Parameter inference for gene circuit models

Linh Huynh Navneet Rai Ilias Tagkopoulos
Department of Computer Science
& UC Davis Genome Center
University of California, Davis
{huynh,nnrai,itagkopoulos}@ucdavis.edu

1. INTRODUCTION

Parameter inference is crucial in any modeling effort. Parameter inference based on sequential fitting of data to each model leads to erroneous solutions due to over-fitting and ill-constrained parameter bounds. Parameter estimation through fitting multiple models simultaneously can reduce this error, albeit it is computationally intractable for most practical applications. Here, we propose an alternative approach of parameter inference cascades, where parameter values with low uncertainty are propagated to the sequentially fitted models. We propose how to deal with noise in the data and we introduce confidence intervals as a selection metric on parameter value propagation. We demonstrate how this approach reduces parameter estimation error in a synthetic circuit case-study.

2. METHODS AND RESULTS

Model: We use a simple model [1] that can capture both the processes of transcription and translation. More specifically, when a repressor R binds to a promoter p_R, the expression level of a gene g at the downstream of p_R is modeled by

$$\nu_g = \beta_{p_R} + \frac{\alpha_{p_R} - \beta_{p_R}}{1 + \left(\frac{\nu_R}{K_{p_R}}\right)^{n_{p_R}}} \quad (1)$$

where ν_g, ν_R are the expression level of g and R respectively, measured in relative expression units (REU) [2]. Parameters $\beta_{p_R}, \alpha_{p_R}, K_{p_R}, n_{p_R}$ represent the basal level, the promoter strength, the binding affinity, and its cooperativity respectively, pertaining to promoter p_R and its repressor R. In the case where a ligand L_R can bind and inactivate R, ν_R in equation 1 is updated by

$$\nu'_R = \frac{\nu_R}{1 + \left(\frac{[L_R]}{K_{L_R}}\right)^{n_{L_R}}} \quad (2)$$

where $[L_R]$ is the ligand concentration. Parameters K_{L_R} and n_{L_R} correspond to the dissociation constant and the Hill coefficient of the ligand, respectively.

A case study: Assume a cascade of repressors, as depicted in Figure 1. If we use the basic model, there are 16 parameters to capture all four circuits. For our evaluation, we fixed the parameter values, and then generated through simulations the corresponding synthetic datasets, on which we added a 10% of Gaussian noise. We also generated the data in triplicate and calculated the standard deviation of the output to simulate the experimental data in practice.

Model fitting: Suppose that each circuit C_i is modeled by

$$y_i = M_i(x_i, \theta_i) \qquad i = 1, \ldots, 4$$

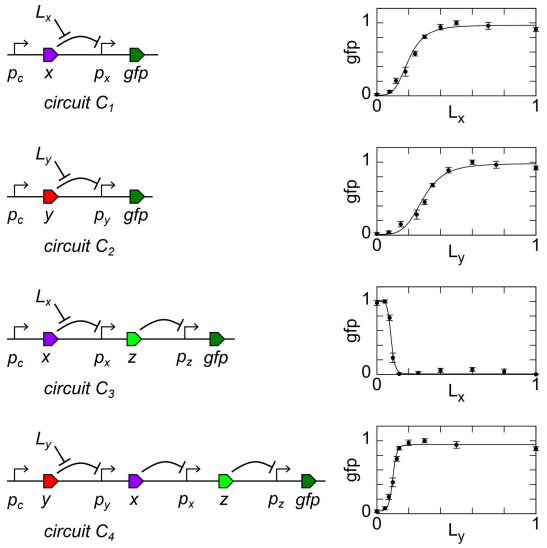

Figure 1: A case study with four cascade circuits (left) and their corresponding simulated data from a model with 10% Gaussian noise.

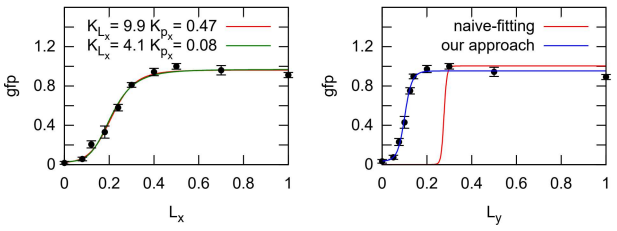

Figure 2: Problems with parameter inference. Multiple optimal solutions exist for the same circuit C_1 (left plot), optimal sequential parameter estimation from circuits 1-3 is unable to simulate points in circuit C_4. The proposed method correctly identifies the optimal parameter set (right plot).

where x_i, y_i, θ_i represent the input, the output, and the set of model parameters, respectively, all for circuit C_i. For example, for the first circuit of Figure 1, x_i is the ligand L_x, y_i is the GFP concentration and θ_i is the set of six parameters that are needed in equations 1 and 2. Each parameter can appear in more than a single circuit, so we denote the set of all parameters $\theta = \bigcup_{i=1}^{4} \theta_i$.

Let $D_i = \{(x_i^{(1)}, y_i^{(1)}, \sigma_i^{(1)}), \ldots, (x_i^{(d_i)}, y_i^{(d_i)}, \sigma_i^{(d_i)})\}$ be a synthetic dataset with d_i data points of the circuit C_i and $\sigma_i^{(j)}$ capturing the standard deviation of the output $y_i^{(j)}$.

