{1}{M} E\{\sum_{t=0}^{M-1} \tilde{g}(z_t, \mu_t(z_t), w_t)\}. \quad (1)$$

where $\tilde{g}(z_t, u_t, z_{t+1})$ represents the cost of going from state z_t to z_{t+1} under the control action u_t. $\tilde{g}$ is higher for undesirable destination states. The control problem here corresponds to minimizing the cost in Eq. 1. Consequently, the optimal infinite-horizon discounted cost is given by:

$$\Phi(\Pi, \omega, z_0) := \min_{\mu \in \Pi} J_{\mu,\omega}(z_0). \quad (2)$$

This work was supported by NSF grant CCF 0953366

N. Berlow is with the Electrical and Computer Engineering Department, Texas Tech University, Lubbock, TX, 79409 USA noah.berlow@ttu.edu

R. Pal is with the Electrical and Computer Engineering, Texas Tech University, Lubbock, TX, 79409 USA ranadip.pal@ttu.edu

To explain the issue of robustness, let us re-consider the nominal control problem described by Eq. 2. In case of uncertainties, we can parameterizing the class of transitions as $\Omega := (P_a^u)_{u \in C, a \in F_a}$, where F_a is the noise parameter distribution.

One of the ways to approach robust intervention in presence of uncertainties is to consider the worst-case scenario. A *minimax(worst-case) intervention policy* is defined as a policy whose worst performance over the uncertainty class Ω is best among all admissible policies. The *minimax robust policy*, denoted μ_{mm}, is the one that satisfies $\Phi(\Pi, \Omega, z_0) := \min_{\mu \in \Pi} \max_{\omega \in \Omega} J_{\mu,\omega}(z_0)$. The worst-case robust policy design approach is typically conservative because it gives too much importance to the scenarios which hardly occur in practice. When our objective is to avoid extremely undesirable results, a minimax design is suitable but when our objective is to improve the expected chances of success, a Bayesian approach will be preferable. Let $\gamma_a(\mu_b) = E_{z_0}[J_{\mu_b,\omega_a}(z_0)]$ denote the expected cost per state at point a of the parameter distribution for the intervention policy μ_b. A *Bayesian robust policy* μ_{bp} is one that minimizes $E_a[\gamma_a(\mu_b)] = E_a[E_{z_0}[J_{\mu_b,\omega_a}(z_0)]]$ where E_a denotes expectation relative to the parameter distribution.

We will consider the case that the uncertainty class is discrete and we have L possible Markov Chains representing the underlying system. Consider a System Λ with two families of N x N matrices, $P_1 = \{M_1^1, M_2^1, \ldots, M_L^1\}$ representing the probability transitions under no control, and $P_2 = \{M_1^2, M_2^2, \ldots, M_L^2\}$, representing the family of related matrices under active control. Let μ be the stationary control policy associated with Λ. A stationary control policy on a family of Markov Chains P_1 and P_2 is a control policy independent of time and dependent on the state of the system (sporadic control). Thus for Markov Chains P_1 and P_2 with N states, the stationary control policy μ is a N length binary vector denoting for each state whether control should be applied (The system will then belong to P_2) or not (System will then belong to P_2). The decimal representation of the stationary policy μ will have a range from 0 to $2^N - 1$. Let $P_\mu = \{M_1^\mu, M_2^\mu, \ldots, M_L^\mu\}$ denotes the controlled set of Markov Chains for control policy μ. For each matrix M_j^μ $j \in [1, \cdots, L]$, the ith row of M_j^μ, $M_{j_i}^\mu$, is defined as $M_{j_i}^\mu = M_{j_i}^1$ if $\mu_i = 0$ and $M_{j_i}^\mu = M_{j_i}^2$ if $\mu_i = 1$. If we revisit Eq. 1 and consider the case where $\tilde{g}(z_t, u_t, z_{t+1})$ is only dependent on the final state z_{t+1} and the cost of control is zero, then minimizing $E_a[\gamma_a(\mu_b)]$ will be equivalent to maximizing the steady state probabilities of desirable states and minimizing the steady state probabilities of undesirable states. In fact, if G is a length N vector representing the cost of the N states and π_i^μ represent the steady state probability distribution for M_i^μ for $i \in [1, 2, \cdots, L]$, then $E_a[\gamma_a(\mu_b)] = \sum_{i=1}^{L} G(\pi_i^{\mu_b})^T$. We will denote $E_a[\gamma_a(\mu_b)]$ by $\Gamma(\mu_b)$ henceforth. For our proposed algorithm, μ_b will be denoted by a set S containing the indices i for which the control is on i.e. $\mu_{b,i} = 1$ and set F will denote application of control for all states i.e. $F = \{1, 2, \cdots, N\}$.

