October 13, 2015

Genomic Time-Series Modeling Using Recurrent Neural Networks

Arya Iranmehr AIRANMEHR@UCSD.EDU
Vineet Bafna VBAFNA@CS.UCSD.EDU
Ali Akbari ALAKBARI@ENG.UCSD.EDU

Abstract

The advent of Next Generation Sequencing (NGS) has made it possible to study genomic data throughout time. This modern paradigm, Evolve-and-Resequence (E&R), enables us to make more accurate and robust inferences, i.e. estimate model parameters, using multiple observations along generations. In this paper, we consider the recently repopularized Recurrent Neural Networks (RNN) to model the genomic time series of E&R. In fact, RNN is used as a generative model which for a initial estate and a choice of model parameter generates a sequence. Parameter estimation procedure involves a (non-convex) optimization of least square loss between observed sequence data and RNN-generated sequence with respect to model parameter. Backpropagation-in-time, is effectively used to compute gradients of objective function, and stochastic gradient descent with momentum algorithm is used for optimization. Experimental study on simulated data shows RNN provides significantly more accurate and robust estimates in shorter times.

1. Introduction

Until very recently, biological data analysis has been considered processing a snapshot of data. However, the emergence of NGS and related technologies has made it possible to not only create larger datasets but also to measure multiple observations of the same quantity in the course of time. In many cases, such as population genetics, it is of the great interest to model the evolutionary process and make inferences, predictions and retrospective studies. Indeed, a random process is better explained by time series data than a single observation.

In addition to inexpensive data availability, over last two decades, a large amount of efforts is dedicated to High-Performance Computing (HPC), which re-popularized and re-branded computationally intensive algorithms such as Neural Networks. The first properly proposed neural network model to exploit full potential of multi layer neural networks published by Hinton and Salakhutdinov (2006); Hinton et al. (2006) and its spectacular performance on image processing problems immediately spawned the field of Deep Neural Networks (DNN), aka Deep Learning. Shortly after, DNNs went beyond the tasks that they are initially indented to accomplish Krizhevsky et al. (2012) and had breakthroughs in time-series DNN models, aka RNNs, such as generative models Sutskever et al. (2011), speech processing Hinton et al. (2012) etc.

In this paper we aim to use the tools and machinery that has been developed for RNNs, to model times-series biological data. In particular, we consider the population genetics problem of finding loci (locus) under selection given observation of allele frequency of a

population in different generations of a Wright-Fisher model. This problem has been previously treated by using Gaussian Processes **?**, spectral methods Steinrücken et al. (2014)

2. Methods

In this section we formally present describe RNN model and a Naive method which takes $\mathcal{O}(1)$ computations as baseline performance.

2.1 Notation

Let $\mathbf{X} \in [0,1]^{R \times T \times L}$ be the 3D array (Tensor) of containing allele frequencies of the population for which $\mathbf{X}_{r,t,l}$ is the allele frequency of population for r^{th} replication, time t and location l an d R is the number of experimental replicates, T is the number of observations along time, and L is the number of sites. Since replicates are identically and independently distributed (iid), we define $X \in [0,1]^{T \times L}$ as matrix of allele frequencies for a replicate, $X^{(t)} \in [0,1]^{L}$ allele frequencies at time t and $X_l \in [0,1]^{T}$ is the sequence of allele frequencies at loci l.

2.2 Wright-Fisher Process

For simplicity, in this paper we use deterministic bi-allelic single-locus Wright-Fisher (WF) model **?**. Under this model allele frequency the allele frequencies evolve

$$x_{t+1} = f(x_t; s, h) \tag{1}$$

where h is the overdominance and transition function is defined

$$f(x) = \frac{(1+s)x^2 + (1+hs)x(1-x)}{(1+s)x^2 + 2(1+hs)x(1-x) + x(1-x)} = x + \frac{s(h+(1-2h)x)x(1-x)}{1+sx(2h+(1-2h)x))} \tag{2}$$

by setting $h = 0.5$ we have

$$f(x) = x + \frac{sx(1-x)}{2+2sx} \tag{3}$$

2.3 Gaussian Process

The Gaussian Process optimizes

$$\arg\max_{\theta} \quad \mathcal{L}(\mathbf{X}|\theta) \tag{4}$$

where $\mathcal{L}$ is Gaussian distribution log-likelihood, i.e. negative weighted least-squares loss. Mean and covariance functions of the GP at any t, l are functionally dependent to parameter of interest θ, and computed using transition function of the WF process Terhorst et al. (2015).

2.4 Recurrent Neural Network

Sutskever (2013) (Rumelhart et al., 1986)

2.5 Naive Method

Stephan et al. (2006)

$$x_t = \frac{x_0}{x_0 + (1 - x_0)e^{-st/2}} \tag{5}$$

where solving for s we have

$$s = 2t\frac{x_t(1 - x_0)}{x_0(1 - x_t)} \tag{6}$$

3. Experiments

Hudson (2002) Peng and Kimmel (2005) Bergstra et al. (2010)

4. Discussions

References

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU Math Expression Compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, 2006.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Others. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Richard R Hudson. Generating samples under a WrightFisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, February 2002.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Bo Peng and Marek Kimmel. simuPOP: a forward-time population genetics simulation environment. *Bioinformatics*, 21(18):3686–3687, September 2005. doi: 10.1093/bioinformatics/bti584.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

Matthias Steinrücken, Anand Bhaskar, and Yun S Song. A novel spectral method for inferring general diploid selection from time series genetic data. *The annals of applied statistics*, 8(4):2203, 2014.

Wolfgang Stephan, Yun S Song, and Charles H Langley. The Hitchhiking Effect on Linkage Disequilibrium Between Linked Neutral Loci. *Genetics*, 172(4):2647–2663, April 2006.

Ilya Sutskever. *Training recurrent neural networks*. PhD thesis, University of Toronto, 2013.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

Jonathan Terhorst, Christian Schlötterer, and Yun S Song. Multi-locus Analysis of Genomic Time Series Data from Experimental Evolution. *PLoS Genet*, 11(4):e1005069, 2015.