For each circuit C_i, if all data points are independent and the output value y_i has a Gaussian distribution then the log-likelihood [3] is

$$LL(D_i|\theta_i) = -\frac{1}{2}\sum_{j=1}^{d_i}\left(\frac{M_i(x_i^{(j)}, \theta_i) - y_i^{(j)}}{\sigma_i^{(j)}}\right)^2 + const$$

We can fit the value θ_i^* for parameters of each circuit C_i seperately by solving the maximum likelihood problem

$$\theta_i^* = \underset{\theta_i}{argmax}\ LL(D_i|\theta_i)$$

If the model M_i is sloppy [4], then two different parameter value combinations may have similar model outputs as in Figure 2. If we fit a sloppy model with a given dataset that contains noise, the fitted parameter values may be far from the actual values. To aleviate this, we can add more constraints on the parameters by fitting all the models simultaneously:

$$\theta^* = \underset{\theta}{argmax} \sum_{i=1}^{4} LL(D_i|\theta_i)$$

However, solving this problem is computationally intractable due to the number of parameters involved. To reduce the computational cost, we can perform a sequential fitting, where the fitted parameter values are propagated to the next fitted model. However, any errors are also propagated and accumulated, which leads to erroneous solutions, as shown in Figure 2. To minimize this issue, we propose to propagate only parameter values of high confidence and introduce confidence intervals for this purpose.

Confidence interval We use the approach in [3] that is based on the profile likelihood [5] to estimate the confidence intervals of parameter values. The profile log-likelihood of a parameter $p_k \in \theta_i$ by fixing it to a value ν can be defined by

$$PLL_{p_k}(\nu) = \max_{\theta_i \in \{\theta_i | p_k = \nu\}} LL(D_i|\theta_i)$$

And the confidence interval for parameter p_k is

$$CI_\alpha(p_k) = \left\{\nu \mid -2PLL_{p_k}(\nu) \leq -2LL(D_i|\theta_i^*) + \Delta(\alpha)\right\}$$

where α is the confidence level. The threshold value $\Delta(\alpha) = icdf(\chi_1^2, \alpha)$ is the α-quantile of a χ^2 distribution with one degree of freedom.

Interestingly, by using this method the final prediction can be reliable even when its parameters have a large confidence interval, as it is the case in circuit C_3. The combination of this ensemble learning and high-confidence parameter propagation is what leads to superior parameter inference results. Figure 3 depicts the solution to our case study by following Algorithm 1. Our approach can estimate the parameter value with a smaller error in all cases except for the parameter n_{L_x}, where both approaches are similar.

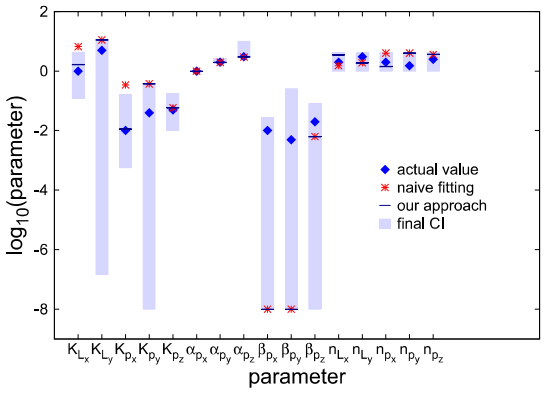

Figure 3: A comparison between actual and estimated parameter values.

3. DISCUSSION

We presented a new approach to infer the parameter value for multiple models with smaller error. Future work will be extension of this technique to more complex models and assessment of various optimization techniques such as symbolic computation or pattern search to reduce the computational cost.

Algorithm 1: Parameter inference

Input: Models M_i, datasets D_i, threshold values α, ε
Output: Parameter values and their confidence interval
begin
  $CI_\alpha(p_k) = Range(\nu_g) = (-\infty, +\infty)\ \forall p_k \in \theta, g \in C_i$
  repeat
    for $i = 1 \rightarrow n$ **do**
      $G = \{g \in C_i \mid Range(\nu_g) < \varepsilon\}$
      $\theta' = \{p_k \in \theta_i \mid \exists g \in G \wedge p_k\ affects\ \nu_g\}$
      $\theta'' = \{p_k \in \theta_i \mid CI_\alpha(p_k) < \varepsilon\}$
      $\hat{\theta}_i = \theta_i \setminus (\theta' \cup \theta'')$
      Fix value of ν_g $(g \in G)$ and $p_k \in \theta''$ in M_i
      Estimate $\hat{\theta}_i$ by fitting M_i with D_i
      Update $CI_\alpha(p_k), Range(\nu_g)\ \forall p_k \in \hat{\theta}_i, g \in C_j$
  until $CI_\alpha(p_k)$ *and* $Range(\nu_g)$ *do not change*

4. REFERENCES

[1] S. B. et al, "A synthetic multicellular system for programmed pattern formation," *Nature*, 2005.

[2] K. T. et al, "Refactoring the nitrogen fixation gene cluster from klebsiella oxytoca," *PNAS*, 2012.

[3] A. R. et al, "Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood," *Bioinformatics*, 2009.

[4] R. N. G. et al, "Universally sloppy parameter sensitivities in systems biology models," *PLoS Comput. Biol.*, 2007.

[5] D. V. et al, "A method for computing profile-likelihood-based confidence intervals," *Applied Statistics*, 1988.