III. Algorithm for Stationary Control Policy Generation

Exhaustive search of stationary policies for Bayesian robustness is almost impossible for large number of states as the total number of possible policies for N states and m controls is m^N. Robust dynamic programming based approaches similar to to min-max control will not apply here as the principle of optimality is not valid for the expected cost formulation. The approach presented in [7] minimizes the expected cost from among the optimal policies for each individual Markov Chains but a policy can exist that is not among the individual optimal policies and produce a lower expected cost. Furthermore, that approach also had huge computational complexity. The purpose of this paper is to present a new computationally efficient sub-optimal approach for calculation of robust control policies for a family of Markov Chains with the objective of altering the expected steady-state distribution of the family of networks. Our proposed approach for *computationally inexpensive sub-optimal solution* is based on exploring the control policies like a feature selection problem and apply a sequential search such as sequential floating forward search (SFFS) [8] approach. Let us consider our case of L possible networks and binary controls and we want a stationary policy μ that will give the best expected cost over the L networks. Since our objective is to alter the steady state probabilities of the networks, then we can consider changing one entry of μ at a time and observe the change in the steady-state probability distributions of the L networks and keep the change that produces the lowest expected cost. Then we add one more change and so forth and the option of going back will be embedded similar to SFFS [8]. The computational complexity in calculation of the L steady state probability distributions for each change will be reduced by using techniques for updating the stationary probability distribution of a Markov chain following a rank one perturbation [9]. The new approach for generating the robust stationary policy is presented as Algorithm 1.

Algorithm 1 Algorithm for generating stationary control policy

Require: G: State Cost Vector
$M_1 = \{M_1^1, M_2^1, \ldots, M_L^1\}$; $M_2 = \{M_1^2, M_2^2, \ldots, M_L^2\}$
Ensure: S, the suboptimal robust stationary control policy
$S = \emptyset$
$x_1 = \arg\max_{x \in F \setminus S} \{\Gamma(S \cup \{x\})\}$
while $\Gamma(S \cup \{x_1\}) > \Gamma(S)$ and $|S| < N$ **do**
  $S = S \cup \{x_1\}$
  $x_2 = \arg\max_{x \in S} \{\Gamma(S \setminus \{x\})\}$
  while $\Gamma(S \setminus \{x_2\}) > \Gamma(S)$ **do**
    $S = S \setminus \{x_2\}$
    $x_2 = \arg\max_{x \in S} \{\Gamma(S \setminus \{x\})\}$
  end while
  $x_1 = \arg\max_{x \in F \setminus S} \{\Gamma(S \cup \{x\})\}$
end while

A. Applying Perturbation Theory of Markov Chains to calculate Γ

We will consider the approach provided in [9] to efficiently calculate Γ at each stage of the addition or deletion process. Note that the addition or deletion of a control for a single state is equivalent to perturbing one row of the controlled matrix. Consider the rank one perturbation on a Markov Chain M_1 with steady-state distribution $\pi^{(1)} = (\pi_1^{(1)}, \pi_2^{(1)}, \ldots, \pi_N^{(1)})$ resulting in a new Markov Chain M_2 represented by $M_2 = M_1 + e_r' b$ where e_r is the $N \times 1$ column vector which is 1 at position r and 0 elsewhere and $b = M_r^2 - M_r^1$ is $1 \times N$ row vector with $b\mathbf{1} \neq 0$. The fundamental matrix Z for an irreducible Markov Chain M with steady state distribution $\pi^{(1)}$ is given by the equation $Z = [I - M + 1\pi^{(1)}]^{-1}$. For Markov Chain M_1 with row r being perturbed, let

$$\beta = (\beta_1, \ldots, \beta_n) = (P_r^2 - P_r^1)[I - P + 1\pi^{(1)}]^{-1} = bZ \quad (3)$$

Using β, the steady-state distribution of M_2 denoted by $\pi^{(2)} = (\pi_2^{(2)}, \pi_2^{(2)}, \ldots, \pi_n^{(1)})$, can be calculated directly. For each $\pi_i^{(2)} \in \pi^{(2)}$, $\pi_i^{(2)} = \pi_i^{(1)} + \pi_r^{(1)}[\beta_i/(1 - \beta_r)]$. In matrix form, this can be quickly calculated with the equation

$$\pi^{(2)} = \pi^{(1)} + \pi_r^{(1)}[\beta/(1 - \beta_r)] \quad (4)$$

This method provides a direct equation for calculating the new steady-state distribution of a perturbed Markov Chain. However, using the definition of the Z matrix on M_1, this recalculation of steady-state distribution can only be performed for single perturbations on M_1. Hence, a new Z matrix needs to be constructed for the matrix resulting from the perturbation, M_2, to allow for calculation of further perturbations. Conveniently, the new Z matrix, Z_2, can be calculated directly from Z without the need for a matrix inversion operation. Again, suppose M_1 has undergone a perturbation in row r. Z_2 can then be calculated as

$$Z_2 = \left[I - \frac{(\pi^{(1)})^T e_r \cdot \mathbf{1} \cdot b \cdot Z}{1 - b \cdot Z \cdot e_r}\right]\left[Z + \frac{Z \cdot e_r \cdot b \cdot Z}{1 - b \cdot Z \cdot e_r}\right] \quad (5)$$

This Z_2 value, which takes the place of the previous Z value, corresponds to the fundamental matrix for the newly perturbed Markov Chain M_2. The systematic approach for generating $\Gamma(\mu)$ for a change in the rth row is illustrated as Algorithm 2.

IV. Complexity Analysis

Let $S = \emptyset$ be the initial stationary control policy. The first step necessary is calculating the steady-state distribution of the matrices in M_1. The steady-state probabilities of a N x N matrix can be computed by a matrix inversion with a complexity of $O(N^3)$ using Gaussian Elimination [1]. Thus, this initial step will have a complexity of $O(L \cdot N^3)$ computation. The next step computation of the L Z matrices has a computational complexity of $O(L \cdot N^3)$.

[1] An improved complexity of $O(N^{2.376})$ can be achieved by Coppersmith and Winograd method [10]

Algorithm 2 Algorithm for generating $\Gamma(\mu)$ for change in the rth row

Require: G: State Cost Vector
rth row perturbations: $b_1, b_2, \cdots, b_N$
Fundamental Matrices: $Z_1, Z_2, \cdots, Z_N$
$\pi_1^{(1)}, \pi_2^{(1)}, \cdots, \pi_N^{(1)}$
Ensure: $\Gamma(\mu)$, the expected cost of the stationary policy μ
$Z_1, Z_2, \cdots, Z_N$, the new fundamental matrices values
$\pi_1^{(2)}, \pi_2^{(2)}, \cdots, \pi_N^{(2)}$, the new steady state probability vectors
for $i = 1$ to L **do**
$\beta^{(i)} = b_i Z_i$
$\pi_i^{(2)} = \pi_i^{(1)} + \pi_i^{(1)}(r)[\beta^{(i)}/(1 - \beta^{(i)}(r))]$
$Z_i = \left[I - \frac{(\pi_i^{(1)})^T \cdot e_r \cdot \mathbf{1} \cdot b_i \cdot Z_i}{1 - b_i \cdot Z_i \cdot e_r}\right]\left[Z + \frac{Z_i \cdot e_r \cdot b_i \cdot Z_i}{1 - b_i \cdot Z_i \cdot e_r}\right]$
end for
$\Gamma(\mu) = 0$
for $i = 1$ to L **do**
$\Gamma(\mu) = \Gamma(\mu) + \pi_i^{(2)} \times G^T$
end for

The next step is the calculation of the single row perturbed steady state distributions. This calculation requires $N^2 + 2 \cdot N + 1$ operations for a single steady-state recalculation. This calculation must be performed on every matrix in M_1 and for each row in each $M^i \in M_1$, resulting in $N \cdot L\left(N^2 + 2 \cdot N + 1\right)$ operations to compute all the possible perturbed steady-state distributions. For subsequent additions to the static control policy, the number of rows requiring perturbation calculations decreases directly with the size of the static control policy: if $|S| = m$, the cost of calculating the $m{+}1$th element of S is reduced to $(N - m) \cdot L\left(N^2 + 2 \cdot N + 1\right)$. The other steps of calculating the cost of the control policy based on the steady state distributions requires additions and comparisons with complexity much lower than calculating the steady state distributions. After two iterations, the last step of the algorithm comes into play; the "floating", or backtracking, portion of the search. The behavior of the backwards search mimics that of the forward in all respects except the backwards only calculates new steady-state distributions for states already in S. Thus, if $|S| = m$, the reverse step requires $m \cdot L\left(N^2 + 2 \cdot N + 1\right)$ calculations and m comparisons to determine which control, if any, would be the most advantageous to remove from S. Suppose the algorithm, in the process of finding the suboptimal control policy, makes f element additions and b element removals. In practice, b would be usually of the same order as N. The algorithm then requires approximately $L \cdot N^3 + f \cdot N \cdot L\left(N^2 + 2 \cdot N + 1\right) + b \cdot |S| \cdot L\left(N^2 + 2 \cdot N + 1\right) + 2 \cdot N$ calculations to conclude its search. Without the floating part, the presented algorithm has a worst case computational complexity of $O(L \cdot N^4)$. For the floating part, when the number of backwards steps at each step are low (around 2 or 3), the complexity of the presented approach is still $O(L \cdot N^4)$. In comparison, the brute force algorithm has a much larger computational complexity of $O(L \cdot N^3 \cdot 2^N)$.

V. Results

To analyze the empirical performance of our algorithms, we considered various sets of coarse-scale Markov Chains representing GRNs. We considered n genes with binary states (thus $N = 2^n$); $L = 1, \cdots, 5$ and the family of Markov Chains are generated from a initial random Markov Chain using the relationship described in [7]. The control problem formulation is as described in the previous sections and the desirable states are considered to be the states with first gene expression being low (binary zero). We present the results for the case of $N = 2^4 = 16$ (any higher n such as $n = 5$ i.e. $N = 32$, makes the brute force calculations enormous with a complexity of magnitude $(L \cdot 32^3 \cdot 2^{32})$) in Table 1. We notice that the stationary control policy generated by our sub-optimal algorithm has same performance as the brute force algorithm but requiring much less time.

L	$OCost$	$SCost$	$TimeO$	$TimeS$
5	.08	.08	93.3	.76
10	.23	.23	147.8	2.3
35	.59	.59	178.61	2.26
50	.52	.52	309.28	2.86

TABLE I

Results for brute-force and sub-optimal approaches. OCost = Cost reduction by brute-force optimal policy, SCost = cost reduction by the proposed sub-optimal algorithm, TimeO and TimeS represents time taken in seconds for 10 runs by the brute force and Sub-optimal algorithms respectively.

We also empirically compared our algorithm performance with the approach presented in [7]. We selected the same 40 Markov Chains generated from melanoma data based on different noise standard deviations. The expected cost of the min-max approach is 22.02 whereas the expected cost of the Bayesian Approach presented in [7] is 15.19. The expected cost with the control generated by the approach presented in this paper is also 15.19 but time taken to compute the stationary policy by the new approach is 5 times less than the previous approach. This shows the advantage of this new algorithm as compared to brute force and other sub-optimal algorithms.

VI. Conclusions

In this article, we presented a novel way of generating a stationary sub-optimal control policy with best expected performance for a family of Markov Chains. The complexity of the approach is much lower than brute force approach while producing comparable results. This algorithm can have applications in systems medicine when a genetic regulatory network is represented by a family of Markov Chains and our objective is to arrive at a sporadic control policy with best expected performance.

VII. ACKNOWLEDGMENTS

This work was supported by NSF grant CCF 0953366.

References

[1] D. T. Gillespie, "A rigorous derivation of the chemical master equation," *Physica A*, vol. 188, pp. 404–425, 1992.

[2] R. Pal and S. Bhattacharya, "Characterizing the effect of coarse-scale pbn modeling on dynamics and intervention performance of genetic regulatory networks represented by stochastic master equation models," *IEEE Transactions on Signal Processing*, vol. 58, pp. 3341–3351, 2010.

[3] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, pp. 261–274, 2002.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 2nd edition, 2001.

[5] G. Iyengar, "Robust dynamic programming," *Math. Oper. Res.*, vol. 30, pp. 1–21, 2005.

[6] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite horizon control for probabilistic boolean networks," *IEEE Transactions on Signal Processing*, vol. 54, pp. 2375– 2387, 2006.

[7] R. Pal, A. Datta, and E. R. Dougherty, "Bayesian robustness in the control of gene regulatory networks," *IEEE Transactions on Signal Processing*, vol. 57, pp. 3667–3678, 2009.

[8] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection.," *Pattern Recog. Lett.*, vol. 15, pp. 1119–1125, 1994.

[9] J.J. Hunter, "Stationary distributions of perturbed markov chains," *Lin. Alg. Appl.*, vol. 82, pp. 201–214, 1986.

[10] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symbolic Comput.*, vol. 9, pp. 251–280, 1990.